Non-Rigid Structure-from-Motion

Thesis by
Suryansh Kumar
to
College of Engineering and Computer Science

in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy
in the subject of
Computer Vision

Australian National University
Canberra, ACT 2601, Australia
2019

Declaration of Originality

I hereby declare that the research presented in this Thesis is my own original work except explicitly referred or indicated. This work contains figures, plots and simulation results which are my own authentic results which has never been submitted before to any university or press except otherwise indicated. The content of this Thesis is based on the published conference proceedings, journal and submitted articles as specified in the "Thesis Outcome".

Copyright Declaration

Advisors: Prof. Yuchao Dai, Prof. Hongdong Li Suryansh Kumar

Non-Rigid Structure-from-Motion

Abstract

This thesis revisits a challenging classical problem in geometric computer vision known as "Non-Rigid Structure-from-Motion" (NRSfM). It is a well-known problem where the task is to recover the motion and 3D shape of a non-rigidly deforming object from image data. A reliable solution to this problem is valuable in several industrial applications such as virtual reality, medical surgery, movies *etc*. To date, there does not exist any algorithm that can solve NRSfM for all kinds of conceivable motion. As a result, additional constraints and assumptions are often employed to solve NRSfM. The task is challenging due to the inherent unconstrained nature of the problem itself as many 3D varying configurations can have similar image projections. The problem becomes even more challenging if the camera is moving along with the object.

The thesis takes on a modern view to this challenging problem and proposes a few algorithms that have set a new performance benchmark to solve NRSfM. The thesis not only discusses the classical work in NRSfM but also proposes some powerful elementary modifications to it. The foundation of this thesis surpass the traditional single object NRSFM and for the first time provides an effective formulation to realize multi-body NRSfM.

Most techniques for NRSfM under factorization can only handle sparse feature correspondences. These sparse features are then used to construct the scene using organization of points, lines, planes or other elementary geometric primitive. Nevertheless, sparse representation of the scene provides an incomplete information about the scene. This thesis goes from sparse NRSfM to dense NRSfM for a single object, and then slowly lifts the intuition to realize dense 3D reconstruction of the entire dynamic scene as a global as rigid as possible deformation problem.

The core of this work goes beyond the traditional approach to deal with deformation. It shows that *relative scales* for deforming objects under perspective projection can be recovered under some mild assumption about the scene. The work proposes a new approach for dense detailed 3D reconstruction of a complex dynamic scene from two perspective frames. Since the method does not need any depth information nor it assumes a template prior, or per-object segmentation, or knowledge about the rigidity of the dynamic scene, it is applicable to

a wide range of scenarios.

Lastly, this thesis provides a new way to perceive the depth of a dynamic scene which essentially trivializes the notion of motion estimation as a compulsory step to solve this problem. Conventional geometric methods to address depth estimation requires a reliable estimate of motion parameters for each moving object, which is difficult to obtain and validate. In contrast, this thesis introduces a new motion-free approach to estimate the dense depth map of a complex dynamic scene for successive/multiple frames. The work show that given per-pixel optical flow correspondences between two consecutive frames and the sparse depth prior for the reference frame, we can recover the dense depth map for the successive frames without solving for motion parameters. Experimental results and MATLAB codes on relevant examples are provided to validate the motion-free idea.

Acknowledgments

First and foremost, I would like to thank my supervisory panel. Most Ph.D. students feel blessed to get one supportive supervisor. I had three!. Yuchao Dai: this thesis would not have been possible without your continuous support and guidance. Thank you for introducing me to the world of non-rigidity. Your meticulous guidance and constant effort to develop me as a researcher is a perpetual source of knowledge and inspiration. Your constant rigor when I have been sloppy, helped me to learn so, so much. Your complete faith and belief in my abilities gave me a certificate to experiment over my ideas and put it to work. Most of all, thank you for your unshakable patience, humility and providing me the freedom to explore new ideas. Hongdong Li: Where do I begin? Thank you for all those productive research discussions, hours of meeting sessions and always putting me to think visually. Your ability to bring ideas from different fields of science to computer vision problems has increased my dimension of thinking, so a BIG thank you for that. Also, thank you for patiently listening to my request and always responding to my "Door knock". Thank you for correcting my paper writing and exposing my common mistakes including paper titles and my terrible figures and, last but not the least, surviving my insufferable presentation slides. Richard Hartley: Thank you for the all those short discussions and sharing the story of "In Defense of Eight Point algorithm" with me. Many thanks to you for yelling at me when I didn't follow the door rules, which invariably reminds me "Always follow the rules".

Thanks to the Australian National University and its HDR team for providing me the financial support through merit scholarship scheme for the entire duration of my Ph.D.

I thank my parents (Mum, Dad), brothers (Aditya Kumar, Vikramaditya Kokil) and sister (Nandini) for their love and support. Thank you Roya Safaei for all the constructive suggestions on writing papers and always encouraging me to go for the best. I thank my fellow collaborators and lab-mates Anoop Cherian, Mehrtash Harandi, Yuo Lu, Yonhon Ng, Rodrigo S. Cruz, Hongsheng Qi and Jing Zhang for the fruitful discussions. Thank you Ram Srivatsav Ghorakavi for simulating deep-learning codes for CVPR'19 work and providing me the Interstellar dataset feature correspondences. Lastly, I thank my friends Dany Fauzan, Peter Lockey, Dannie and Marcela Velasquez for their time and support.

I dedicate this thesis to my family. Many thanks to my elder brother "Aditya Kumar", you are all the reasons I came this far.

Contents

Thesis Outcome

Publications

[1] Suryansh Kumar, "Non-Rigid Structure from Motion: Prior-Free Factorization Method Revisited" (WACV), IEEE, 2020, Colorado, USA. [101, 102, 103].

[2] Suryansh Kumar, Ram Srivatsav Ghorakavi, Yuchao Dai, Hongdong Li, "Dense Depth Estimation of a Complex Dynamic Scene without Explicit 3D Motion Estimation", (Under Preparation) [115, 116].

[3] Suryansh Kumar, Yuchao Dai, Hongdong Li, "Superpixel Soup: Monocular Dense 3D Reconstruction of a Complex Dynamic Scene", IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI) 2019 [113].

[4] Suryansh Kumar, "Jumping Manifolds: Geometry Aware Dense Non-Rigid Structure from Motion" (CVPR) 2019, Long Beach, CA, USA [99, 100].

[5] Suryansh Kumar, Anoop Cherian, Yuchao Dai, Hongdong Li, "Scalable Dense Non-Rigid Structure from Motion: A Grassmannian Perspective", (CVPR) 2018, Utah USA [105, 104].

[6] Suryansh Kumar, Yuchao Dai, Hongdong Li, "Monocular Dense 3D Reconstruction of a Complex Dynamic Scene from Two Perspective Frames", (ICCV) 2017, Italy Venice [111, 112].

[7] Suryansh Kumar, Yuchao Dai, Hongdong Li, "Spatio-Temporal Union of Subspaces for Multi-body Non-rigid Structure-from-Motion" 71:428-443, (Pattern Recognition), Elsevier (2017) [107, 108].

[8] Suryansh Kumar, Yuchao Dai, Hongdong Li, "Multi-body Non-rigid Structure from Motion" (3DV), IEEE, 2016, Stanford University, California, USA [109, 110].

Awards

- Nominated for J. G. Crawford Prize at ANU for Best Thesis 2019 (Interdisciplinary).
- Best Algorithm Award for "Multi-body Non-rigid Structure-from-Motion" in NRSFM Challenge at CVPR 2017 awarded by Disney Research (AUD $1200) as prize money.
- Vice Chancellor Grant and Student Funding to attend CVPR'18, ICCV'17 and ICML'17.
- ANU Merit Scholarship Student funded in part by Australian Research Council.

1 Introduction

Contents

1.1 Structure from Motion

What is Structure-from-Motion and why it's so important? The problem of estimating three dimensional structure of the scene from images when either the camera or the object or both are in motion is known as Structure-from-Motion (SfM). This topic has been of interest to

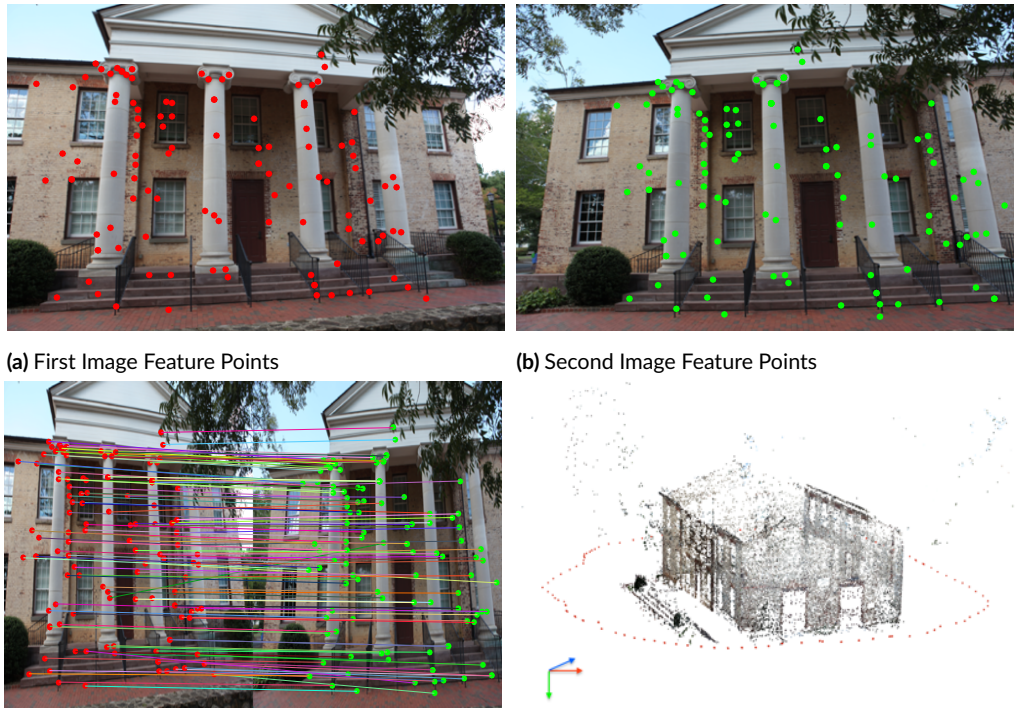
the researchers since the inception of computer vision field and is still an active field of research [18, 111, 106]. Solving this inverse problem to infer the geometry of the scene from images has alone taken more than three decades and still counting [18]. The main reason for such gradual progress in this field is possibly due to the nature and setting of the problem itself. Despite that, a lot of successful SfM algorithms has been proposed in the past which works quite well under certain assumptions about the scene and motion. Having said that, SfM for any general dynamic scene is still an open area for researchers to solve.

Solving SfM is important not only for machines but also for humans in resolving and understanding the extraordinary abilities of human perception. Solution to this problem can be of paramount importance to medical surgery, street mapping, coal mining, space exploration, scene understanding, autonomous driving and many more.

Due to its wide range of applications, this field has been the center of attention to researchers from vision, robotics, medical etc. In the field of robotics, the core challenge is autonomous navigation which requires reliable algorithms for obstacle avoidance, frontier detection, sensor localization etc [167, 114, 117, 179]. For robots to emulate the human ability to localize and understand the geometry of the environment, it needs structure or map of the scene. In a similar way, medical researchers needs an accurate and precise understanding of the human body parts from images for surgery or treatment. The success of all these applications to large extent depends on the richness of information represented by the reconstructed scene model. For instance, inference about an object can be greatly improved by the knowledge of its 3D structure.

In quest of finding a reliable solution to this problem, researchers spend considerable period to time to realize that SfM for rigid scenes can be solved with a reasonable accuracy [125, 169, 86, 160, 4]. However, for a dynamic or non-rigid scene it is still a challenging task. For dynamic scenes any projected position in a camera image plane can have several possible 3D configuration. Therefore, additional information which may be related to the geometry, appearance or motion of the objects in the scene is required to solve this problem. These additional information or prior knowledge helps to reduce the number of degrees of freedom. For example: constraints such as parallelism, co-planarity, orthogonality can be used to reconstruct simple geometric shapes. To gather more knowledge about the scene two or more images of the scene is used for reconstruction. Additionally, several other assumptions such as orthographic projection, low-rank shape are used to solve this problem.

Due to stiff theoretical complexity of the problem, modification in sensors has also been employed. Sensors such as stereo camera, RGB-D has been exploited to procure the scene ge-



(a) First Image Feature Points

(b) Second Image Feature Points

(c) Image Feature Matching

(d) 3D reconstruction of the scene

Figure 1.1: A high-level illustration of basic pipeline for rigid scene reconstruction using multiple-view geometry method [85]. (a)-(b) Detect the interest points across multiple frames (shown only for two images). (c) Assign descriptor to each features and match these feature descriptors across images. (d) Solve for motion and 3D points using essential matrix decomposition and triangulation respectively. Refine the solution using bundle-adjustment [177]. The above dataset is taken from gerrard-hall sequence [153]. (Best viewed on screen)

ometry. Although new understanding has been interleaved with this massive engineering in development of sophisticated sensors. Nevertheless, reconstructing dense detailed structure of a dynamic scene is still a challenging problem as these sensors have their own limitations.

Based on the algorithms proposed in past, one can classify SfM based on different attributes such as types of sensors, types of motion, number of frames, types of projection and many more. However, this thesis covers SfM based on types of motion, which can be broadly classified into two significant family: rigid SfM and non-rigid SfM. Briefly, a transformation such that the distance between the points is preserved before and after motion is called rigid motion transformation else it can be termed as non-rigid motion transformation.

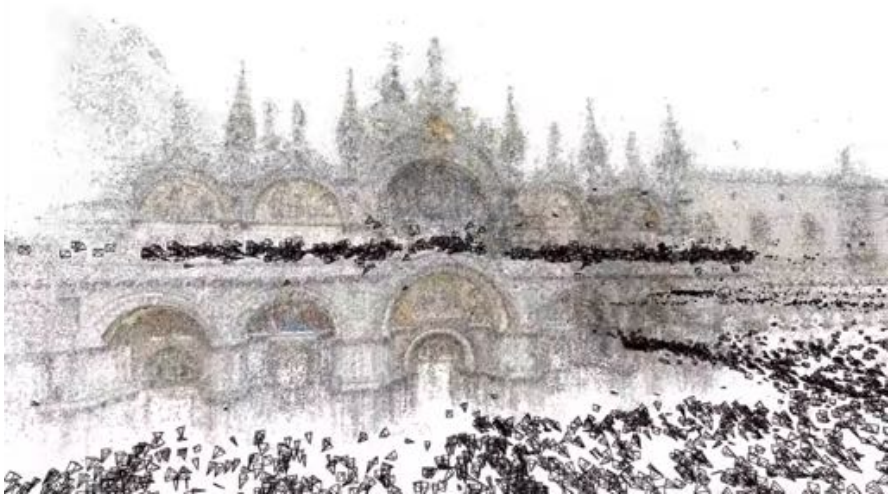


Figure 1.2: Large scale structure from motion using Internet photo collection [4]. This 3D structure of Saint Mark's Basilica is recovered by harvesting the images from the web. Here, the black color frustum show the camera position. (Note: This image is taken from Agrawal.S *et al.* work [4])

1.2 Rigid Structure from Motion

Structure from motion under the key assumption that object is moving rigidly or only camera is moving and the scene is static can be termed as rigid structure from motion. Theory related to the solution of SFM under rigid motion assumption is very mature and can be considered as a solved problem [169, 85, 88, 160, 4, 95] (see Figure 1.1). With elegant theory and optimization techniques in hand researchers have extended this to reconstruct multiple object in the scene while camera is moving, popularly known as multi-body structure from motion [57]. The idea of multi-body SFM is to cluster feature tracks and fit rigid motion model to each cluster. As each cluster is assumed to be rigid, techniques described in [86] can be applied to infer 3D points. The theory of rigid SFM has also been extended to large scale multi-view reconstruction, where the goal is to reconstruct entire city by collecting images from Internet of the same scenes taken by different user camera's [36, 156, 160, 4] (see Figure 1.2). The magnitude of success accomplished with the theory of rigid SfM is enormous [157] but still it has certain limitations.

Limitations

Despite the fact that the classical theory developed for rigid structure from motion provides satisfactory results for rigid scenes [87, 88, 138, 177], its usage to large scale application needs

efficient optimizers and different variety of modern SfM pipelines [158, 159, 156, 4]. To provide robustness to the solution of rigid Structure-from-Motion, incremental approach is also adopted but it makes the execution quite slow. As a result, motion averaging approaches are adopted in the recent past which provides robust results for large scale problems [29, 76, 75, 54]. In conclusion, several ideas in the past are proposed to provide a reliable result for large scale rigid SfM problems, however, there is still scope of improvements in its intrinsic pipeline such as camera registration, robustness to noise, convexity, dense solution to rigid SfM *etc*.

Matrix factorization also provides an alternative way to solve rigid Structure-from-Motion using batch of frames [169, 43, 130]. However, it's application to large scale problems are limited.

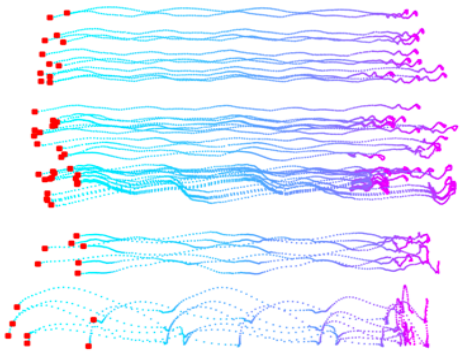
1.3 Non-Rigid Structure from Motion

The other family of SfM is popularly known as "Non-Rigid Structure-from-Motion". Under non-rigid deformation, its difficult to infer the shape and motion model of the object using only image data. Also, if arbitrary deformations are allowed, then, 3D reconstruction of a non-rigid moving object is still considered as an ill posed problem. Consequently, additional assumption about the object or the scene is required to solve this problem. Some of the popular assumptions for handling non-rigid SFM problem are a) Restrict the shape to lie on a low-dimensional subspace [21], [174]. b) Orthographic camera projection [21], [174], [42], [6]. (c) Only one non-rigid shape is present in the scene . Figure (1.3) illustrates the basic working pipeline for non-rigid shape reconstruction using factorization approach.

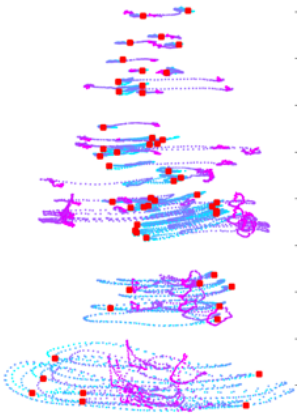
The first practical solution to NRSfM [21] extended the classical *factorization* framework [169] under the assumption that 3D shape in each frame is a linear combination of a set of basis shapes. However, such an assumption does not provide satisfactory solution to the problem as its formulation is inherently under-constrained and it requires more prior knowledge/constrained on 3D shape deformation to supply better results. Xiao *et al.* [197] in 2004 proposed that NRSfM is an ill-posed problem and orthonormality constraints proposed in the previous works are *alone* not sufficient to recover shape basis and shape coefficient uniquely. Consequently, Xiao *et al.* proposed to add extra basis constraint to solve the problem. Following the same underlying theory Torresani *et al.*[175] used Gaussian priors to estimate shape coefficients. In contrast, Akther *et al.* [7] first pointed out that the metric constraints are alone sufficient to 3D shape without ambiguity, though the ambiguity in shape basis is inherent.

A recent break-through in NRSfM proposed a prior free approach to NRSFM under low-rank shape as the only assumption [42]. This paper is able to show —both theoretically

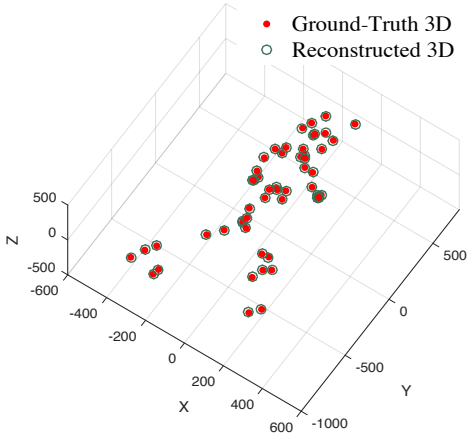
(a) 2D Image Feature Points



(b) Feature Point track across all the images



(c) Mean centralized 2D features tracks



(d) 3D reconstruction of the non-rigidly moving object

Figure 1.3: A high-level illustration of basic pipeline for 3D reconstruction of a non-rigidly deforming object using factorization approach. (a) The 2D image feature for the first image. (b) Feature tracks or trajectories of the 2D points across all the frames. (c) Mean centralization of the feature track to remove translation component. (d) 3D reconstruction of the non-rigidly moving object points. The above dataset is taken from walking sequence introduced by Torresani *et al.*[174] (Best viewed on screen)

and practically, that without any extra knowledge about the non-rigid object (other than low-rank) it's possible to reconstruct the shape without any inherent basis-ambiguity. However, its performance on the benchmark dataset [6, 174, 93] is arguable. Therefore, the main concern is, with the recent theoretical surge in the understanding of NRSfM both theoretically and practically [44], Is SFM for any general non-rigidly moving object/scene is solved? At the time of writing this thesis, the answer remains 'NO'!. Some of the reasons for this disappointing answer are listed in the following limitations.

Limitations

Even though few successful algorithms in NRSFM can provide satisfactory results, its still an unsolved problem for any general dynamic scene. The reasons are as follows:

- Most of the successful research in NRSFM assumes orthographic projection which limits its application to widely used perspective camera model.
- NRSfM methods assumes single non-rigid object is present in the scene for entire image sequence. In general, scene is composed of multiple moving objects.
- To realize per-pixel (dense) deformation of a shape, usually 3D templates are employed, which again is a non-practical and non-scalable approach to solve dense NRSfM.
- One of the most important limitation of the NRSfM in general is the validity of the correct representation of the deformation model. More precisely, which type of deformation model can explain the non-rigidity of the object in the scene, for example low-rank, isometric deformation model, piece-wise rigid model etc. To me, its an open-problem.

Due to the above limitations, NRSfM algorithms are not widely applicable for practical purposes. In this thesis, we develop algorithms which enables new insight to solve NRSfM and provides some practical approaches to solve real world NRSfM problems. This thesis is also about realizing rigid 3D reconstruction problem as a small subset of problem available in this vast world of non-rigidity.

In the remaining section of this introduction chapter, we provide small summary on the motivation behind each of the work. Firstly, we introduce modification to the existing single body NRSFM framework to supply better 3D reconstruction results. Then, we outline the limitations with single body framework and how matrix factorization approach can be used to achieve the multi-body NRSfM. Further, we introduce the problem with sparse representation of the object and brief on our two different algorithm to estimate dense 3D reconstruction of a non-rigid object. After that we discuss on the drawbacks of orthographic

model assumption. We slowly lift the insight of the readers and motivate them to think of any general dynamic scene as a global as rigid as possible scene, if observed closely within subsequent time frame (assuming this time is small enough), hence a NRSfM problem. To endorse our intuition we outline two methods to solve dense 3D reconstruction of a general dynamic scene. Finally, we provide the chapter-wise progression of the thesis from sparse to dense 3D reconstruction.

1.4 Prior-Free NRSFM Factorization: Modifications and Improvement

Bregler *et al.*[21] matrix factorization approach proposed in the year of 2000 is one of the most widely used framework to solve NRSfM problem. After that, more than a decade of profound attempts to extend this framework were unable to provide a practical algorithm to solve this problem. Finally, it was in the year 2012 that Dai *et al.* provided a new insight to solve NRSfM which is popularly known as "prior-free" approach. The theory and algorithm proposed by Dai *et al.*[42] in a way changed the course of current research in NRSfM. However, overtime it was observed that their method fails to provide acceptable 3D reconstruction results on available dataset. As a result the prevailing view about this work is that it provides arguable results and hence, methods using compact data representation lashes on its performance [205]. So, the question we ask is Dai *et al.* seems theoretically correct but suffers practically "Why"?.

This thesis firstly provides the possible reasons for its practical failure. Our work gives an in-depth understanding of "prior-free" method and how some powerful elementary measures and modifications can significantly improve its performance. We argue that by properly utilizing the well-established assumption about a non-rigidly deforming shape *i.e.*, it deforms smoothly over frames and it spans a low-rank space, the simple prior-free method can provide results which is comparable to the best available algorithms —at the time of writing this thesis. Similar to prior-free method the only assumption we make is "low-rank" shape, and we show that a better solution to motion which satisfies smooth motion assumption is already present within the estimated "Gram matrix", and explicit regularization on motion is not essentially required. Secondly, we propose how to better utilize the low-rank assumption. The improved performance is justified and empirically verified by extensive experiments on several benchmark datasets. Finally, this work also conjecture some theoretical problems which we think needs attention for further developments in factorization approach to NRSfM.

1.5 From single body to multi-body NRSFM.

The existing algorithms to solve the task of NRSFM are limited to handle one non-rigid object in the scene, which restricts its application to a general scene. Real world scenario's often

consists of multiple objects undergoing non-rigid deformation. Therefore, we must look for an approach that solves multi-body NRSFM. One way to handle this it to solve 3D reconstruction task for each non-rigid object one at a time by pre-segmenting different objects in the scene. Nonetheless, its not a optimal way to solve the problem in which both the motion and shape interacts. Under the assumption that each non-rigid object spans a distinct global linear subspace, this thesis present the first algorithm to realize multi-body NRSfM [109]. To compactly represent complex multi-body non-rigid scenes, we propose to exploit the deformation in both spatial and temporal space, thus achieving a spatio-temporal representation. Specifically, we represent the 3D shape deformation in a union of subspaces in the temporal space and the 3D trajectories in the union of subspaces in the spatial space. Such spatio-temporal representation not only provides competitive 3D reconstruction but also gives reliable segmentation of multiple non-rigid objects present in the scene.

1.6 From Sparse NRSFM to Dense NRSFM

For many real-world applications, such as facial expressions, heart-surgery *etc*, dense or per-pixel reconstruction of the object is very essential. NRSfM algorithms developed to reconstruct few sparse points of the non-rigid object fails to provide dense reconstruction of the object and therefore, its unable to cater the subtle deformation in the object. The framework developed under the assumption of global low-rank shape and the shape spans global linear subspace may not hold for dense deforming surface. The main reason for it is, any complex deforming surface can be composed of several local linear subspace structure. Therefore, the algorithm developed for sparse NRSfM fails to cater the inherent local structure of the deforming shape over space and time. This thesis lifts the intuition developed for union of subspaces in NRSfM problem and modifies it further to provide a scalable dense NRSfM algorithm. Our work utilizes Grassmannian representation to solve dense NRSfM which was previously studied only to represent set of images.

1.7 Dense monocular 3D reconstruction of a complex dynamic scene.

The method outlined before works well under the orthographic camera model assumption. However, the widely used camera are perspective in nature and orthographic camera model may not hold. With the proliferation of monocular perspective camera in mobile robots and cell phones has increased the demands for sophisticated reconstruction algorithm's. These reconstruction algorithm should not be restricted to camera model and types of motion in the scene. Hence, it should be flexible enough to work smoothly for any general scene of unknown rigidity type. In order to support geometric reasoning for images, maps, obstacles, environment etc. such algorithm is required for future smart devices. In this thesis, we pro-

pose two geometric algorithm that can help in achieving dense detailed 3D reconstruction of a dynamic scene under some mild assumption. Consider a general real-world dynamic scene, the change we observe in the scene between consecutive time frame is not arbitrary, rather it is regular. Hence, if we observe a local transformation closely, it changes rigidly, but the overall transformation that the scene undergoes is non-rigid. Therefore, to assume that the dynamic scene deforms as rigid as possible seems quite convincing and practically works well for most real-world dynamic scenes.

Importance

The topic covered in this thesis is of sheer importance to science and technology as it has tremendous application in medical, robotics, architecture, design, tourism, gaming, and many more. For instance, imagine a mobile robot which can capture the spatial layout of underground coal mine field, a precise medical surgery without any human supervision, automatic traffic or driving system, 3D models of your favorite monuments or building or actors, a truly immersed virtual reality experience for 3D game. All these application needs a robust dense 3D reconstruction of the involved scene. One can argue to use laser and depth sensing devices. However, such sensor is very costly with its own limitations and it is not portable enough to be embedded in smart portable devices with current technology. So, the argument here is; can we come up with some algorithm that uses the current imaging and computing resources to supply reliable geometry of a general dynamic scene.

1.8 Thesis Outline

After a brief introduction on structure from motion and brief overview of our thesis, we are ready provide progression of this thesis. At the beginning of each chapter in this thesis, we briefly discuss the motivation behind the concerned work. This discussion is followed by a comprehensive literature survey, where we review the relevant research area specific to topics covered therein. Our literature survey also tries to highlight the gray areas of the previous works.

The thesis starts with the classical approach to NRSfM. In the Chapter (2), we attempt to make the baseline factorization approach more accurate and usable to real world application.

In Chapter (3) we describe our first multi-body NRSFM that enables joint reconstruction and segmentation of deformable objects. We also extend the formulation to compactly represent both shape space and trajectory space via elastic net regularizer. Later, we describe the solution and implementation of the developed optimization framework followed by experi-

ments and results.

In Chapter (4) provides our work on dense non-rigid structure from motion which focuses on extending the idea of compact data representation using union of linear subspace to obtain per pixel 3D reconstruction of a deforming object.

An extension of the Chapter (4) is presented Chapter (5) where the motivation is to better utilize the Grassmannian representation developed in the previous chapter. The representation to group high dimensional data points inevitably introduce the drawbacks of categorizing samples on the high-dimensional Grassmann manifold. Therefore, to deal with such limitations, we propose to jointly exploits the benefit of high-dimensional Grassmann manifold to perform reconstruction, and its equivalent low-dimensional representation to infer suitable clusters. To achieve this, we project each Grassmannians onto a low-dimensional Grassmann manifold which preserves and respects the deformation of the structure w.r.t its neighbors. These Grassmann points in the lower-dimension then act as a representative for the selection of high-dimensional Grassmann samples to perform each local reconstruction.

In Chapter (6) we propose an efficient optimization framework to solve dense 3D reconstruction of complex dynamic scene using two perspective images. This work investigate on the rigidity of the scene using piecewise planar assumption. Under these assumptions, relative scale of objects in the scene can be recovered faithfully. We describe the details of the formulations and its implementation followed by extensive experimental results. These experimental results help conclude that dense detailed reconstruction using two perspective images is possible under some mild assumptions about the scene. In the following chapter (7), we took the assumption made in the Chapter (6) about a dynamic scene to next level. We proposed that if the depth for the reference frame is known a prior then we can estimate the dense depth map of a dynamic scene without using any 3D motion parameters.

1.9 State of the art

This brief section is included in the thesis to provide a quick reference to the state of the art in non-rigid structure from motion at the time of writing this thesis. The table below provides the evaluation statistics of NRSfM under classical setting *i.e.*, orthographic camera model with an assumption that the features tracks are given for the entire sequence of frames. These evaluations were done as a part of NRSfM Challenge organized at CVPR 2017. Interested readers are encouraged to refer to the Jensen *et al.* [93] work for detailed explanation on evaluation metric and experimental setups.

Algorithm	Mean RMS	Articulated	Balloon	Paper	Stretch	Tearing
Multibody[109, 107]	24.64mm	45.51mm	14.55mm	22.88mm	18.30mm	21.98mm
CSF2 [73]	26.09mm	35.51mm	19.01mm	33.95mm	23.22mm	18.77mm
RIKS [81]	26.75mm	42.11mm	18.45mm	32.18mm	22.88mm	18.12mm
KSTA [72]	26.86mm	35.63mm	24.88mm	31.96mm	24.25mm	17.59mm
MetricProj [142]	28.73mm	37.96mm	25.28mm	34.45mm	25.51mm	20.43mm
CSF [71]	30.83mm	36.84mm	30.43mm	32.17mm	28.87mm	25.82mm
PTA[8]	32.18mm	36.71mm	28.88mm	41.72mm	30.45mm	23.14mm
Bundle[46]	41.38mm	64.48mm	36.40mm	41.64mm	35.64mm	28.73mm
ScalableSurface[10]	41.84mm	58.12mm	31.71mm	45.45mm	38.88mm	35.03mm
RigidTriangle [166]	43.83mm	65.71mm	34.38mm	43.57mm	40.54mm	34.94mm
SoftInext [186]	45.80mm	61.43mm	36.75mm	47.41mm	45.56mm	37.87mm
EM PPCA [173]	47.86mm	46.62mm	36.87mm	51.56mm	58.01mm	46.21mm
BALM [47]	48.79mm	75.09mm	35.84mm	53.13mm	40.31mm	39.58mm
Compressible [98]	59.98mm	72.77mm	52.53mm	62.44mm	57.45mm	54.71mm
SPFM [44]	63.81mm	89.40mm	45.65mm	64.19mm	64.04mm	55.79mm
MDH [32]	67.37mm	88.66mm	58.27mm	66.98mm	66.27mm	56.67mm
Concensus [121]	70.53mm	105.38mm	54.71mm	64.25mm	69.22mm	59.10mm

Table 1.1: State of the art evaluation presented at the CVPR 2017 NRSfM Challenge [93]

1.10 Preliminaries

Before we start to discuss on the problem of non-rigid structure from problem, we provide an overview on the basics of algebra and optimization concepts. The discussion on these topics by no means comprehensive, and is provided for the understanding and completeness of the thesis. In case the reader wants to get a very clear picture on most of the concepts used in this thesis, I strongly recommend to go through MIT Linear Algebra on-line course [163] and articles on the subspace clustering [187, 188] and ADMM optimization [19].

(A) Trace of a Matrix: Let $X \in \mathbb{R}^{n \times n}$ be a square matrix. The trace of the matrix X is the sum of its main diagonal element. The trace of a matrix is also the sum of its eigen values.

$$Tr(X) = \sum_{i=1}^{n} X_{ii} = x_{11} + x_{22} + x_{33} + ... + x_{nn} = \sum_{i=1}^{n} \lambda_i(X) \tag{1.1}$$

where $\lambda_i(X)$ refers to the eigen values of X.

Basic Properties

- $Tr(X+Y) = Tr(X) + Tr(Y)$, Assuming X, Y are the matrix of same dimension
- $Tr(kX) = kTr(X)$, where k is a constant.
- $Tr(XY) = Tr(YX), X \in \mathbb{R}^{m \times n}, Y \in \mathbb{R}^{n \times m}$
- $Tr(X^T Y) = Tr(XY^T) = Tr(Y^T X) = Tr(YX^T)$
- $Tr(XYZW) = Tr(YZWX) = Tr(ZWXY) = Tr(WXYZ)$ *i.e.*, Trace is invariant under cyclic permutation

(B) Inner Product: Let $x \in \mathbb{R}^n, y \in \mathbb{R}^n$ be vectors of real numbers. The standard inner product on $\mathbb{R}^n$ is given by

$$< x, y >= x^T y = \sum_{i=1}^{n} x_i y_i \tag{1.2}$$

The standard inner product on real matrix $X, Y \in \mathbb{R}^{m \times n}$ is given by

$$< X, Y >= Tr(X^T Y) = \sum_{i=1}^{m} \sum_{j=1}^{n} X_{ij} Y_{ij} \tag{1.3}$$

(C) Rank of a Matrix: The rank of a matrix is defined as the maximum number of linearly independent columns/rows of the matrix. If $X \in \mathbb{R}^{m \times n}$ then

$$0 \leq rank(X) \leq min(m, n) \tag{1.4}$$

The rank can be thought as the intrinsic dimension of the matrix. Any matrix of rank r can be written as sum of r rank-one matrix *i.e.*,

$$X = \sum_{i=1}^{r} \lambda_i u_i v_i^T \tag{1.5}$$

Equivalently every $m \times n$ matrix can be decomposed as $X = U\Sigma V^T$ popularly known as singular value decomposition (SVD) where $U \in \mathbb{R}^{m \times r}$, $\Sigma \in \mathbb{R}^{r \times r}$ and $V \in \mathbb{R}^{n \times r}$.

(D) Norms: The function f usually denotes as $\|.\|_{symbol}$: $\mathbb{R}^n \mapsto \mathbb{R}_+$ is called norm if it satisfies the following properties:

1. f is non-negative *i.e.*, $f(x) \geq 0,\ \forall\, x \in \mathbb{R}^n$
2. f is definite *i.e.*, $f(x) = 0$ only if $x = 0$
3. f is homogeneous *i.e.*, $f(kx) = kf(x), \forall x \in \mathbb{R}^n$ and $k \in \mathbb{R}$
4. f must satisfy triangle inequality *i.e.*, $f(x + y) \leq f(x) + f(y), \forall x, y \in \mathbb{R}^n$

Examples of vector norms

Let x be a n-dimensional vector. The two very frequently used *norms* are l_1 norm and l_2 norm. The l_1 and l_2 norm of a vector is given by

$$\|x\|_1 = |x_1| + |x_2| + |x_3| + ... + |x_n| = \sum_{i=1}^{n} |x_i| \tag{1.6}$$

$$\|x\|_2 = (|x_1|^2 + |x_2|^2 + |x_3|^2 + ... + |x_n|^2)^{\frac{1}{2}} = \sum_{i=1}^{n} (|x_i|^2)^{\frac{1}{2}} \tag{1.7}$$

More generally l_p norm for $p \geq 1$ is defined as

$$\|x\|_p = (|x_1|^p + |x_2|^p + |x_3|^p + ... + |x_n|^p)^{\frac{1}{p}} = \sum_{i=1}^{n} (|x_i|^p)^{\frac{1}{p}} \tag{1.8}$$

As $p \to \infty$, the l_∞ norm or *Chebyshev* norm is defined as

$$\lim_{p\to\infty} \|x\|_p = \max(|x_1| + |x_2| + |x_3| + ... + |x_n|) \tag{1.9}$$

Example of matrix norms

Let $X \in \mathbb{R}^{m,n}$ be a matrix. Here, we will define some commonly used matrix norm in literature.

- l_1-norm of a matrix:

$$\|X\|_1 = \max_{1\leq j\leq n} \sum_{i=1}^{m} |x_{ij}| \tag{1.10}$$

- l_2-norm or spectral or operator norm:

$$\|X\|_2 = \lambda_{\max}(X) = (\lambda_{\max}(X^T X))^{\frac{1}{2}} \tag{1.11}$$

  where, $\lambda_{\max}$ refers to the largest singular value of X.

- Frobenius Norm:

$$\|X\|_F = \sqrt{< X, X >} = \sqrt{Tr(X^T X)} = \sqrt{\sum_{i}^{r} \lambda_i^2} \tag{1.12}$$

  where, λ_i is the i^{th} singular value of the matrix and r is the rank of the matrix.

- Nuclear norm or Trace norm:

$$\|X\|_* = \sum_{i=1}^{r} \lambda_i(X) \tag{1.13}$$

  i.e., nuclear norm is the sum of singular values of a matrix. Here r is the rank of the matrix.

(E) Vector Spaces and Subspaces: Our brief discussion on this topic is inspired from Strang.G book [163] and Lecture 6 of MIT 18.06.
Let $v, w \in \mathbb{R}^n$ be two vector in a n-dimensional space. The vector space requirements are:

- If we add these two vector in the space, the answer stays in the same space *i.e.*, v, w, and $v + w$ are in the same space.

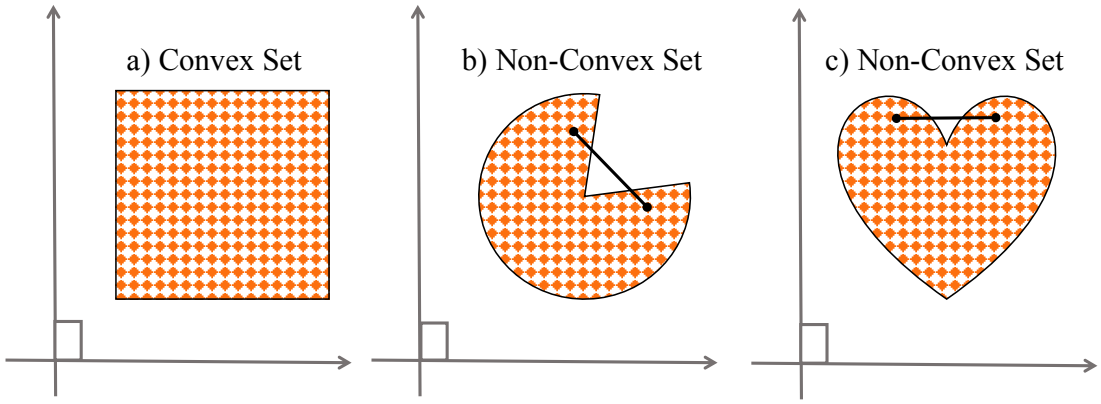


Figure 1.4: Examples of convex and non-convex sets. (a) The square which includes its boundaries is convex (b) The pacman shaped set is non-convex, the line segment between the shown points in the set is not contained in the set. Similarly, set (c) is non-convex.

- If we multiply vectors with some scalars in the space, the answer remains in the same space *i.e.*, v, kv are in the same space for some real number k.
- All the linear combination $k_1v + k_2w$ stay in the same space. Here k_1 and k_2 denotes any real numbers.

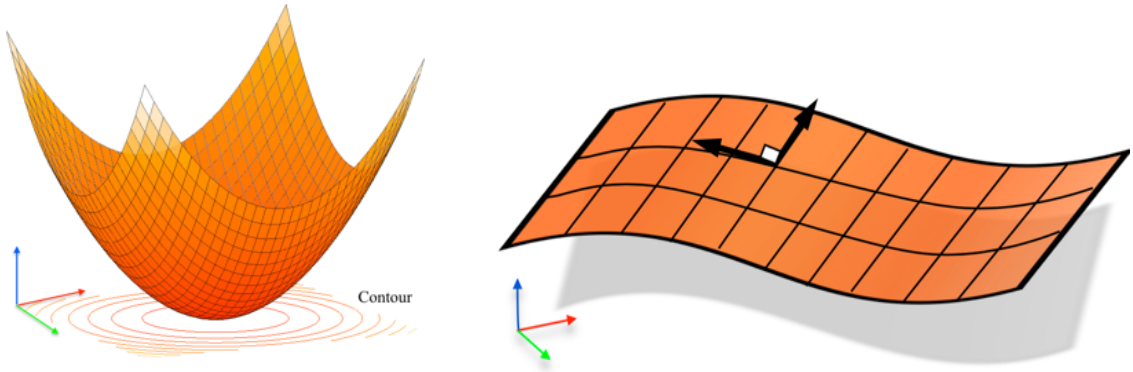
The vector space inside this n-dimensional vector space is called the subspace of $\mathbb{R}^n$. The subspace of a vector space is a set of vectors (including 0) that satisfies two requirements: *If v, w are the vectors in the subspace and k is any scalar, then*

1. $v + w$ is in the subspace.
2. kv is in the subspace.
3. All linear combination stay in the subspace.

(F) Convex Analysis: Here we will discuss some the basic definition that are important for the convex analysis an optimization problem. Our discussion is inspired from Boyd.S and Vandenberghe.L book on Convex Optimization [20].

Definition 1. *A set C is convex if the line segment between any two points in C lies in C, i.e., if for any $x_1, x_2 \in C$ and any $\vartheta \in [0, 1]$, we have*

$$\vartheta x_1 + (1 - \vartheta)x_2 \in C \tag{1.14}$$



(a) Convex Function **(b)** Non-Convex Function

Figure 1.5: Some examples of convex function and non-convex function

In other words, every point on the line segment connecting two points within the set lies in the set. Fig.(1.4) show some examples of convex and non-convex sets.

Definition 2. *A function* $f : \mathbb{R}^n \to \mathbb{R}$ *is convex if domain of* f *is a convex set and if* $\forall\, x, y \in$ *domain of* f*, and* ϑ *with* $0 \leq \vartheta \leq 1$*, the following relation hold*

$$f(\vartheta x + (1-\vartheta)y) \leq \vartheta f(x) + (1-\vartheta)f(y) \tag{1.15}$$

More specifically, the line segment joining $(x, f(x))$ and $(y, f(y))$ lies above the graph of function f. The function is strictly convex if the above inequaltiy holds whenever $x \neq y$ and $0 < \vartheta < 1$. Fig.(1.5) show some examples of convex and non-convex functions.

(G) Topological Manifold: A topological n-manifold $(\mathcal{M})$ is a *topological space* which is *locally homeomorphic* to a n-ball $(\mathbb{B}^n)$, where n is a positive integer which is well-defined, and it is the dimension of the manifold. Here, the space $(\mathcal{M})$ is assumed to be Hausdorff and second countable. Fig.(1.6) shows an abstract example of a topological n-manifold.

Topological space: A topological space is a set endowed with the notion of open set and closed set.

Locally homeomorphic: Locally homeomorphic to a n-ball means that every point in the space $(\mathcal{M})$ contained in an open set $\mathcal{O}$ such that, there is a continuous one-to-one onto map $f : \mathcal{O} \to \mathbb{B}^n$.

In this thesis we used a particular class of Riemannian manifold known as Grassmann manifold. A point on the Grassmann manifold $\mathcal{G}(p, d)$ is a linear subspace, which may be specified by an arbitrary orthogonal basis stored as an $p \times d$ matrix [50]. Formally, the Grassmann manifold $\mathcal{G}(p, d)$ is the the space of n dimensional linear subspace of $\mathbb{R}^d$, where,

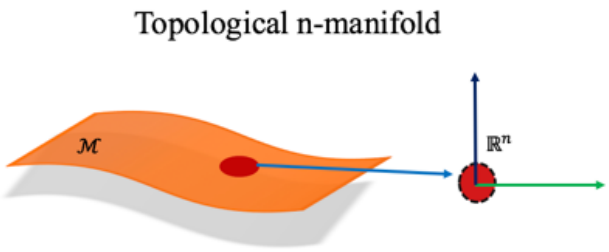


Figure 1.6: Visual intuition of a topological n-manifold. The dotted black-line along the boundaries of the circle denotes that the set is open.

$0 < p \leq d$. Fig.(1.7) show each observation spans a one-dimensional subspace of $\mathbb{R}^2$, therefore, its a point on $\mathcal{G}(1,2)$.

(H) Mathematical Optimization: A mathematical optimization problem has the form

$$\begin{aligned} &\underset{x}{\text{minimize}}\ f_0(x) \\ &\text{subject to } f_i(x) \leq b_i, i = 1,2,....m \end{aligned} \tag{1.16}$$

Here x is the optimization variable of the problem, the f_0 is called the objective function or cost function. The functions f_i's are the constraint functions (may be equality or inequality function) imposed on the optimization variable. The constant b_i's are the bounds for the constraint. The solution to the above cost function is considered optimal, if it has the smallest objective value among all the possible x that satisfy the constraint. For more rigorous and detailed explanation on this, kindly refer to Boyd.S and Vandenberghe.L book on Convex Optimization [20].

(I) Low Rank Approximation Problem: The problem

$$\begin{aligned} &\underset{Y}{\text{minimize}}\ \|X - Y\|_F^2 \\ &\text{subject to}\ \text{rank}(Y) \leq r \\ &\text{where},\ X \in \mathbb{R}^{m \times n} \end{aligned} \tag{1.17}$$

has an analytic solution using singular value decomposition. Let $[U, \Sigma, V^T] = svd(X)$. The rank r solution to this problem can be found by preserving the top r singular values and

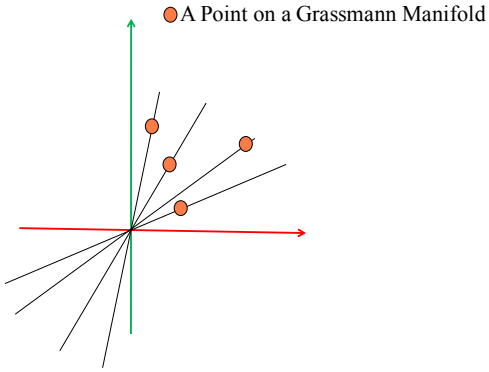


Figure 1.7: Circled point represent a 1-dimensional subspace of $\mathbb{R}^2$ which is a point on Grassmann manifold $\mathcal{G}(1, 2)$.

replacing the remaining singular values by zeros. The result is referred to as the matrix approximation lemma or Eckart–Young–Mirsky theorem [49]. More precisely,

$$Y^* = U\text{diag}(\sigma_1, \sigma_2, ...\sigma_r, 0, 0, ..0)V^T. \tag{1.18}$$

With brief overview on some of the mathematical concept, we will start our discussion on non-rigid structure from motion. I assume the readers are familiar with matrix differentiation. In case you want to revise it, kindly refer to Matrix cookbook [145], I personally found it very handy book for reference.

2

Revisiting simple prior free approach to NRSFM factorization

Contents

In this chapter we start our discussion with one of the classical work done in NRSfM [44]. We detail the problem with the execution of this approach and how it can be improved to perform better on available dataset.

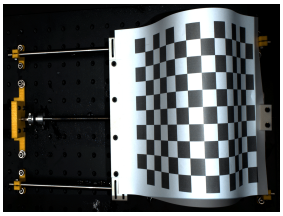
2.1 Why revisting?

A simple prior free factorization algorithm[44] is quite often cited work in the field of Non-Rigid Structure from Motion (NRSfM). The benefit of this work lies in its simplicity of implementation, strong theoretical justification to the motion and structure estimation, and its invincible originality. Despite this, the prevailing view is, that it performs exceedingly inferior to other methods on several benchmark datasets[93, 7]. However, our subtle investigation provides some empirical statistics which made us think against such views. The statistical results we obtained supersedes Dai *et. al.*[44] originally reported results on the benchmark datasets by a significant margin under some elementary changes in their core algorithmic idea[44]. Now, these results not only exposes some unrevealed areas for research in NRSfM but also give rise to new mathematical challenges for NRSfM researchers. In this chapter, we will explore some of the hidden intricacies missed by Dai *et. al.* work[44] and how some elementary measures and modifications can significantly enhance its performance, as high as 18% on the benchmark dataset. The improved performance is justified and empirically verified by extensive experiments on several datasets. We believe this chapter has both practical and theoretical importance for the development of better NRSfM algorithms. Practically, it can also help improve the recently reported state-of-the-art [107, 93] and other similar works in this field which are inspired by Dai *et al.* work[44].

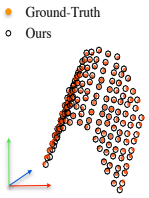
2.2 Introduction

Notation: *For consistency and ease of understanding to the readers, the notation we used in this paper is similar to Dai et al. work [44] unless otherwise stated. We assume that the reader is familiar with Dai et. al. work [44].*

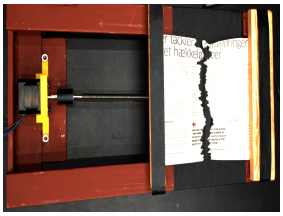
A simple prior-free method for computing non-rigid structure from motion (NRSfM) introduced by Dai *et al.* is now considered as a classical work in NRSfM [44]. In their work, the camera motion is estimated by imposing the null space constraint and the rank-3 positive semi-definite matrix cone constraint on the Gram matrix (Q_k). Further, nuclear norm minimization of the reshuffled shape matrix ($S^\sharp$) was introduced to proffer stronger rank bound on the shape matrix for non-rigid shape estimation. The striking part of their work is that



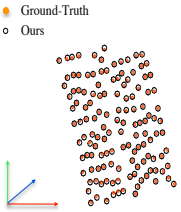
(a) Paper Sequence



(b) Paper 3D Reconstruction



(c) Tearing Sequence



(d) Tearing 3D Reconstruction

Figure 2.1: The method recovers 3D dimensional structure of the deforming object over multiple frames. Our elementary but powerful changes provides a substantial improvement in the reconstruction accuracy than the previous results reported for "prior-free" approach. The example images are taken from the recently released NRSfM Challenge Dataset [93]. Our reconstruction results are nearly as good as the best performing algorithm without using very complex and involved mathematical optimization [107].

it not only challenged the myth of the inherent basis ambiguity in NRSfM [196] but also supplied a practical "prior-free" algorithm to solve NRSfM.

The elementary idea of Dai *et al.* [44] work conveniently encapsulates all the basic intuitions which are required to solve a general NRSfM problem. One may immediately argue on its performance when the deforming shape is composed of a union of low-rank subspace [107, 205]. However, in this chapter, we restrict our discussion to the classical representation of a NRSfM problem [22], without paying much attention to, how clustering benefits 3D reconstruction of the non-rigid object and other such notions of compact data representation. The reason for this choice is that the improvement in the performance of a classical baseline will automatically benefit the methods built on top of it [107, 205].

The main purpose of this chapter is to uncover some of the unexplored mathematical intricacies in the prior free factorization approach to NRSfM and improve on the idea supplied by Dai *et al.*[44]. Our exposition leads to the possible reason for its inferior performance on the benchmark datasets [7, 93, 175]. It is shown in this chapter that the rotation estimate using Dai *et al.*work [44] is *not unique* under the same model complexity prior (K/rank), and they overlooked to utilize the well-known assumption of *smooth* non-rigid deformation of the object[148]. A simple search for the proper column-triplet for the correction matrix (G_k) based on the smoothness of camera motion can indeed help improve the accuracy of the algorithm. Further, we argue that the weighted nuclear norm minimization of the shape matrix

($S^\sharp$) is a far better choice than its global trace norm minimization. Lastly, due to our extensive analysis, we are able to posit some unsolved issues in NRSfM under "prior-free" idea which needs attention for further progress in this field.

It is not claimed that we achieve state-of-the-art results on the benchmark datasets using our new approach. However, we empirically show that we can get very close to the best performing approaches and the difference is not very great, without the employment of complex and involved mathematical optimization [107, 119]. In this chapter, we also argue that the inferior performance of "prior-free" method may not be due the proposed algorithmic idea but because they overlooked some of the mathematical construction in their own formulation, and missed on properly utilizing the well-known assumptions about non-rigidly moving object *i.e.*, *smooth deformation*[148] and *low-rank shape* [44]. Hence, the conclusion, understanding and use of simple "prior-free" algorithm to NRSfM is not complete and precise. This chapter try to amend and nullify the prevailing perception about the "prior-free" approach, and how it can be used to its maximum potential. We feel that our work touches some critical points which are essential to establish a theoretical closure to some of the elementary problems within the factorization approach to NRSfM.

Contribution: Firstly, this work postulates some rectification to the usage of "Intersection Method" [44] to compute camera motion. With the suitable example, we establish that the generalization made on the rotation matrix estimation by Dai *et al.* work [44] is *not convincing* and therefore, the knowledge about the strength of "Intersection theorem" is not completely exploited. Secondly, we provide an analytic solution to estimate suitable rotation using Intersection theorem and conjecture some challenges associated with it. Lastly, we propose a weighted nuclear norm minimization problem to estimate non-rigid 3D shape. Our approach shows a substantial improvement in the 3D reconstruction accuracy (as high as 17.6%). We also observed improvement in the performance of the algorithm in the presence of noisy data §2.5 and missing data §2.5 (with a minor adjustment).

In this work, our attempt is to make the baseline method* more accurate, both in terms of understanding and performance, subject to the mathematical simplicity. To achieve this, we attempt to avoid the usage of complex mathematical notions such as union of independent subspace, dependent subspace representation [205, 107, 118], procrustean normal distribution [119], kernelization [72] *etc.* Hence, it is simple to understand the theoretical and practical justification of our method. We show that by applying simple but powerful logical and mathematical modifications to prior free idea [44], we can get close to or even perform better at times than the best algorithms on the benchmark datasets. Additionally, our approach

*By baseline, we mean the methods that solve NRSfM using its classical representation $W = RS$ that have withstood the test of time [170, 22].

shall help improve the other state-of-the-art methods built on top of the targeted baseline [44].

2.3 Classical Representation

Tomasi and Kanade factorization method to structure-from-motion under orthographic camera projection appropriately summarizes the behavior of the 3D points over frames [170]. The relation between 3D shape, motion and its projection over frames was defined as

$$W = RS \tag{2.1}$$

where, $W \in \mathbb{R}^{2F \times P}$ is the measurement matrix formed by stacking all the image coordinates ($x = [u, v]^T$) for 'P' points along 'F' rows *i.e.*, total number of frames. $R = \text{blockdiagonal}(R_1, R_2, .., R_F) \in \mathbb{R}^{2F \times 3F}$ denotes the orthographic camera rotation matrix with each $R_i \in \mathbb{R}^{2\times 3}$ as per frame rotation. $S \in \mathbb{R}^{3F \times P}$ represent the shape matrix with each row triplet as a 3D shape. This representation was later extended by Bregler *et al.* [22] to recover non-rigid 3D shapes. More concretely,

$$\begin{aligned} W &= \begin{bmatrix} x_{11} \dots x_{1P} \\ \dots \\ x_{F1} \dots x_{FP} \end{bmatrix} = \begin{bmatrix} R_1 S_1 \\ .. \\ R_F S_F \end{bmatrix} = \begin{bmatrix} c_{11} R_1 \dots c_{1K} R_1 \\ \dots \\ c_{F1} R_F \dots c_{FK} R_F \end{bmatrix} \begin{bmatrix} B_1 \\ .. \\ B_K \end{bmatrix} \\ &\Rightarrow W = R(C \otimes I_3)B = \Pi B \end{aligned} \tag{2.2}$$

The matrix 'B' and 'C' are composed of shape bases and shape coefficients respectively, with 'K' as the number of shape bases. '$\otimes$' denotes the kronecker product and 'I_3' is a 3×3 identity matrix. It is evident from the above formulation that the rank of $W \leq 3K$ and also $\text{rank}(S) \leq 3K$. However, S is not a general rank $3K$ matrix but own a special structure due to $C \otimes I_3$ factor [44].

2.3.1 Null Space Representation to Orthonormality Constraint

An initial step in the factorization approach to NRSfM is to perform a rank $3K$ decomposition of the measurement matrix W via singular value decomposition (svd) *i.e.*$W = \hat{\Pi}\hat{B}$. This is then followed by the estimation of Euclidean corrective matrix 'G' to solve rotation and 3D structure. The main reason for such a procedure is due to the fact that the singular value decomposition of 'W' matrix is not unique as any non-singular matrix $G \in \mathbb{R}^{3K \times 3K}$ in between the two matrices $\hat{\Pi}$ and $\hat{B}$ can form a valid factorization. Mathematically,

$$W = \hat{\Pi}\hat{B} = (\hat{\Pi}G)(G^{-1}\hat{B}) = \Pi B \tag{2.3}$$

Now, once we are able to solve G correctly, then rotation and shape can be estimated using the above relations [22]. To solve G, orthonormality constraints are imposed *i.e.*$R_i R_i^T = I_2$. Representing the i^{th} double row of $\hat{\Pi}$ as $\hat{\Pi}_{2i-1:2i} \in \mathbb{R}^{2\times 3K}$ and $G_k \in \mathbb{R}^{3K\times 3}$ as the k^{th} column triplet of G, then using Eq:(2.2) and Eq:(2.3) we can write

$$\hat{\Pi}_{2i-1:2i} G_k = c_{ik} R_i, \forall\, i = \{1, 2, .., F\}, k = \{1, 2, .., K\} \tag{2.4}$$

Multiplying both sides by R_i^T from right side gives

$$\begin{aligned} &\hat{\Pi}_{2i-1:2i} G_k G_k^T \hat{\Pi}_{2i-1:2i}^T = c_{ik}^2 I_2 \\ &\text{This leads to two linear equation constraint} \\ &\hat{\Pi}_{2i-1} Q_k \hat{\Pi}_{2i-1}^T = \hat{\Pi}_{2i} Q_k \hat{\Pi}_{2i}^T \\ &\hat{\Pi}_{2i-1} Q_k \hat{\Pi}_{2i}^T = 0 \end{aligned} \tag{2.5}$$

where, $Q_k \in \mathbb{R}^{3K\times 3K} = G_k G_k^T$. Using the algebraic relation $\text{vec}(AXB^T) = (B \otimes A)\text{vec}(X)$, Dai *et al.* transformed these constraints (Eq:2.5) to a null space representation as follows:

$$\begin{bmatrix} \hat{\Pi}_{2i-1} \otimes \hat{\Pi}_{2i-1} - \hat{\Pi}_{2i} \otimes \hat{\Pi}_{2i} \\ \hat{\Pi}_{2i-1} \otimes \hat{\Pi}_{2i} \end{bmatrix} \text{vec}(Q_k) = \mathcal{A}\text{vec}(Q_k) = 0 \tag{2.6}$$

Using the above form and previous work in NRSfM [196], Dai *et al.* proposed the *intersection theorem* and supplied a SDP solution to estimate the Q_k matrix and the Euclidean corrective matrix G_k using $\text{svd}()$.

Theorem 2.3.1. *Intersection Theorem: Under non-generate and noise-free conditions, any correct solution of Q_k must lie in the intersection of the $(2K^2 - K)$ dimensional null-space of $\mathcal{A}$ and a rank 3 positive semi-definite matrix cone i.e.Q_k must belong to*

$$\{\mathcal{A}\text{vec}(Q_k)\} \cap \{Q_k \succeq 0\} \cap \{\text{rank}(Q_k) = 3\} \tag{2.7}$$

2.3.2 Dai *et al.* solution to rotation

They proposed that once the Q_k is solved, rather than solving for full Euclidean corrective matrix $G \in \mathbb{R}^{3K\times 3K}$, use $\text{svd}()$ to extract rank 3 G_k. The solved $G_k \in \mathbb{R}^{3K\times 3}$ can then be use to find R (Eq:2.4) up to sign (c_{ik}). The method quote "we adopt a simpler approach that directly computes the camera motion R from single column-triplet G_k without need to fill in a big and full G matrix". Naturally, this single column-triplet is chosen to be the first column-triplet (G_1) of the G matrix (see Fig:2.2). Now, such strategy give rise to few legitimate *concerns*

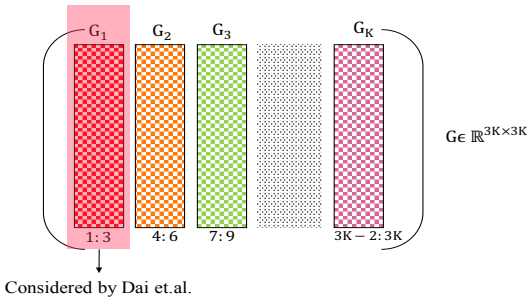


Figure 2.2: (a) The column triplet (1:3) of euclidean corrective matrix (G_k) used by Dai *et al.*work [44] shown in red shade. It is stated with the notion that there is no loss of generality to choose G_1. However, choosing other column triplet may result in better rotation and shape estimate as shown in Figure 2.4a and 2.4b

(a) When each column triplet $\{G_i\}_{i=1}^{K}$ qualifies for a suitable correction matrix, then why G_1 has a high preference? Are we loosing useful information by such unwarranted preference? Whether such solution to rotation caters the assumption of smooth deformation?
(b) Will each $\{G_i\}_{i=1}^{K}$ provide the same solution to the rotation matrix?

Dai *et al.*overlooked all these intrinsic issues with their approach to obtain rotation.

2.3.3 Plausible Rectification

Our experiment show that Dai *et al.*[44] solution to rotation estimation actually aborted the useful information present in the $G \in \mathbb{R}^{3K\times 3K}$. Each of the '$K$' column triplets in G (*i.e.*G_k) gives a possible rotation matrix which is different from each other (see Fig:(2.3)). Our empirical evaluations on several datasets show that the first column triplet is *not* always the best choice to estimate rotation. Hence, the details provided by Dai *et al.* work [44] is *incomplete* and there is a *loss of generality* with such procedure to estimate rotation under the well-known assumption of smooth deformation [148]. Fig:(2.4a) and Fig:(2.4b) provides few statistical results with comparison for both rotation and shape error estimate respectively. For clarity, we also provide the column triplet index that gives the better results for the corresponding data sequence and therefore, provides few counter-examples to such generalization.

Theoretically and practically, this result is of significant importance as it helps in inferring that the solution provided by "Intersection Theorem" has a lot of useful information left to be exploited completely and Dai *et al.* work ignored this. Also, it gives rise to some challenges that finding the best column triplet for G_k is not an easy task. With these results, we posit few propositions for further research in NRSfM that are: (a) Can we find a best possible column triplet for the corrective matrix with a given rank prior '(K)', or (b) At least can we put an upper bound on the value $k \subset K$ such that there exists no such 'k' for G_k which will

provide better rotation and structure estimate (c) Upper bound on the value of 'K' which can guarantee a smooth solution. The problem seems hard keeping in view that the prior rank (K) in NRSfM factorization methods is an assumed approximation and it changes for different datasets to achieve better results.

2.3.4 A solution to motion

In this work, we use an analytical observation based on the smoothness and regularity[†] of the camera trajectory to filter G_k to infer better rotation matrix. Let $\psi(.)$ be a function that takes G_k as input and gives R as output using Intersection Theorem. We estimate different $R \in \mathbb{R}^{2F\times 3F}$ for all the column triplets *i.e.*$\{G_k\}_{k=1}^{K}$, then computed the smoothness of the camera motion '$\eth_f$' for each G_k as:

$$\begin{aligned} &\text{Suppose, } R = \psi(G_k), \text{ via Intersection method, then,} \\ &\eth_f = \|R_f - R_{f+1}\|_F^2 \ \forall f = 1, 2, \ldots, F-1. \ [84] \text{ Sec.4.} \end{aligned} \tag{2.8}$$

By examining the smoothness of the camera motion for each G_k, we select the suitable rotation matrix for structure estimation (see Fig. 2.3). Our strategy to select smooth camera motion over frames based on Eq:(2.8) consistently supplied us with better performance than the previously proposed approach. We acknowledge that this is not a profound way to infer the best rotation, however, it does provide a possibility to deduce better rotation using "prior-free" approach which respects the well-known assumption of smooth deformation in NRSfM. Further, it helps endorse our claim on the generalization of rotation estimate by [44]. You may use variable '$\eth_f$' Eq:(2.8) as a smoothness term in the final optimization (Eq:(2.11)) to further improve rotation, however, to show the competence within the "prior-free" framework [44], we stick to the classical way.

2.4 Structure Estimation

Once the rotation is estimated based on the smoothness of the camera motion, the next step is to solve for 3D structure. The block matrix method (BMM) by Dai *et al.*[44] proposed the following optimization problem to estimate the non-rigid low-rank shape.

$$\begin{aligned} &\underset{S^\sharp}{\text{minimize}} \ \|S^\sharp\|_* \\ &\text{subject to:} \\ &W = RS, S^\sharp = g(S) \end{aligned} \tag{2.9}$$

[†]The term <<regularity>> is used in a loose sense (Mathematically).

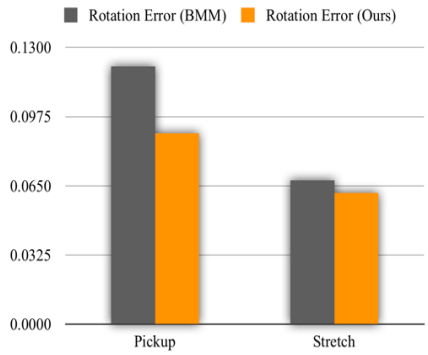
(a) $R \leftarrow G_1$ **(b)** $R \leftarrow G_2$ **(c)** $R \leftarrow G_3$

(d) $R \leftarrow G_4$ **(e)** $R \leftarrow G_5$ **(f)** $R \leftarrow G_6$

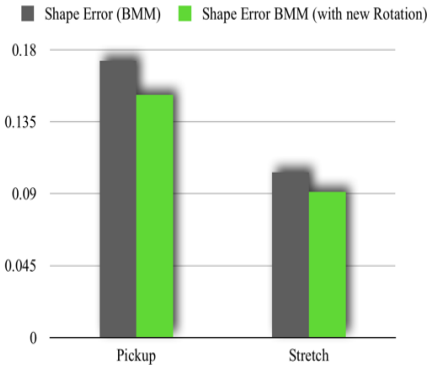
(g) $R \leftarrow G_7$ **(h)** $R \leftarrow G_8$ **(i)** $R \leftarrow G_9$

(j) $R \leftarrow G_{10}$ **(k)** $R \leftarrow G_{11}$ **(l)** $R \leftarrow G_{12}$

Figure 2.3: The rotation samples on $\mathbb{SO}(3)$ using $\{G_k\}_{k=1}^{12}$ for *Pick-up sequence*. Below each $\mathbb{SO}(3)$ manifold is the graph showing the per frame change in the camera motion using Eq:(2.8) '∂_f'. A simple observation establishes that all rotation matrix (R) are not the same. '∂_f' graph analysis on this dataset show that the rotation estimate provided by G_7, G_8, G_9 has a smoother camera motion than other G_k's, with G_9 being the smoothest. Any one out of these 3 G_k's supply better performance than G_1. Note: Each $R_i \in \mathbb{R}^{2\times 3} \mapsto R_i \in \mathbb{R}^{3\times 3}$ via cross product. (Best viewed on screen)



(a) Motion Improvement



(b) Shape Improvement

Figure 2.4: Few counter examples on benchmark dataset [8]. (a) Rotation error in comparison to BMM [44] on synthetic data. (b) 3D reconstruction error using global trace norm minimization of shape matrix as used in BMM with rotation matrix estimate using other column triplet in comparison to $G(1{:}3)$. The column triplets of (G) for which the method perform better on Pickup and Stretch sequence are (19:21), and (19:21) respectively. Note that we used the same rank prior value 'K' used in Dai *et al.*work [44].

where, $S^\sharp \in \mathbb{R}^{F\times 3P}$ is a rearranged shape matrix with each row corresponds to the shape for that frame. The trace norm minimization on '$S^\sharp$' is enforced instead of 'S' to provide a stronger rank bound on the shape matrix [44]. The second term in Eq:(2.9) enforces the re-projection error constraint. The function $g(.)$ maps $S \in \mathbb{R}^{3F\times P}$ to $S^\sharp \in \mathbb{R}^{F\times 3P}$.

2.4.1 Dai *et al.* solution to shape

Following the work of Ma *et al.*[129] on rank minimization problems, Dai *et al.*[44] proposed a solution to the optimization in Eq:(2.9). The method enforces low-rank constraint on '$S^\sharp$' matrix and provide the solution by solving Eq:(2.9) via ADMM[19] using matrix shrinkage operator $\mathcal{S}[\lambda](X) = U\text{diag}(s[\lambda](\sigma))V^T$, where $s[\lambda](\sigma) = \bar{\sigma}$ with $\bar{\sigma}_i = \{\sigma_i - \lambda \text{ if } \sigma_i - \lambda > 0$ and '0' otherwise $\}$.

2.4.2 Plausible Rectification

Despite the trace norm minimization provides a satisfactory solution to non-rigid structure estimation, it has some serious issues. The proposed solution to nuclear norm minimization gives equal priority to each singular values, as a result, the shrinkage operator penalizes each singular value with the same quantity (λ). For a non-rigid object, we always have this prior assumption that the shape lies in a low-rank subspace, therefore, it's not a better choice to penalize the major component of the shape data and its very minor component equally. Conse-

quently, nuclear norm minimization of the shape matrix struggles to appropriately conserve the useful component of the non-rigidly deforming shape.

Truncated nuclear norm regularization can be a choice to handle such issues, however, it depends on the binary decision, hence not versatile in nature [203]. To really cater the behavior of the deformations based on its low-rank nature, we propose to use weighted nuclear norm minimization approach to solve for non-rigid structure [162, 79]. In contrast to the previous notation to the nuclear norm of the shape matrix *i.e.*$\|S^\sharp\|_*$, we introduce a new notation for its weighted nuclear norm

$$\|S^\sharp\|_{\Theta,*} = \sum_{j=1}^{K} \Theta_j \sigma_j(S^\sharp) \tag{2.10}$$

where $\sigma_j(.)$ denotes the j^{th} singular value of $S^\sharp$. We assume that the weights Θ_j's are non-negative scalar *i.e.*$\Theta_j \geq 0$. Using this representation, we redefine the optimization proposed in the Eq:(2.9) as follows:

$$\begin{aligned} &\underset{S^\sharp, S}{\text{minimize}}\ \mu\|S^\sharp\|_{\Theta,*} + \frac{1}{2}\|W - RS\|_F^2 \\ &\text{subject to: } S^\sharp = g(S) \end{aligned} \tag{2.11}$$

The motivation for such formulation is quite clear, however, the proposed optimization (Eq:2.11) is generally non-convex, and is more difficult to solve than the nuclear norm minimization. Fortunately, recent results [202, 128, 79] in compressed sensing have shown that we can achieve a global optimal solution to Eq:(2.11) in the case when $0 \leq \Theta_1 \leq \Theta_2 \leq .... \leq \Theta$ §2.4.3

2.4.3 A solution to shape

This section provides the mathematical derivation to the optimization proposed in Eq:(2.11). Our solution use the following theorems and proofs as stated and used in [202, 79, 27].

Theorem 2.4.1. *For all $Y \in \mathbb{R}^{m\times n}$, denoted by $Y = U\Sigma V^T$, the SVD of it. The solution to minimize$_X\|Y - X\|_F^2 + \|X\|_{\Theta,*}$, with non-negative weight vector Θ, its solution $\hat{X}$ can be written as $\hat{X} = U\hat{B}V^T$, where $\hat{B}$ is the solution to the following optimization problem*

$$\hat{B} = argmin_B\|\Sigma - B\|^2 + \|B\|_{\Theta,*} \tag{2.12}$$

Theorem 2.4.2. *If the singular values $\sigma_1 \geq .... \geq \sigma_K$ and the weights satisfy $0 \leq \Theta_1 \leq \Theta_2 \leq .... \leq \Theta_K$ then the weighted nuclear norm minimization problem minimize$_X\|Y - X\|_F^2 + \|X\|_{\Theta,*}$ has a globally optimal solution*

$$\hat{X} = U\mathcal{S}_\Theta(\Sigma)V^T \tag{2.13}$$

where $Y = U\Sigma V^T$ is the SVD of Y, and $\mathcal{S}_\Theta(\Sigma)$ is the generalized soft-thresholding operator with weight vector Θ

$$\mathcal{S}_\Theta(\Sigma) = max(\Sigma_{ii} - \Theta_i, 0) \tag{2.14}$$

The readers are encouraged to refer to [202, 79] work for detailed derivations to the lemma's leading to the proof of the theorems. In conclusion, if the weights satisfies non-descending order, not necessarily with the same value, the weighted nuclear norm minimization problem is still convex and optimal solution can be obtained using a soft-thresholding operator with different weights [202, 79].

Optimization: We propose our solution to the optimization problem defined in Eq:(2.11) using alternating direction method of multipliers [19] (ADMM), a simple, fast but powerful algorithm used to solve many convex and non-convex problems in computer vision and mathematical optimization. The ADMM algorithm decomposes the original problem into several sub-problems, where each of them is solved separately by introducing Lagrange multipliers and penalty parameters to estimate convergence. Using the method of multipliers, the Augmented Lagrangian form for Eq:(2.11) is written as follows:

$$\begin{aligned}\mathcal{L}_\varrho(S^\sharp, S) = \mu\|S^\sharp\|_{\Theta,*} + \frac{1}{2}\|W - RS\|_F^2 + \frac{\varrho}{2}\|S^\sharp - g(S)\|_F^2 + \\ < Y, S^\sharp - g(S) >\end{aligned} \tag{2.15}$$

here $Y \in \mathbb{R}^{F\times 3P}$ is a Lagrange multiplier and $\varrho > 0$ is the penalty parameter. The solution to each variable is obtained by solving the following subproblems over iterations (indexed with the variable i):

$$(S^\sharp)^{i+1} = \underset{S^\sharp}{\operatorname{argmin}}\ \mathcal{L}_\varrho\big((S^\sharp)^i, S\big) \tag{2.16}$$

$$(S)^{i+1} = \underset{S}{\operatorname{argmin}}\ \mathcal{L}_\varrho\big(S^\sharp, (S)^i\big) \tag{2.17}$$

The Lagrange multiplier and the penalty parameter are updated as follows:

$$\begin{aligned}Y &= Y + \varrho(S^\sharp - g(S)) \\ \varrho &= \text{minimum}(\varrho_{\text{max}}, \lambda\varrho)\end{aligned} \tag{2.18}$$

ϱ_{max} refers to the maximum value of 'ϱ' and λ is an empirical constant ($\lambda > 1$). The mathematical derivations to each sub-problems are provided in the Appendix (A) for reference. The closed form solution to the Eq:(2.17) is obtained by taking the derivative of Eq:(2.15) w.r.t variable 'S' and equating it to zero *i.e.*,

$$S = \Big(\frac{\varrho I + R^T R}{\varrho}\Big)\backslash\Big(\big(g^{-1}(S^\sharp) + \frac{g^{-1}(Y)}{\varrho}\big) + \frac{R^T W}{\varrho}\Big) \tag{2.19}$$

Note '\' is a Matlab slang *i.e.*if $\mathcal{A}x = B$ implies $x = \mathcal{A}\backslash B$. Similarly, rewriting the Eq:(2.15) treating $S^\sharp$ as variable.

$$= \underset{S^\sharp}{\operatorname{argmin}}\, \mu\|S^\sharp\|_{\Theta,*} + \frac{\varrho}{2}\|S^\sharp - g(S)\|_F^2 + < Y, S^\sharp - g(S) > \tag{2.20}$$

In contrast to the previous form, the solution to Eq:(2.20) is not straight forward. To obtain a closed form solution to this problem, lets define a soft-thresholding function $\mathcal{S}[\tau](\sigma) = \text{sign}(\sigma).\max(|\sigma| - \tau, 0)$. Also, let $[U, \Sigma, V]$ be the singular value decomposition of $(g(S) - Y/\varrho)$, then the optimal solution to Eq:(2.20) is given by:

$$S^\sharp = U\mathcal{S}[\frac{\Theta\mu}{\varrho}](\Sigma)V \tag{2.21}$$

Here, Θ is the weight assigned to the different singular values in the non-descending order based on its significance to the deformation data. For detail discussion on the initialization of weights refer section §2.5 (2). Its important to note that the ADMM based solution to our optimization problem Eq:(2.15) gives us a satisfactory solution (near optimal) which may not be globally optimal.

2.5 Experiment and Discussion

To endorse our claim, we performed extensive experiments on real and synthetic benchmark datasets [7, 93, 175]. We compared the performance of our algorithm against different state-of-the-art methods on these datasets [73, 119, 107]. Additionally, we unveil the substantial percentage boost in the reconstruction accuracy as high as 18% in comparison to the previous results reported for "simple prior-free" approach. For real-world applications to NRSfM, noisy data and missing feature tracks over frames are crucial, therefore, we also performed experiments to tackle such issues. Before we provide details on the performance analysis, we discuss the variable initialization.

Initialization: Our algorithm has few parameters and variables to initialize. For all our experiments on different datasets, we initialize $\mu = 1$, $\lambda = 1.1$, $\varrho_{\max} = 1e^{10}$, $\varrho = 1e^{-4}$, $Y = \text{zeros}(F, 3P)$ and the 'K' values are kept same as Dai *et al.* method[44]. Practically, we considered the convergence of our optimization, if the gap $\max\|(S^\sharp - g(S))\|_\infty < 1e^{-8}$ or $\varrho > \varrho_{\max}$ over iteration.

1. Structure initialization: Using the result of Liu *et al.* [124] on the uniqueness of minimizer for the rank minimization problem, we initialize the the 3D shape 'S' as 'S' = $\text{pinv}(R)W$ and $S^\sharp = g(S)$. The pseudo-inverse solution to shape matrix provides a good enough initialization to our algorithm. Reader may refer to Dai *et al.* [44] and Valmadre *et al.* [180] work

for detailed discussion on the uniqueness and planarity of pseudo inverse solution to 'S' in NRSfM.

2. Weight (Θ) initialization: It is well-known in NRSfM under factorization approach that the shape matrix lies in a low-rank space. Generally, the largest singular value of the shape matrix contains the most information about the non-rigid shape, therefore, while optimizing for the shape matrix, it's illogical to treat each singular value equally. The singular values with major component must be penalized less and vice-versa. Using this inverse relation between singular values and its significance to the shape deformation modeling, we assign the weight (Θ) to be inversely propositional to the singular values of the shape matrix.

$$\Theta_j = \frac{\xi}{\sigma_j(S^\sharp) + \gamma} = \frac{\xi}{\sigma_j(g(S)) + \gamma} \tag{2.22}$$

where, ξ is a positive number and $\gamma = 1e^{-6}$, a very small positive number to avoid division by zero as some singular values are likely to be zero (low rank). We initialized the weights by substituting the pseudo-inverse initialization of '$S^\sharp$' *i.e.* using the relation $S^\sharp = g(S)$ §2.5 in the Eq:(2.22).

Performance Analysis After a detailed discussion on the variable initialization, we are ready to present our experimental evaluation. We performed extensive experiments on both new and previously released benchmark datasets [7, 175, 93]. We report the quantitative result on the previous benchmark dataset using mean normalized 3D reconstruction error formulation *i.e.*

$$e_s = \frac{1}{F} \sum_{i=1}^{F} \frac{\|S^i_{\text{est}} - S^i_{GT}\|_F}{\|S^i_{GT}\|_F} \tag{2.23}$$

where, S_{est}, S_{GT} are the estimated 3D shape and ground-truth 3D shape respectively. To keep our statistics consistent with the newly proposed NRSfM challenge dataset, we used their error evaluation code to compute the robust root mean square error (RMSE) metric as proposed in Taylor *et al.* work [166]. For more details on NRSfM CVPR 2017 challenge dataset evaluation metric, kindly refer to Jensen *et al.* work[93].

1). Benchmark datasets: Most of the methods proposed in non-rigid structure from motion often use it to evaluate the performance of the algorithm. Loosely speaking, this dataset is composed of eight standard sequences namely Drink, Pickup, Yoga, Stretch, Dance, Walking, Face and Shark. The number of frames (F) to number of points (P) *i.e.*(F, P) set for these datasets are (1102, 41), (357, 41), (307, 41), (370, 41), (264, 75), (316, 40) and (240, 91) respectively. Table (2.1) show the statistical comparison of our approach in comparison to the other competing approaches for single body NRSfM. Our evaluation results clearly present

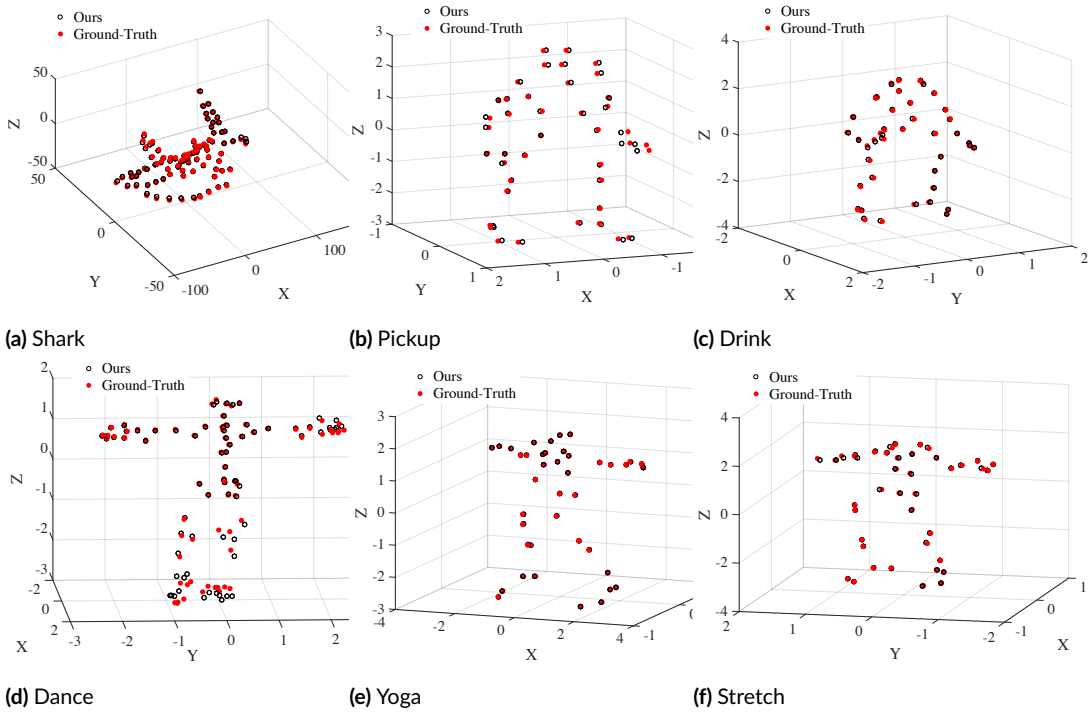
Dataset/Method	PTA	CSF2	PND	BMM	Ours
Drink	0.0287	0.0227	0.0037	0.0266	0.0119 (1.470%)
Pickup	0.1939	0.1791	0.0372	0.1731	0.0198 (15.33%)
Yoga	0.1243	0.1179	0.0140	0.1150	0.0129 (10.21%)
Stretch	0.1035	0.1136	0.0156	0.1034	0.0144 (8.900%)
Dance	0.2426	0.1877	0.1454	0.1864	0.1060 (8.040%)
Walking	0.3761	0.1938	0.0465	0.1298	0.0882 (4.160%)
Face	0.0489	0.0319	0.0165	0.0303	0.0179 (1.240%)
Shark	0.2933	0.1117	0.0135	0.2311	0.0551 (17.60%)

Table 2.1: Performance comparison in the shape recovery using our new approach with some of the state-of-the-art methods in single body NRSfM. The statistics clearly demonstrate our claim that we can achieve a significant improvement in the reconstruction accuracy without using complex mathematical formulation. The percentage value in the last column (blue) show the improvements over the result documented by Dai *et al.* original work (BMM) [44].

a significant improvement in the reconstruction accuracy in comparison to the previously reported results for "prior-free" approach. Figure (2.5) show the qualitative reconstruction results w.r.t ground-truth on all of these sequences.

2). NRSfM challenge datasets: Jensen *et al.* recently proposed this dataset as a part of NRSfM competition held at CVPR 2017[93]. This is a high quality challenging dataset divided into five categories based on the deformation type, namely, Articulated, Balloon, Paper, Stretch and Tearing. Each of these categories is again shot using six different camera paths namely circle, flyby, line, semi-circle, tricky and zig-zag. This dataset is significantly larger and diverse to really test the performance of a NRSfM algorithm's. However, the dataset provides only a single frame ground-truth 3D for each of the five categories to test the algorithm. To estimate the reliability of our approach, we compared our performance against the best performing algorithm on this dataset. Table (2.2) show the quantitative results of our method. The performance clearly demonstrates the significant improvement in the accuracy using "prior-free" idea under our modification. It also help infer that without using complex mathematical notions, we can reach performance accuracy close to the state-of-the-art. Figure (2.6) show some qualitative results using our method.

3). Noisy data: The feature tracks captured from a real-world motion capture system is noisy most of the time. Therefore, to test the reliability and robustness of our new approach, we performed experiments by re-synthesizing the trajectories added with Gaussian noise. We introduced the Gaussian noise with standard deviation set as $\sigma_{\text{noise}} = r * \max\{|W|\}$, where r is varied from 0.05-0.25. Figure (2.7a) shows the variation in the normalized average 3D error



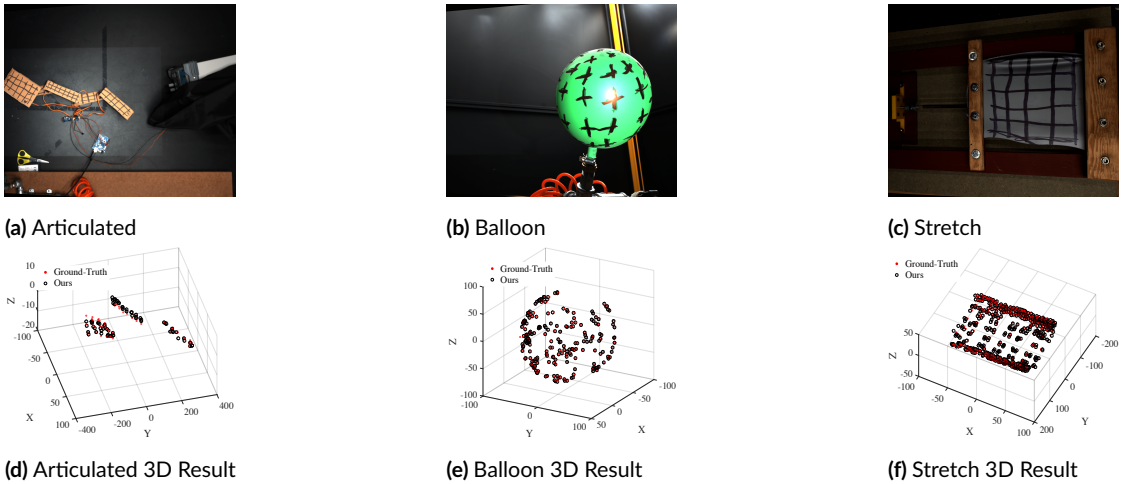
(a) Shark **(b)** Pickup **(c)** Drink

(d) Dance **(e)** Yoga **(f)** Stretch

Figure 2.5: Reconstruction results of our method on the NRSfM synthetic benchmark dataset [7, 8]. Ground-truth and reconstructed points are shown in filled(red) and non-filled circles respectively. Note: We used the same 'K' value as in [44] work for all the experiments.

Method / Data	Articulated	Ballon	Paper	Stretch	Tearing
Multi-body [107]	10.15	10.64	15.78	9.96	14.17
BMM [44]	24.54	12.91	22.37	18.71	18.87
Ours	12.02	11.79	16.21	12.05	16.08

Table 2.2: Performance comparison of our method in comparison to the best performing algorithm (Multi-body) [107] on NRSfM challenge dataset [93]. The above statistics shows the average root-mean-square error in millimeters for the single test image on the orthogonal sequence available with the dataset. Our method shows a clear improvement over the originally proposed BMM approach and it's accuracy got very close to the multi-body. The statistics of multi-body [107] is taken from it's public presentation slides.



(a) Articulated **(b)** Balloon **(c)** Stretch

(d) Articulated 3D Result **(e)** Balloon 3D Result **(f)** Stretch 3D Result

Figure 2.6: Reconstruction results of our method on the NRSfM challenge dataset [93]. The results shown here are for the circular camera path. Ground-truth 3D and reconstructed 3D points are shown with filled and non-filled circles respectively.

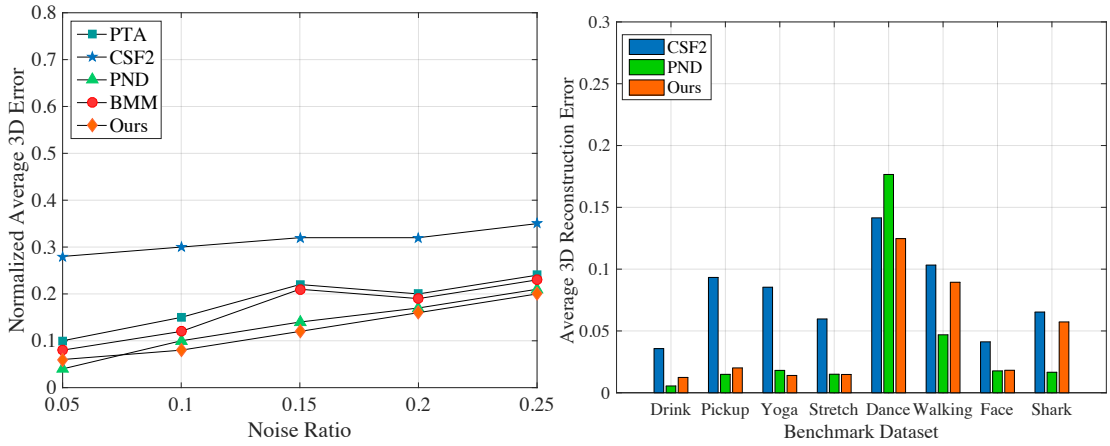
for the stretch sequence using the performance of different algorithm recorded over 20 times. The plot clearly shows the robustness of our algorithm in comparison to other methods in the presence of large noise ratio's.

4). Missing Data: In addition to the noisy data, the other problem with 3D reconstruction from a real video sequence is the missing trajectories over frames. We handle the missing trajectory quite robustly by incorporating a simple modification to the optimization proposed in Eq:(2.11). Let's assume $\tilde{W} \in \mathbb{R}^{2F\times P}$ is the incomplete measurement matrix and $\mathcal{M} \in \{0,1\}$ is the mask matrix which indicates the presence or absence of the tracks over frames. Given $\tilde{W}$, $\mathcal{M}$, we first find a complete W matrix using the following optimization

$$\underset{W}{\text{minimize}}\ \frac{1}{2}\|\mathcal{M}\odot(\tilde{W}-W)\|_F^2,\ \text{subject to: rank}(W)\leq 3K \tag{2.24}$$

The above optimization is a well studied optimization form. To keep things simple, we used Cabral *et al.*work [26] to estimate W. The motive is to first solve for complete 'W' to estimate camera motion using our rectified approach §2.3.1, and then solve for shape using the following cost function:

$$\begin{aligned}&\underset{S^\sharp,S}{\text{minimize}}\ u\|S^\sharp\|_{\Theta,*}+\frac{1}{2}\|\mathcal{M}\odot(\tilde{W}-RS)\|_F^2\\&\text{subject to: } S^\sharp=g(S)\end{aligned} \tag{2.25}$$



(a) Performance on Noisy Trajectory **(b)** Performance on Missing Data

Figure 2.7: (a) 3D reconstruction error comparison over noisy trajectories. (b) Comparison of our method performance with other competing methods with missing data in the measurement matrix.

Clearly, it's just a minor adjustment to the proposed method based on the kind of data available in different situations. To evaluate our performance, we randomly set 30% of the data missing from the sequence same as Lee *et al.*work [119]. Figure (2.7b) shows the performance of our algorithm with missing data.

Discussion:

1. In some applications, we have more prior knowledge about the shape in addition to its low-rank matrix assumption, for example: exact rank of the clean shape matrix. In such cases, one may choose to minimize partial sum minimization of singular values optimization *i.e.*,

$$\begin{aligned} &\underset{S^\sharp, S}{\text{minimize}}\ \mu|\text{rank}(S^\sharp) - T| + \frac{1}{2}\|W - RS\|_F^2 \\ &\text{subject to: } S^\sharp = g(S) \end{aligned} \tag{2.26}$$

where, T is the target rank of the shape matrix. However, such an optimization needs an introduction to new operator known as PSVT [140] to optimize the problem. Nevertheless, PSVT can be regarded as special case of solving the weighted nuclear norm minimization [30, 60]. Therefore, the point is, depending on the application, the proposed approach can be modified or changed, hence, its flexible.

Q. Why not add the motion regularisation $\|R_t - R_{t-1}\|_F$ in the final optimisation and solve for both motion and shape?

It's definitely a valid argument. Nevertheless, adding this motion regularization goes against "simple prior free approach" [44] algorithm which is "solve for motion first and then solve for shape without any extra motion constraint", therefore, we didn't add it in the final optimisation. We showed that smooth solution already exist within the solution to intersection theorem. Comprehensive analysis of our algorithm after adding motion regularisation to solve the final optimisation is left as an extension to the present idea.

2.6 Closing Remarks on Prior-Free Approach

With weighted nuclear norm minimization of the shape matrix and an analytic solution to the rotation matrix based on the smoothness of the camera motion, we witnessed that the simple prior-free idea performs almost as good as the best algorithm's. Without exploiting the "prior-free" idea[44] *fully* based on the well-known assumptions of smooth deformation of the non-rigid object and its low-rank shape, it may perform badly, which might be the reason that researchers have had poor results using it, even for the non-rigid objects that span a single linear subspace. Our work revealed the possibility of making "prior-free" algorithm[44] more accurate under the different conditions of measurement matrix with elementary modifications, and also posed some open problems. The accuracy of our algorithm on the benchmark datasets empirically validates that the "prior-free" theory is still a very powerful way to solve NRSfM and therefore, the proposition before the NRSfM researchers to consider is, it's not the failure of the *concept* behind the prior-free idea for its inferior performance but, it's possibly due to our inability to correctly cater, and cleverly exploit the arc of information and perspectives provided by it to solve NRSfM.

Note: The next three chapters in the thesis uses nuclear norm minimization of the shape matrix rather than weighted nuclear norm minimization. The reason of this inconsistency is: The research work presented in those chapters were done before chapter (2). Nevertheless, it should not affect the overall flow of idea in the thesis. The foundation developed in this chapter shall make it simple for the readers to understand the upcoming chapters without any loss of generality.

3

From single body to multi-body non-rigid structure from motion

Contents

3.1 Motivation for multi-body NRSFM

Until now, NRSFM methods are focused on recovering the 3D structure of a single non-rigidly deforming object. To handle the real world scenarios where multiple deforming objects are present, existing methods can be used by pre-segmenting different objects in the scene

and perform non-rigid 3D reconstruction for each individual subject. However, such an approach fails to exploit the inherent behavior of the motion and structure problem. This is important because, in NRSfM factorization setting, motion and structure interact and therefore, to completely isolate structure from motion seems difficult. As a result, any joint solution to segmentation and reconstruction could benefit each other. In this chapter, we will introduce a unified framework to jointly segment and reconstruct multiple non-rigid objects. To compactly represent complex multi-body non-rigid scenes, we propose to exploit the change in the behavior of the object along both spatial and temporal space. Specifically, we represent the 3D deforming shapes as lying in a union of subspaces along the temporal space and represent the 3D shape trajectories as lying in the union of subspaces along the spatial space. We will show that solution to this spatiotemporal representation provides reliable 3D reconstruction and segmentation of multiple non-rigid objects present in the scene.

3.2 Introduction to Multi-body NRSFM

Non-rigid structure from motion (NRSfM) is central to many computer vision applications and has received considerable attention in recent years. Although existing approaches in NRSfM [21] [42] [172] [64] [6] have presented promising results but all of these methods assume that there is only one object is present in the scene. However, real-world scenes are much more complex, for example, multiple persons performing different activities in a traffic scene, soccer players in the playground, salsa dance *etc*. All these real-world examples constitute multi-body non-rigid deformation which could not be explained well with the single non-rigid object assumption. Therefore, it is quite natural to extend single-body NRSfM to multi-body NRSfM where the task would be to jointly reconstruct and segment multiple 3D deforming objects over-time.

To solve the problem of multi-body NRSfM, a natural and direct two-stage process is to reconstruct non-rigid multi-body structure by applying state-of-the-art non-rigid reconstruction methods [45][120] [206] and then segment distinct objects using clustering algorithms and vice-versa. However, by adopting such pipelines the inherent structure of the problem has never been exploited, *i.e.*non-rigid motion segmentation provides critical information to constrain 3D reconstruction while 3D non-rigid reconstruction could also constrain the corresponding motion segmentation problem. Furthermore, since the non-rigid shape deformation actually occurs in 3D space, it is more intuitive to perform segmentation of objects in 3D space rather than on projected 2D image space.

Additionally, it is always convenient–both computationally and numerically, to solve a given task using a unified framework than solving it in different stages. Therefore, in this chap-

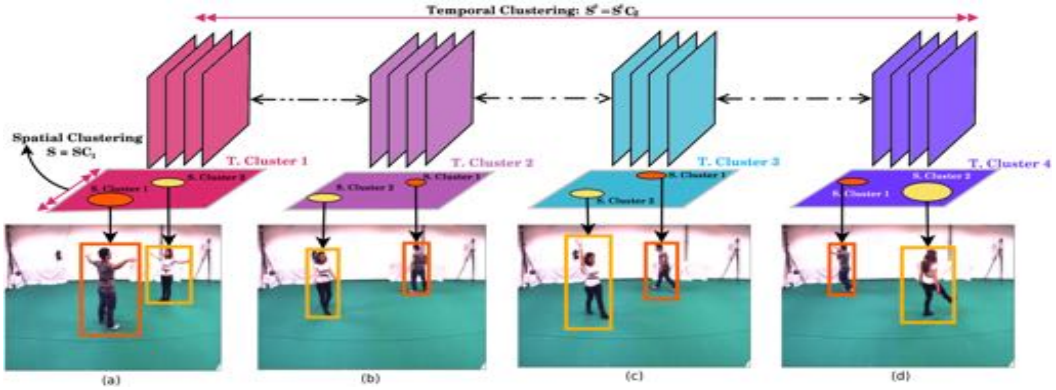


Figure 3.1: Illustration of the two clustering constraints used in our framework. We observe that, when different objects are undergoing complex non-rigid motion, the temporal clustering helps in improving the 3D reconstruction by clustering different activities over-time such as stretch, walking, jumping and etc. The spatial clustering helps in explaining the segmentation of distinct structures over images. Frames with similar activities are shown in the same colors and different subjects undergoing deformations are shown in box. Here, T. Cluster refers to the Temporal cluster and S. Cluster refers to the Spatial Cluster. This flow diagram demonstrates that subjects performing different activities over-time lie in distinct temporal subspace and spatial subspace, subsequently different 3D trajectories spanned by different structures lies in distinct subspace. The example images are taken from the UMPM dataset [181].

ter, we introduce an algorithm that help reconstruct and cluster multiple non-rigid shapes present in the scene. Using this algorithm can explain the dynamics of non-rigid shape in a more intuitive way. Explicitly, we represent multi-body NRSfM as a union of subspace problem both in 3D trajectory space (spatially) and 3D shape space (temporally). We use the notion that each 3D trajectory can be expressed with other trajectories only if the trajectory is from the same subspace (spatial clustering) [109], and each individual shape can be expressed with other shapes belonging to the same subspace (temporal clustering) [206]. A visual illustration of the spatiotemporal subspace concept is presented in Fig. (3.1). Concretely, spatial clustering tries to reconstruct a trajectory using an affine combination of other trajectories from the same deforming object, while temporal clustering tries to explain the shape of deforming objects using an affine combination of other shapes at different frame instance.

By exploiting the spatio-temporal clustering structure, the algorithm is able to procure the affinity matrices that naturally encode subspace information. From the affinity matrices, direct inference about number of deformable objects, different activities and membership of each sample to achieve reconstruction can be deduced. Furthermore, we exploit the fact that the connectivity between subspaces must be tight if it belongs to the same subspace and loose if belongs to different subspaces. Therefore, we propose to use a mixture of ℓ_1 norm and ℓ_2 norm regularization (also known as the Elastic Net [208]), which helps in controlling the

sparsity of the affinity matrices.

3.3 Previous Relevant Work

Multi-body structure from motion (SfM) is an important problem in computer vision. To work out this problem for rigid motion is a direct extension to multi-view geometry techniques [57][141][201]. However, the solution to multi-body NRSfM is not straightforward, due to the difficulty in modeling complex non-rigid variations. Recent state-of-the-art in NRSfM reconstruction [45] has shown promising results while Zhu *et al.* [206] proposed that such an approach may fail while modeling long-term complex non-rigid motions. The work quoted that Dai *et al.* [42] work is "highly dependent on the complexity of the motion" [206]. Hence, to overcome this difficulty they suggested to represent long-term non-rigid motion as a union of subspace rather than a single subspace. Subsequently, Cho *et al.* [34] used probabilistic variations to model complex shape.

Despite the above accomplishments, NRSfM is still far behind its rigid counterpart. This gap is principally due to difficulty in modeling real-world non-rigid deformation. If the deformation is irregular or arbitrary then to explain the 3D structure using image data seems very difficult. Nevertheless, many real-world deformations are not arbitrary but are regular/smooth and therefore, it can be constrained. For example, Bregler *et al.* in his seminal work demonstrated that non-rigid deformation can be represented by a linear combination of a set of shape basis [21]. Following the work, several researchers tried to model NRSfM by utilizing additional constraints [174], [195], [142]. In 2008, Akhter *et al.* [6] presented a dual approach by modeling 3D trajectories. In 2009, Akhter *et al.* [5] proved that even there is an ambiguity in shape bases or trajectory bases, non-rigid shapes can still be solved uniquely without any ambiguity. In 2012, Dai *et al.* [42] proposed a "prior-free" method to recover camera motion and 3D non-rigid deformation by exploiting low-rank constraint only. Besides shape basis model and trajectory basis model, the shape-trajectory approach [70] combines two models and formulates the problems as revealing trajectory of the shape basis coefficients. Besides linear combination model, Lee *et al.* [120] proposed a Procrustean Normal Distribution (PND) model, where 3D shapes are aligned and fit into a normal distribution. Simon *et al.* [155] exploited the Kronecker pattern in the shape-trajectory (spatial-temporal) priors. Zhu and Lucey [207] applied the convolutional sparse coding technique to NRSFM using point trajectories. However, the method requires to learn an over-complete basis of 3D trajectories a priori to perform 3D reconstruction.

Recently, Russell *et al.* [151] proposed to simultaneously segment a complex dynamic scene containing a mixture of multiple objects into constituent objects and reconstruct a 3D model

of the scene by formulating the problem as hierarchical graph-cut based segmentation, where the whole scene is decomposed into background and foreground objects with complex motion of non-rigid or articulated objects are modeled as a set of overlapping rigid parts.

3.4 Chapter contribution

Our algorithm varies from the aforementioned works in the following aspects:

1. It introduces the first algorithm to solve multi-body non-rigid structure from motion under factorization [109].
2. A joint segmentation and reconstruction framework to solve the task of complex multi-body NRSfM by exploiting the inherent spatio-temporal union of subspace constraint.
3. Efficient solution to the resultant non-convex optimization problem based on the Alternating Direction Method of Multipliers (ADMM) method [19].
4. Extensive experimental results on both synthetic and real multi-body NRSfM datasets demonstrate the superior performance of our proposed algorithm.

3.5 Problem formulation and solution

Under our formulation, we intend to reconstruct 3D non-rigid shapes such that they satisfy both the spatio-temporal union of affine subspace constraint and the non-rigid shape constraints (low rank and spatial coherency). Similar to last chapter, let $W \in \mathbb{R}^{2F\times P}$ represent the *measurement matrix* with F as the number of frames and P be the number of feature points. We use the *orthographic camera* model and eliminate the translation component of the motions as suggested in [21].

$$W = RS, \tag{3.1}$$

where $R = \text{blockdiagonal}(R_1, \cdots, R_F) \in \mathbb{R}^{2F\times 3F}$ denotes the camera rotation matrix and S represents the 3D shapes of deforming objects over entire frames. This classical representation for NRSfM problem [21] aims at recovering both the *camera motion* R and the non-rigid 3D shapes $S \in \mathbb{R}^{3F\times P}$ from the 2D *measurement matrix* $W \in \mathbb{R}^{2F\times P}$ such that $W = RS$. Following the same representation to cater 2D-3D relation, we use $\|W - RS\|_F^2$ to infer the re-projection error.

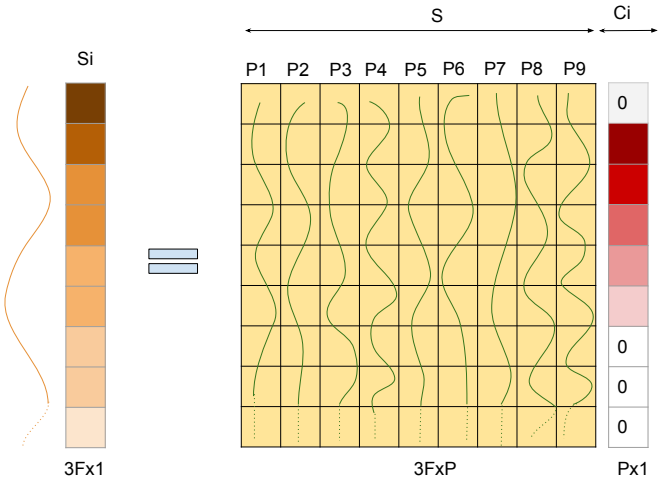


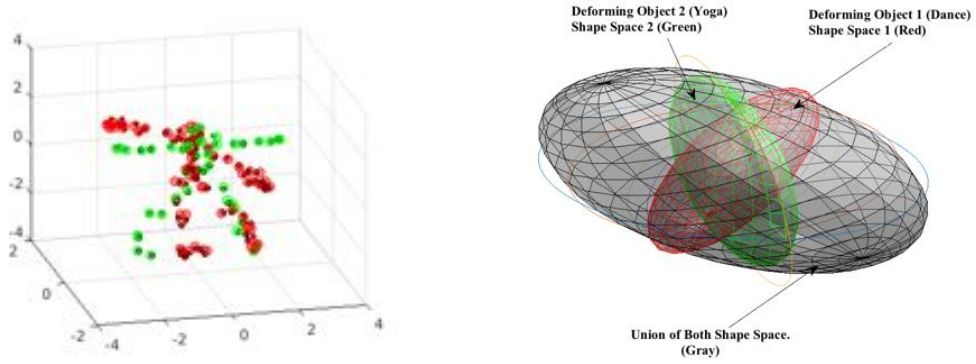
Figure 3.2: Visual illustration of the affine subspace constraint $S_i = SC_i$ in trajectory space. Each column of S is a trajectory of a 3D point (shown in green). This visualization states that a trajectory S_i can be reconstructed using affine combination of few other trajectories. *Note* : This pictorial representation is provided for better understanding and is only for illustration purpose. (Best viewed in color)

Representing multiple non-rigid deformations in trajectory space

To represent multiple non-rigid objects using a single linear trajectory space does not provide compact representation of 3D trajectories [206]. When there are multiple non-rigid objects, each object can be characterized as lying in an affine subspace. As a result, the 3D trajectories lying in a union of affine subspaces can equivalently be formulated in terms of self-expressiveness *i.e.*,

$$S = SC_1, \text{diag}(C_1) = 0, 1^T C_1 = 1^T. \tag{3.2}$$

where $S \in \mathbb{R}^{3F \times P}$, $C_1 \in \mathbb{R}^{P \times P}$. To get rid of the trivial solution of $S = S$ or $C_1 = I$, we explicitly enforce the diagonal constraint as $\text{diag}(C_1) = 0$. As we represent each non-rigid object as lying in an affine subspace, we further enforce the affine constraint $1^T C_1 = 1^T$. Besides the above constraint, we also want to enforce a constraint that if the trajectories belong to the same deforming object then it must be tightly connected or loosely connected the otherwise. To cater this idea of inter-class and intra-class trajectories clustering, we use the elastic net formulation [199] to compromise between connectedness and sparsity. Combining all



(a) Dance and Yoga, 3D reconstruction & segmentation **(b)** Shape space representation using Ellipsoid

Figure 3.3: Visual representation of union of subspace in shape space. (a) Two different subjects are performing Dance (Red) and Yoga (Green) respectively. (b) Equivalent representation of both activities in shape space for a single frame with green ellipsoid showing the shape space for Yoga activity and red ellipsoid showing the Dance activity. It can be observed that the space spanned by different shapes performing different activities span a distinct subspace. Gray color ellipsoid shows the union of both subspaces. (Best viewed in color)

the constraints together, we reach the following optimization:

$$\begin{aligned} &\underset{C_1}{\text{minimize}}\ \lambda_1 \|C_1\|_1 + \frac{(1-\lambda_1)}{2}\|C_1\|_F^2 \\ &\textit{subject to:} \\ &S = SC_1, \text{diag}(C_1) = 0, 1^T C_1 = 1^T, \lambda_1 \in [0,1]. \end{aligned} \tag{3.3}$$

A visual illustration of this idea in trajectory space for a single trajectory is provided in Fig. (3.2). Here, $\|.\|_1$ and $\|.\|_F$ denote the ℓ_1-norm and the Frobenius norm respectively.

Representing multiple non-rigid deformations in shape space

An example of complex non-rigid motion is shown in Fig. (3.1), where the subjects are performing different activities at different time instances. Such distinct activities adheres to different local subspace and complete non-rigid behavior throughout the video lies in union of shape subspace. As mentioned in [206] such assumption leads to superior 3D reconstruction. To incorporate this concept in our formulation that different activities lie in union of affine subspaces, we express the 3D shapes in terms of self-expressiveness of frames along temporal direction.

$$S^\sharp = S^\sharp C_2, \text{diag}(C_2) = 0, 1^T C_2 = 1^T. \tag{3.4}$$

where $S^\sharp \in \mathbb{R}^{3P\times F}$ is the reshuffled version of S representing the per-frame 3D shape as a column vector, $C_2 \in \mathbb{R}^{F\times F}$. A visual intuition of this idea in shape space for single frame is

provided in Fig.(3.3).

For temporal coefficient matrix, we again use the elastic net regularizer due to the aforementioned reason. Using it gives the following optimization:

$$\begin{aligned} &\underset{C_2}{\text{minimize}}\ \lambda_3\|C_2\|_1 + \frac{(1-\lambda_3)}{2}\|C_2\|_F^2 \\ &\textit{subject to:} \\ &S^\sharp = S^\sharp C_2, \text{diag}(C_2) = 0, 1^T C_2 = 1^T, \lambda_3 \in [0,1]. \end{aligned} \tag{3.5}$$

Enforcing the global shape constraint

In seeking a compact representation for multi-body non-rigid objects, we penalize the number of independent non-rigid shapes. Similar to [42] and [64], we penalize the nuclear norm of the reshuffled shape matrix $S^\sharp \in \mathbb{R}^{3P\times F}$, this is because the nuclear norm is known as the convex envelope of the rank function. In this way, the global shape constraint is expressed as:

$$\|S^\sharp\|_*, \tag{3.6}$$

where $\|\|_*$ denotes the nuclear norm of the matrix, i.e, sum of singular values.

Joint Reconstruction and Segmentation Formulation

Putting all the above constraints (spatio-temporal union of subspace constraint and global shape constraint) together, we reach a multi-body non-rigid reconstruction and segmentation formulation:

$$\begin{aligned} &\underset{S,C_1,C_2,S^\sharp}{\text{minimize}}\ \frac{1}{2}\|W - RS\|_F^2 + \lambda_1\|C_1\|_1 + \frac{1-\lambda_1}{2}\|C_1\|_F^2 + \lambda_2\|S^\sharp\|_* + \lambda_3\|C_2\|_1 + \frac{1-\lambda_3}{2}\|C_2\|_F^2 \\ &\textit{subject to:} \\ &S = SC_1, S^\sharp = S^\sharp C_2, \\ &1^T C_1 = 1^T, 1^T C_2 = 1^T, \\ &\text{diag}(C_1) = 0, \text{diag}(C_2) = 0, \\ &\lambda_1, \lambda_3 \in [0,1]. \end{aligned} \tag{3.7}$$

where $S^\sharp \in \mathbb{R}^{3P\times F}$, $C_1 \in \mathbb{R}^{P\times P}$, and $C_2 \in \mathbb{R}^{F\times F}$. $\lambda_1, \lambda_2, \lambda_3$ are the trade-off parameters.

Solution

To solve the proposed optimization we introduce decoupling variables in Eq. 3.7, which leads to the following formulation:

$$\begin{aligned}
&\underset{S,J,E_1,E_2,C_1,C_2,S^\sharp}{\text{minimize}} \ \frac{1}{2}\|W - RS\|_F^2 + \lambda_1\|E_1\|_1 + \frac{1-\lambda_1}{2}\|E_1\|_F^2 + \lambda_2\|J\|_* + \lambda_3\|E_2\|_1 + \frac{1-\lambda_3}{2}\|E_2\|_F^2 \\
&\textit{subject to:} \\
&S^\sharp = g(S), S^\sharp = J, \\
&S = SC_1, S^\sharp = S^\sharp C_2, \\
&1^T C_1 = 1^T, 1^T C_2 = 1^T, \\
&\text{diag}(C_1) = 0, \text{diag}(C_2) = 0, \\
&C_1 = E_1, C_2 = E_2, \\
&\lambda_1, \lambda_3 \in [0, 1].
\end{aligned} \tag{3.8}$$

The auxiliary variables E_1, E_2, J are introduced to simplify the derivation. $g(.) : S_{3F\times P} \rightarrow S^\sharp_{3P\times F}$ denotes the linear mapping from $S \in \mathbb{R}^{3F\times P}$ to its reshuffled version $S^\sharp \in \mathbb{R}^{3P\times F}$. S = $\begin{bmatrix} X_{11} & X_{12} & X_{13} & \dots & X_{1P} \\ Y_{11} & Y_{12} & Y_{13} & \dots & Y_{1P} \\ Z_{11} & Z_{12} & Z_{13} & \dots & Z_{1P} \\ \dots & \dots & \dots & \dots & \dots \\ X_{F1} & X_{F2} & X_{F3} & \dots & X_{FP} \\ Y_{F1} & Y_{F2} & Y_{F3} & \dots & Y_{FP} \\ Z_{F1} & Z_{F2} & Z_{F3} & \dots & Z_{FP} \end{bmatrix}$ and $S^\sharp = \begin{bmatrix} X_{11}\dots X_{1P} & Y_{11}\dots Y_{1P} & Z_{11}\dots Z_{1P} \\ X_{21}\dots X_{2P} & Y_{21}\dots Y_{2P} & Z_{21}\dots Z_{2P} \\ \dots & \dots & \dots \\ X_{F1}\dots X_{FP} & Y_{F1}\dots Y_{FP} & Z_{F1}\dots Z_{FP} \end{bmatrix}^T$. The first term in the above optimization is meant for penalizing re-projection error under *orthographic* projection. Under single-body NRSFM configuration, 3D shape S can be well characterized as lying in a single low dimensional linear subspace. However, when there are multiple non-rigid objects, each non-rigid object could be characterized as lying in an affine subspace. To represent this idea mathematically in shape and trajectory space respectively, we introduce E_1 and E_2.

In addition to this, to reveal the intrinsic structure of multi-body non-rigid structure-from-motion (NRSfM), we seek for the sparsest solution both in trajectory and shape space. Consequently, we enforce the ℓ_1 norm for E_1 and E_2. However, high sparsity may lead to misclassification of samples or trajectories. Therefore, to maintain the balance between sparsity and connectedness, we incorporate the elastic net for both E_1 and E_2. Lastly, we enforce a global shape constraint ($\|J\|_*$) for compact representation of multi-body non-rigid objects

by penalizing the rank of the entire non-rigid shape.

Due to the two bilinear terms $S = SC_1$ and $S^\sharp = S^\sharp C_2$, the overall optimization of Eq.-(3.8) is non-convex. We solve it via the alternating direction method of multipliers (ADMM), which has a proven effectiveness for many non-convex problems and is widely used in computer vision task. ADMM works by decomposing the original optimization problem into several sub-problems, where each sub-problem can be solved efficiently. To this end, we seek to decompose Eq.(3.8) into several sub-problems.

Introducing Lagrangian multipliers in the Eq: (3.8) gives the Augmented Lagrangian formulation for Eq.(3.8) as

$$\begin{aligned}
&\mathcal{L}(S, S^\sharp, C_1, C_2, E_1, E_2, J, \{Y_i\}_{i=1}^8) = \frac{1}{2}\|W - RS\|_F^2 + \lambda_1\|E_1\|_1 + \gamma_1\|E_1\|_F^2 + \lambda_2\|J\|_* + \\
&\lambda_3\|E_2\|_1 + \gamma_3\|E_2\|_F^2 + < Y_1, S^\sharp - g(S) > + \frac{\beta}{2}\|S^\sharp - g(S)\|_F^2 + < Y_2, S - SC_1 > + \\
&\frac{\beta}{2}\|S - SC_1\|_F^2 + < Y_3, S^\sharp - S^\sharp C_2 > + \frac{\beta}{2}\|S^\sharp - S^\sharp C_2\|_F^2 + < Y_4, 1^T C_1 - 1^T > + \\
&\frac{\beta}{2}\|1^T C_1 - 1^T\|_F^2 + < Y_5, 1^T C_2 - 1^T > + \frac{\beta}{2}\|1^T C_2 - 1^T\|_F^2 + < Y_6, C_1 - E_1 > + \\
&\frac{\beta}{2}\|C_1 - E_1\|_F^2 + < Y_7, C_2 - E_2 > + \frac{\beta}{2}\|C_2 - E_2\|_F^2 + < Y_8, S^\sharp - J > + \frac{\beta}{2}\|S^\sharp - J\|_F^2,
\end{aligned} \tag{3.9}$$

where we define $\gamma_1 = (1-\lambda_1)/2$ and $\gamma_3 = (1-\lambda_3)/2$. Y_i for $i = \{1, \cdots, 8\}$ are the Lagrange multipliers. β is the penalty parameter, where we use the same parameter for each augmented Lagrange term to simplify the derivation and parameter setting. The symbol $< ., . >$ represents the Frobenius inner product of two matrices, i.e, the trace of the product of two matrices. For example, given two matrices $A, B \in \mathbb{R}^{m\times n}$, the Frobenius inner product is calculated as $< A, B >= \mathrm{Tr}(A^T B)$.

The ADMM works by minimizing Eq. (3.9) with respect to one variable while treating others as constant. During each iteration, we update each variable and the Lagrange multipliers in sequel. The detailed derivation for the solution is presented in the Appendix (B).

Solution for S: The closed form solution for S can be derived by taking derivative of Eq: (3.9) w.r.t to S and equating it to zero.

$$\frac{1}{\beta}(R^T R + \beta I)S + S(I - C_1)(I - C_1^T) = \frac{1}{\beta}R^T W + (g^{-1}(S^\sharp) + \frac{g^{-1}(Y_1)}{\beta} - \frac{Y_2}{\beta}(I - C_1^T)). \tag{3.10}$$

Solution for $S^\sharp$: The closed form solution for $S^\sharp$ can be derived by taking derivative of Eq: (3.9) w.r.t $S^\sharp$ and equating to zero.

$$S^\sharp(2I + (I - C_2)(I - C_2^T)) = (g(S) - \frac{Y_1}{\beta}) + (J - \frac{Y_8}{\beta}) - \frac{Y_3}{\beta}(I - C_2^T). \tag{3.11}$$

Solution for C_1 : The closed form solution for C_1 can be derived as

$$(S^T S + 11^T + I)C_1 = S^T(S + \frac{Y_2}{\beta}) + 1(1^T - \frac{Y_4}{\beta}) + (E_1 - \frac{Y_6}{\beta}). \tag{3.12}$$

$$C_1 := C_1 - \text{diag}(C_1), \tag{3.13}$$

Solution for C_2 : The closed form solution for C_2 can be derived as

$$((S^\sharp)^T S^\sharp + 11^T + I)C_2 = (S^\sharp)^T(S^\sharp + \frac{Y_3}{\beta}) + 1(1^T - \frac{Y_5}{\beta}) + (E_2 - \frac{Y_7}{\beta}). \tag{3.14}$$

$$C_2 := C_2 - \text{diag}(C_2), \tag{3.15}$$

Solution for J : The optimization of J given all the remaining variables can be expressed as:

$$\begin{aligned} J &= \underset{J}{\text{argmin}} \lambda_2 \|J\|_* + < Y_8, S^\sharp - J > + \frac{\beta}{2}\|S^\sharp - J\|_F^2. \\ &= \underset{J}{\text{argmin}} \lambda_2 \|J\|_* + \frac{\beta}{2}\|J - (S^\sharp + \frac{Y_8}{\beta})\|_F^2. \end{aligned} \tag{3.16}$$

A closed-form solution exists for this sub-problem. Let's define the soft-thresholding operation as $\mathcal{S}[\tau](x) = \text{sign}(x)\max(|x| - \tau, 0)$, the optimal J can be obtained as:

$$J = U\mathcal{S}[\frac{\lambda_2}{\beta}](\Sigma)V, \tag{3.17}$$

where $[U, \Sigma, V] = \text{SVD}(S^\sharp + Y_8/\beta)$.

Solution for E_1: The closed-form solution for E_1 can be obtained similarly:

$$E_1 = \mathcal{S}[\frac{\lambda_1}{\gamma_1 + \beta/2}]\left(\frac{\beta}{2\gamma_1 + \beta}(C_1 + \frac{Y_6}{\beta})\right). \tag{3.18}$$

Solution for E_2 The derivation for the solution of E_2 is similar to E_1.

$$E_2 = \mathcal{S}[\frac{\lambda_3}{\gamma_3 + \beta/2}] \left(\frac{\beta}{2\gamma_3 + \beta}(C_2 + \frac{Y_7}{\beta}) \right). \tag{3.19}$$

Detailed derivations to each sub-problems solution are provided in Appendix (B). Finally, the Lagrange multipliers $\{Y_i\}_{i=1}^{8}$ and β are updated as:

$$Y_1 = Y_1 + \beta(S^\sharp - g(S)), Y_2 = Y_2 + \beta(S - SC_1), \tag{3.20}$$

$$Y_3 = Y_3 + \beta(S^\sharp - S^\sharp C_2), Y_4 = Y_4 + \beta(1^T C_1 - 1^T) \tag{3.21}$$

$$Y_5 = Y_5 + \beta(1^T C_2 - 1^T), Y_6 = Y_6 + \beta(C_1 - E_1), \tag{3.22}$$

$$Y_7 = Y_7 + \beta(C_2 - E_2), Y_8 = Y_8 + \beta(S^\sharp - J). \tag{3.23}$$

$$\beta = \min(\beta_m, \beta\varrho). \tag{3.24}$$

Initialization: Since the proposed problem is non-convex, proper initialization is required for fast convergence. In this work, we obtained rotation using [42] and initialized the S matrix as pinv$(R)^* W$. β_0, β_m, ϱ were kept as 10^{-3}, 10^3, and 1.1 respectively. The complete implementation is provided in Algorithm (1).

3.6 Experiments and results

We provide extensive experiments on freely available benchmark data-sets. We tested our approach on both real data and synthetic data under sparse and semi-dense scenarios. Denote S^{est} as the estimated 3D structure and S^{GT} as the ground-truth structure. We use the following error metrics to evaluate the performance of the approach:
(i) Mean normalized error in multi-body non-rigid 3D reconstruction

$$e_{3D} = \frac{1}{F} \sum_{f=1}^{F} \|S_f^{est} - S_f^{GT}\|_F / \|S_f^{GT}\|_F, \tag{3.25}$$

(ii) Error in multi-body non-rigid motion segmentation,

$$e_{MS} = \frac{\text{Total number of incorrectly segmented trajectories}}{\text{Total number of trajectories}}. \tag{3.26}$$

Algorithm 1 Multi-body non-rigid 3D reconstruction and segmentation using ADMM

Require:

2D feature track matrix W, camera motion R, λ_1, λ_2, λ_3, $\varrho > 1$, β_m, ε;

Initialize: $S^{(0)}$, $S^{\sharp(0)}$, $C_1^{(0)}$, $E_1^{(0)}$, $C_2^{(0)}$, $E_2^{(0)}$, $\{Y_i^{(0)}\}_{i=1}^{8} = 0$, $\beta^{(0)} = 1e^{-3}$;

while not converged do

1. Update $(S, S^{\sharp}, E_1, E_2, C_1, C_2)$ by Eq. (3.10), Eq. (3.11), Eq. (3.18), Eq. (3.19), Eq. (3.13) and Eq. (3.15); The new value for each variable is updated over iteration.
2. Update $\{Y_i\}_{i=1}^{8}$ and β by Eq. (3.20)-Eq. (3.24);
3. Check the convergence conditions $\|S^{\sharp} - g(S)\|_{\infty} \leq \varepsilon$, $\|S - SC_1\|_{\infty} \leq \varepsilon$, $\|S^{\sharp} - S^{\sharp}C_2\|_{\infty} \leq \varepsilon$, $\|1^T C_1 - 1^T\|_{\infty} \leq \varepsilon$, $\|1^T C_2 - 1^T\|_{\infty} \leq \varepsilon$ and $\|C_1 - E_1\|_{\infty} \leq \varepsilon$, $\|C_2 - E_2\|_{\infty} \leq \varepsilon$; $\|S^{\sharp} - J\|_{\infty} \leq \varepsilon$;

end while

Ensure: C_1, C_2, E_1, E_2, S, $S^{\sharp}$.

Form an affinity matrix $\mathcal{A}_1 = |C_1| + |C_1^T|$, then apply spectral clustering [136] to $\mathcal{A}_1$ to achieve non-rigid motion segmentation.

Experiment 1: Performance on sparse dataset

Since our approach simultaneously reconstructs and segments multi-body non-rigid motions. Thus, we conducted the first experiment to verify the advantage of our method compared with alternative two stage approaches. To this end, we devise the following experimental setup, namely first segmenting the 2D tracks and then reconstructing each body with single body non-rigid structure-from-motion algorithm and vice-versa. Specifically, the two baseline setups are:

1. Baseline method 1: Single body non-rigid structure-from-motion (State-of-the-art "block-matrix method" [42] was used) followed by subspace clustering of the 3D trajectories (SSC [53] was used), denoted as "BMM+SSC(3D)".

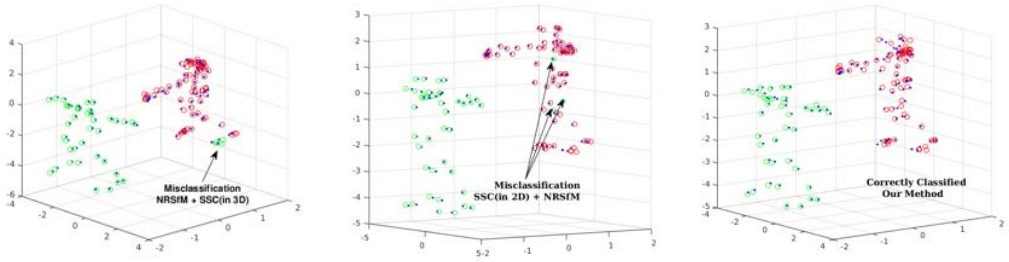
2. Baseline method 2: Subspace clustering of the 2D feature tracks (2D trajectories) followed by single body non-rigid structure-from-motion for each cluster of 2D feature tracks, denoted as "SSC(2D)+BMM".

In Table (3.1), we provide the statistical comparison between our method and the two baseline methods in dealing with multi-body non-rigid structure-from-motion problem.

Comments: In all of these sequences, our method achieves perfect motion segmentation and better non-rigid 3D reconstruction in most of the sequences compared with the two-staged

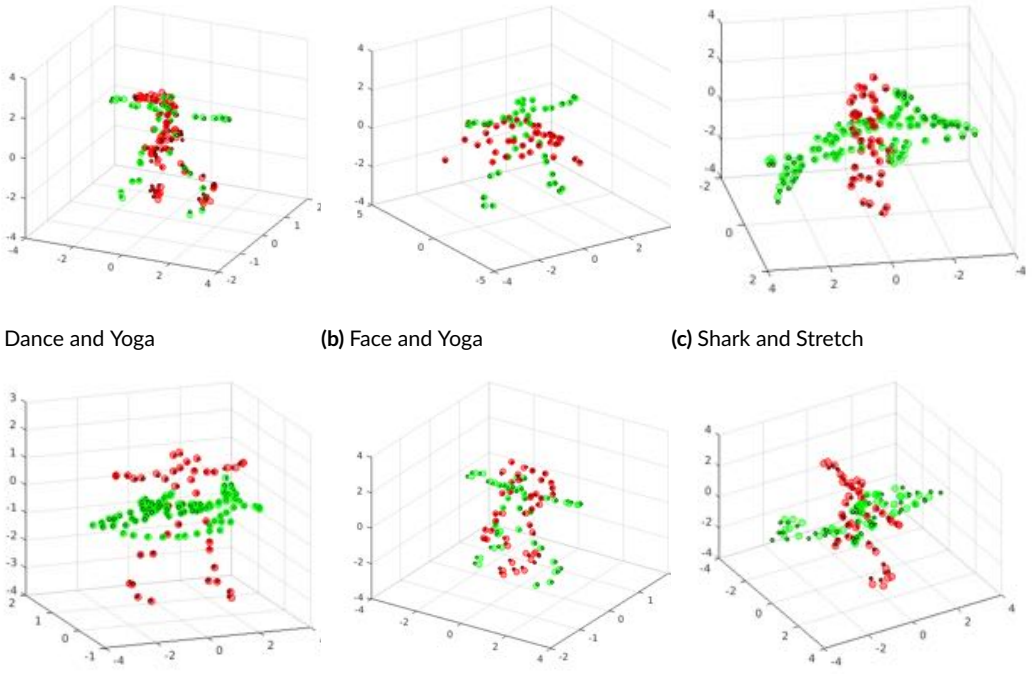
Datasets	BMM+SSC(3D)		SSC(2D)+BMM		Our Method	
	e_{3D}	e_{MS}	e_{3D}	e_{MS}	e_{3D}	e_{MS}
Dance + Yoga	0.045	0.034	0.058	0.026	0.045	0.00
Drink + Walking	0.074	0.0	0.085	0.0	0.073	0.00
Shark + Stretch	0.024	0.401	0.098	0.394	0.021	0.00
Walking + Yoga	0.070	0.0	0.090	0.0	0.066	0.00
Face + Pickup	0.032	0.098	0.023	0.098	0.027	0.00
Face + Yoga	0.017	0.012	0.033	0.012	0.021	0.00
Shark + Yoga	0.035	0.416	0.105	0.409	0.033	0.00
Stretch + Yoga	0.039	0.0	0.055	0.0	0.036	0.00

Table 3.1: Performance comparison between our method and the two stage methods i.e first cluster and then reconstruct or vice-versa, where 3D reconstruction error (e_{3D}) and non-rigid motion segmentation error (e_{MS}) are used as error metrics. The statistics clearly shows the superior performance of our method in both 3D reconstruction and motion segmentation compared with the two stage methods.



(a) NRSFM [44] and then SSC **(b)** SSC and then NRSFM [44] **(c)** Joint NRSFM and Segmentation

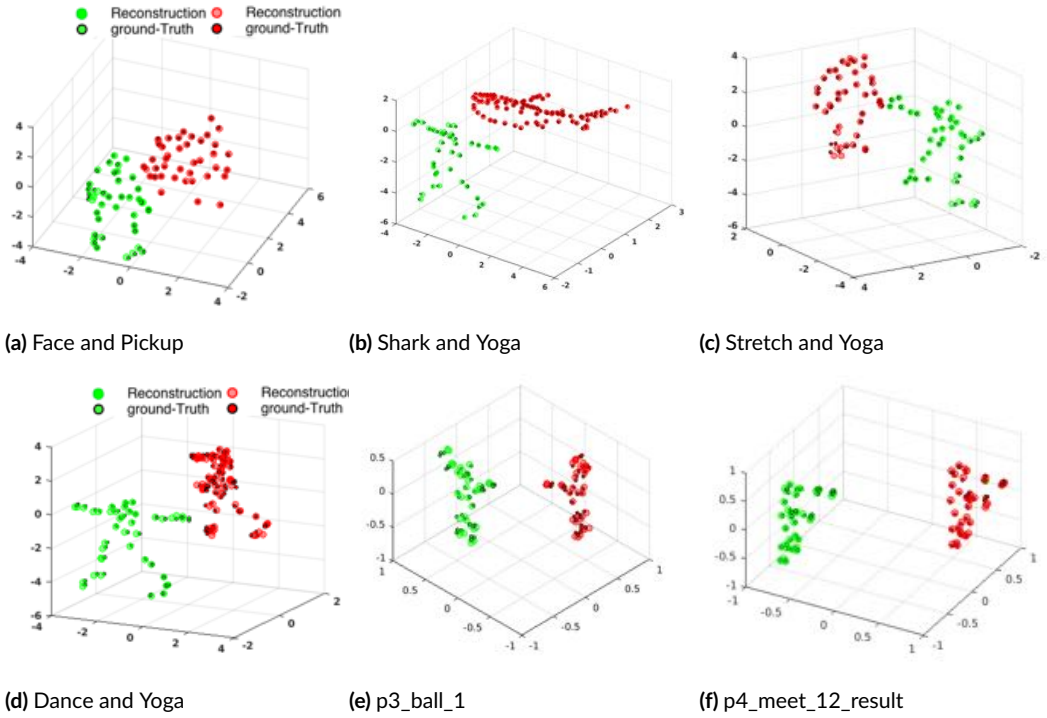
Figure 3.4: An illustration of the efficacy of our approach. The plot shows the results on the "Dance + Yoga" sequence. (a) Result obtained by applying BMM method [42] to get 3D reconstruction and then using SSC [53] to segment 3D points. (b) Result obtained by applying SSC [53] to 2D feature tracks and then using BMM [42] to each cluster to get 3D reconstruction. (c) Result from our simultaneous reconstruction and segmentation framework.



(a) Dance and Yoga **(b)** Face and Yoga **(c)** Shark and Stretch

(d) Shark and Yoga **(e)** Stretch and Yoga **(f)** Walking and Yoga

Figure 3.5: 3D reconstruction and segmentation of different complex multi-body non-rigid motion sequences, where different objects intersect with each other. a) Dance-Yoga Sequence b) Face-Yoga Sequence c) Shark-Stretch Sequence d) Shark-Yoga Sequence e) Stretch-Yoga Sequence f) Walking-Yoga. Different colors indicate different clusters with dark small circles in the respective segments shows the ground-truth 3D points. (Best viewed in color)



(a) Face and Pickup **(b)** Shark and Yoga **(c)** Stretch and Yoga

(d) Dance and Yoga **(e)** p3_ball_1 **(f)** p4_meet_12_result

Figure 3.6: 3D reconstruction and segmentation of different multi-body non-rigid motion sequences a) Face-Pickup Sequence b) Shark-Yoga Sequence c) Stretch-Yoga Sequence d) Dance-Yoga Sequence e) p3_ball_1 f) p4_meet_12. The non-rigid motion sequences are generated from the CMU MoCap dataset [6], Torresani et al. [176] dataset and the UMPM dataset [181]. Different colors indicate different clusters with dark small circles in the respective segments shows the ground-truth 3D points. (Best viewed in color)

approaches–statistical value for the same sequences can be inferred from Table (3.1). Visual comparison is also provided in Fig. (3.4) for easy understanding. The results clearly illustrates that with the proposed framework we can procure correct features belonging to each object than the two-stage pipeline.

To further test the segmentation of different deforming objects performing different activities, we designed two synthetic experimental settings. In the first setting, we combined non-rigid objects such that they are well separated in 3D space. In the next experiment setting the objects are intersecting with each other in 3D space. We obtained perfect segmentation results for both settings. Fig. (3.5) and Fig. (3.6) show the qualitative segmentation and reconstruction results for the corresponding experiment. Quantitative performance comparison of segmentation with SSC [52] on synthetic sequence is presented in Table 3.1 .

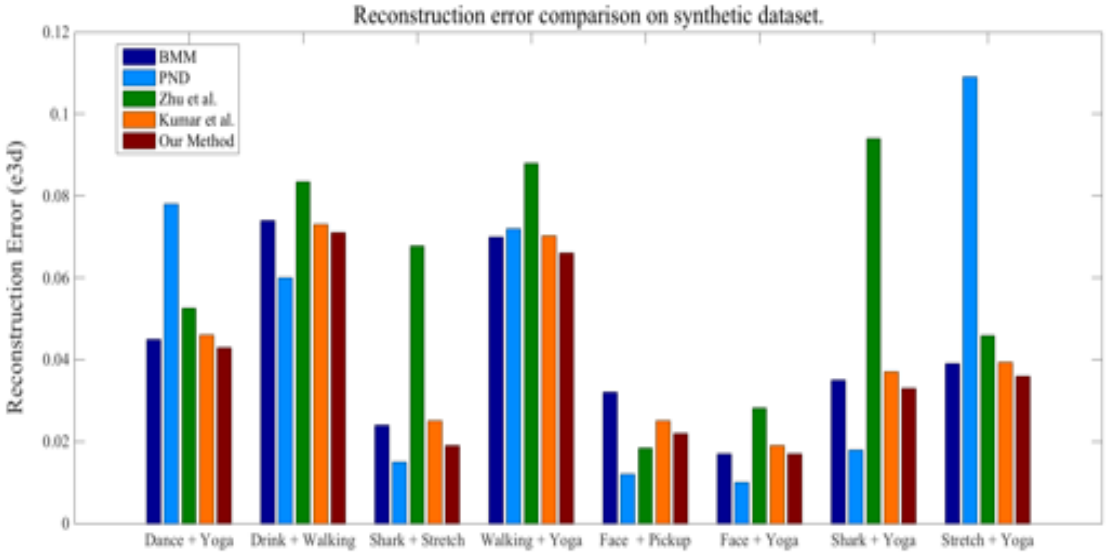


Figure 3.7: Comparison of 3D reconstruction error with other competitive methods on synthetic datasets (CMU Mocap [6] and [174]). The comparison methods (BMM [42], PND [120], Zhu et al. [206], Kumar et al. [109]) present state-of-the-art approaches. Note: Code for Zhu *et al.* [206] work is not publicly available, the statistics we provide here are obtained from our own implementation of this method. For exact numerical values, please refer to Table B.1 (Best viewed in color).

Performance comparison of reconstruction error with state-of-the-art methods on synthetic dataset

We compared the performance of our approach with other state-of-the-art NRSFM methods on the same dataset under similar settings. Synthetic dataset that are used for evaluating reconstruction error of multi-body non-rigid deformations are created by combining different objects from the Mocap [6] and Torresani *et al.* dataset [176]. We compare our approach with methods such as BMM [42], PND [120], Zhu *et al.* [206] and Kumar *et al.* [109]. Statistical results are provided in Fig. (3.7) which clearly indicates the improvement in 3D reconstruction accuracy using our method in comparison to other approaches.

Comments: It can be observed from Fig. (3.7) that the reconstruction error obtained by our method in comparison to other state-of-the-art is either better or close to other competing approaches on all the datasets. We would like to mention that code for Zhu *et al.* [206] is not publicly available. Therefore, we used our own implementation of this algorithm for numerical comparison. MATLAB codes for other method such as BMM [42] and PND [120] are freely available for research purpose.

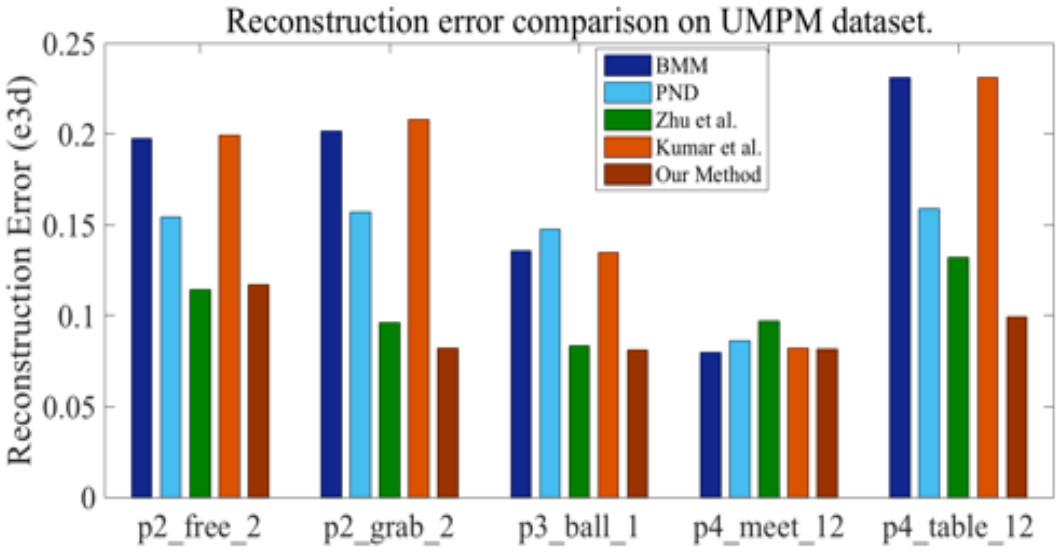


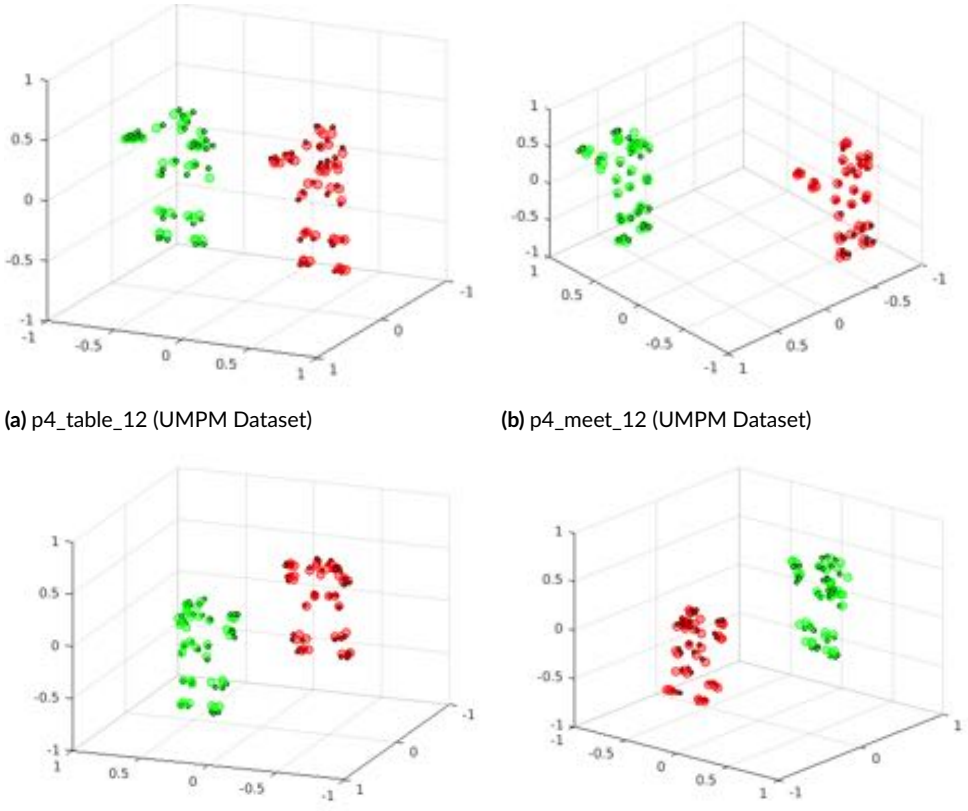
Figure 3.8: Comparison of 3D reconstruction error with other competitive methods on real image data-set(UMPM [181]), which is composed of complex non-rigid deformation along with different activities over-time. The comparison methods (BMM [42], PND [120], Zhu et al. [206], Kumar et al. [109]) present state-of-the-art approaches. For exact numerical values, please refer to the Table B.2 (Best viewed in color).

Experiment 2: Performance on real image dataset UMPM [181].

UMPM : The Utrecht Multi-Person Motion (UMPM) dataset [181] is a benchmark dataset for multiple person interaction. It consists of synchronized videos with 644×484 resolution images. Each dataset consists of long-video sequence with multiple activities and different articulated motions. Although data are provided from four view point for each category, we only used one view point for evaluation. This dataset has been used in the past as a benchmark to evaluate multi-person motion capturing technique and many state-of-the-art techniques have used it to evaluate the performance of NRSfM methods [120], [51].

Performance comparison of 3D reconstruction with state-of-the-art methods on UMPM dataset [181]

Following previous works on this topic, we used the UMPM dataset for evaluation of our method in comparison to other competing methods. We evaluated our performance on five long video sequence which are composed of complex non-rigid motion and extensive variations of daily human actions with severe pose changes. Namely we tested our method on p4_table_12, p4_meet_12, p2_grab_2, p2_free_2, and p3_ball_1 sequence.



(a) p4_table_12 (UMPM Dataset)

(b) p4_meet_12 (UMPM Dataset)

(c) p2_grab_2 (UMPM Dataset)

(d) p2_free_2 (UMPM Dataset)

Figure 3.9: In (a), (b), (c), (d) larger and smaller circles shows the 3D reconstruction and ground-truth of p4_table_12, p4_meet_12, p2_grab_2, p2_free_2 data-set respectively. Different colors show the corresponding segmentation. (Best viewed in color)

Comments: The observations on real image experiments are very similar to the synthetic ones. In all the aforementioned datasets, we obtained almost perfect segmentation along with reliable 3D reconstruction. Fig.(3.8) demonstrates the superior 3D reconstruction performance of our method in comparison to other methods. Furthermore, qualitative results obtained using our approach on the UMPM dataset can be inferred from Fig.(3.9) and Fig.(3.10). Spatial and temporal affinity matrices obtained during the experiment on real sequence are analogous to synthetic sequence and therefore, similar inference can be drawn. The numerical values clearly indicate the superiority of our approach on 3D reconstruction, in addition it provides robust segmentation of multiple deformable objects.

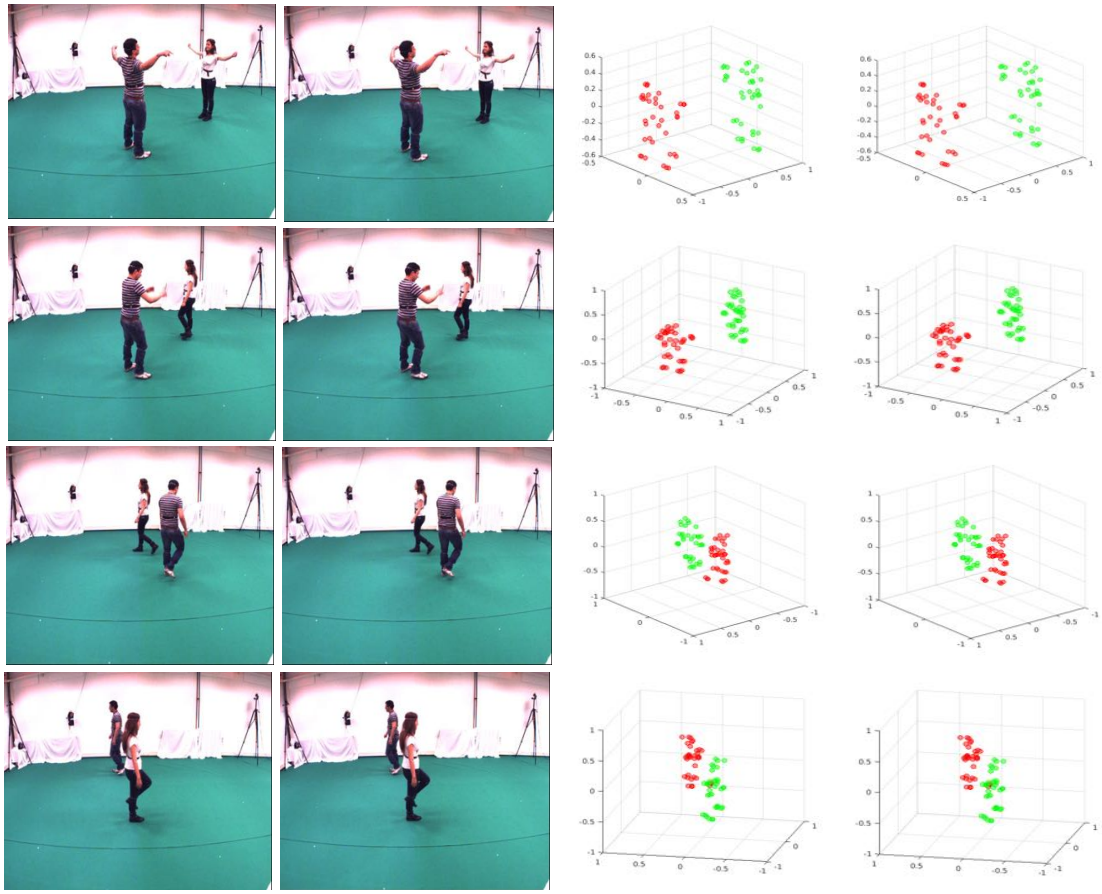


Figure 3.10: 3D non-rigid reconstruction and segmentation results on p2_free_2 sequence of the UMPM dataset [182]. We obtained perfect segmentation and reliable 3D reconstruction over the entire video sequence which comprises of complex non-rigid deformation followed by different activities. (Best viewed in color)

Experiment 3: Performance on dense sequences

We also tested our method on freely available dense datasets [64]. Although our method is not scalable to millions of feature tracks, for completeness of our evaluation, we tested our method on the uniformly sampled features points of the original sequences. We performed experiments on benchmark dense NRSfM synthetic and real dataset sequence [64] introduced by Grag *et al.*[65]. The synthetic sequence consists of four different face datasets. Each sequence has different deformation and camera motion over frames.

We sampled 3275 trajectories from each synthetic face sequence to verify the performance of our approach. The 3D reconstruction errors obtained on these four face sequence are shown

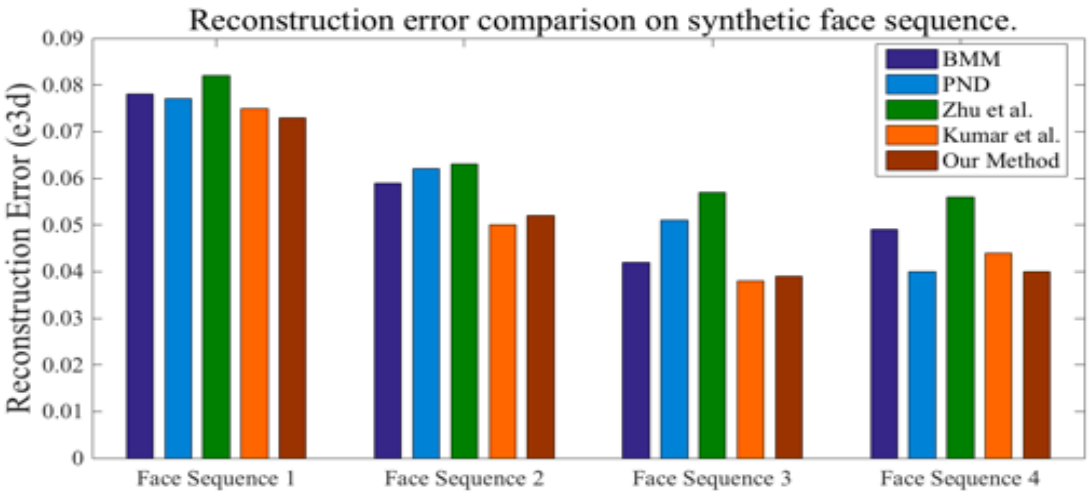


Figure 3.11: Comparison of 3D reconstruction error with other competitive methods on synthetic dense face sequence ([64]) which is composed of non-rigid face deformation of different facial expression over-time. The comparison methods (BMM [42], PND [120], Zhu et al. [206], Kumar et al. [109]) represent the state-of-the-art approaches. This comparison is made over 3275 feature tracks which is taken by uniformly sampling the dense feature tracks. For exact numerical values, please refer to the Table B.3. (Best viewed in color).

in Fig. (3.11). Fig.(3.12) show the qualitative of reconstruction result that is obtained using our method. In qualitative illustration *i.e.*, Fig. (3.12), the green dots show the reconstructed points whereas the red dots show the ground-truth 3D structure.

Face with a background is very common in real world scenarios. To test segmentation and reconstruction in such cases, we combined synthetic face with an artificial background and projected it using an orthographic camera model. We use these projected 2D feature tracks as input to our algorithm and obtained its 3D shapes as shown in Fig.(3.13). Different colors represent distinct clusters that are recovered using our method.

Real face, back and heart sequence

Garg *et al.* [64] dataset is composed of three monocular videos namely face, back and heart sequence. These sequence captures the natural human deformation with considerable displacements from one frame to other. In the face sequence, the subject performs day-to-day facial expression whereas in the back sequence the person is stretching and shrinking his back wearing a textured t-shirt. Lastly, this dataset also provides a challenging monocular heartbeat sequence taken during bypass surgery. Quantitative evaluation on this dataset is not performed due to the absence of ground-truth 3D values. However, qualitative results obtained are shown in Fig.(3.14a), Fig.(3.14b) and Fig.(3.14c) respectively. The qualitative results show

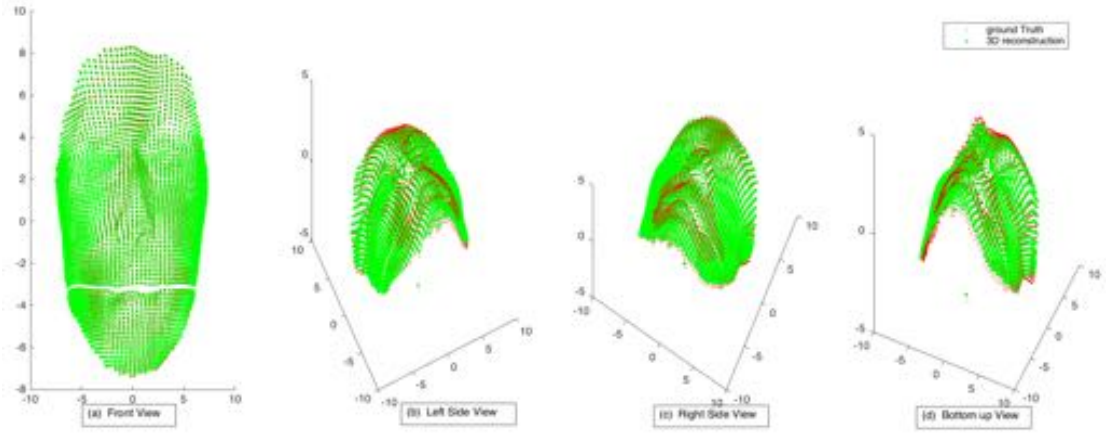


Figure 3.12: Results on synthetic face sequence [64]. Red and green color show the ground-truth and reconstructed 3D structures respectively. (Best viewed in color)

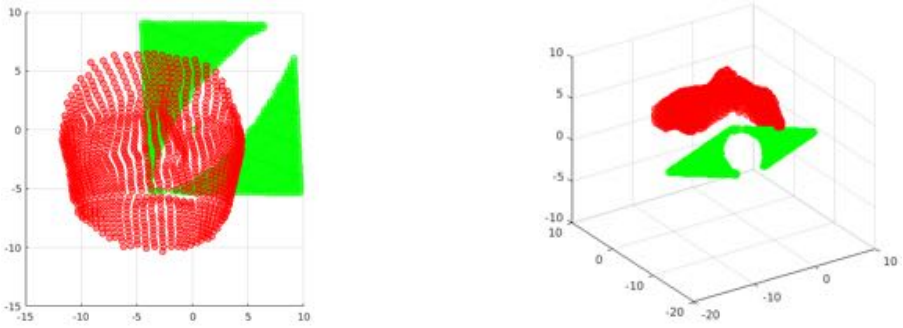
the high-quality 3D reconstruction of the object using our method on real world video's.

Experiment 4: Evaluation on more than two objects.

We also evaluated our method when three objects in the scene are performing complex motions over time. It was observed during the experiment that shape clustering with trajectory clustering does not affect the segmentation, while it can help improve 3D reconstruction. A graphical illustration of such example along with our obtained results is shown in Fig.(3.15).

Experiment 5: Convergence and analysis of the proposed optimization.

Since the proposed optimization is non-convex, we conducted experiments to study the convergence and timings of our approach. Fig. (3.16) shows a typical convergence curve of the proposed optimization on Shark+Yoga dataset. The optimization curve is provided only for better intuition of the algorithm. Similar trends of the convergence curves were observed for other datasets as well. In this figure different curves show the primal residuals for each optimization terms over iteration. The current implementation takes around 5-7 minutes for thousand feature tracks to converge on commodity desktop installed with Ubuntu 14.04 OS. The above simulation time is observed using MATLAB R2015b software running on Intel core i7 processor with 16GB RAM.



(a) Foreground (Red), Background (Green) [Front View] **(b)** Foreground (Red), Background (Green) [Side View]

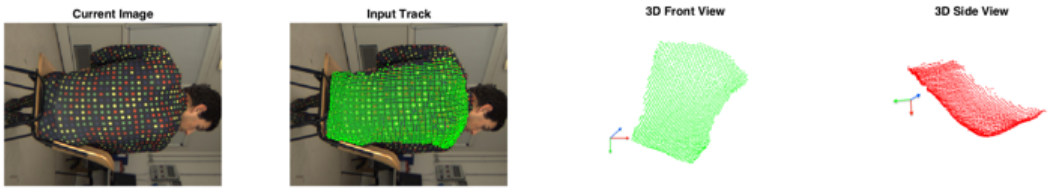
Figure 3.13: (a), (b) show the front view and side view of the reconstruction and segmentation result obtained on "Face+Background" Sequence. This dataset was synthetically generated by combining synthetic face sequence [66] with background as mask. (Best viewed in color)

Additional analysis

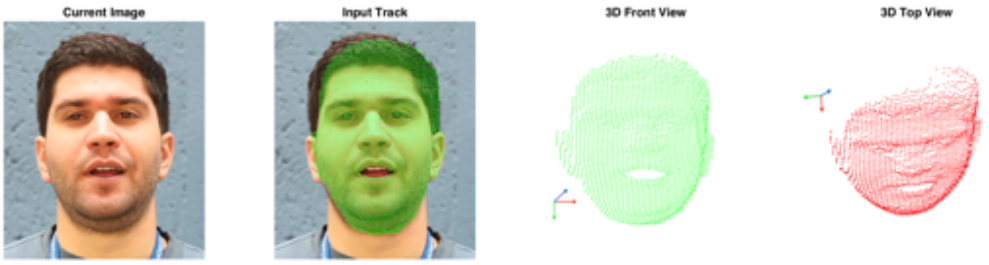
High values of λ_1 and λ_3 (say 0.6 or 0.7) during optimization may lead to high segmentation error due to very sparse structure in matrices. The benefit of elastic net is that it provides the flexibility to trade off between the sparsity and connectedness among different classes. Mathematically, it means, with elastic net we have the freedom to adjust between ℓ_1 and ℓ_2 minimization of the same optimization variable. Such regularization is handy in controlling the sparsity of the matrix. Fig.(3.18) shows the sparsity of C_1 matrix with variation in λ_1 for different sparse synthetic dataset. Fig.(3.17a) and Fig.(3.17b) show the affinity matrix of $C_1 \in \mathbb{R}^{P\times P}$ and $C_2 \in \mathbb{R}^{F\times F}$ for the Dance-Yoga sequence. The block-diagonal structure corresponding to both deforming objects is shown in Fig. (3.17a). Clearly, the two objects span different subspace that are independent of each other. The obtained affinity matrix of C_1 implies that the trajectories of each individual objects are self-expressive and thus each trajectory can be represented as a linear combination of other trajectories. Similarly, Fig.(3.17b) show similar shapes spans its own subspace and therefore, the frames corresponding identical activity can be clustered.

3.7 Limitations of the proposed approach

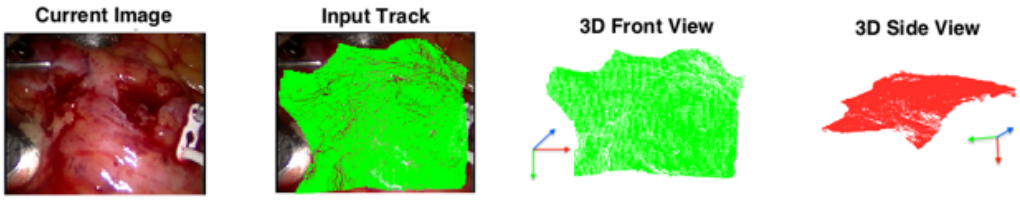
The proposed approach is not scalable to millions of feature tracks. Consequently, dense non-rigid structure from motion using the formulation discussed in this chapter is difficult. The major computational complexity stems due to the calculation of clustering matrix which is of the dimension $P \times P$. Additionally, we assumed orthographic camera projection which has its own limitations to approximate real world scenes.



(a) Back sequence

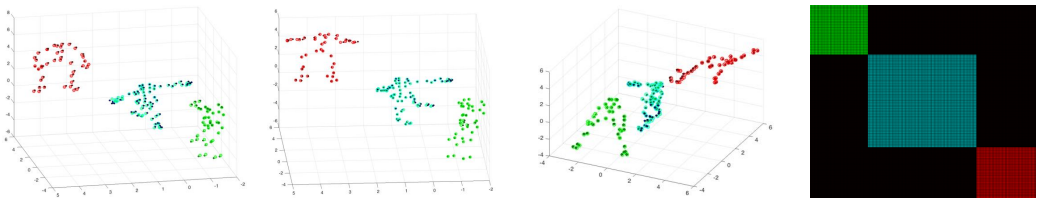


(b) Face sequence



(c) Heart sequence

Figure 3.14: (a), (b), (c) shows the 3D reconstruction obtained on the Back, Face and Heart sequences respectively. Here, 2D trajectories are shown over the images to give more intuitive representation of the obtained structure. These results were obtained on uniformly sampled feature tracks. The number of feature points used for reconstruction of the Back, Face and Heart sequence are 2281, 3146 and 7546 respectively. (Best viewed in color)



(a) Scene with 3 subject **(b)** Scene with 3 subject **(c)** Scene with 3 subject **(d)** Block Diagonal Matrix

Figure 3.15: (a)-(c) 3D reconstruction with segmentation results in a three subject scene taken from MoCap dataset [6]. Our approach is able to reconstruct and segment each action such as stretch (red), dance (cyan) and yoga (green) faithfully with overall 3D reconstruction error of 0.0407. Here, different color corresponds to distinct deforming object, while dark and light color circles show ground-truth and reconstructed 3D coordinates respectively. (d) Affinity matrix obtained after spectral clustering [136]. (Best viewed in color)

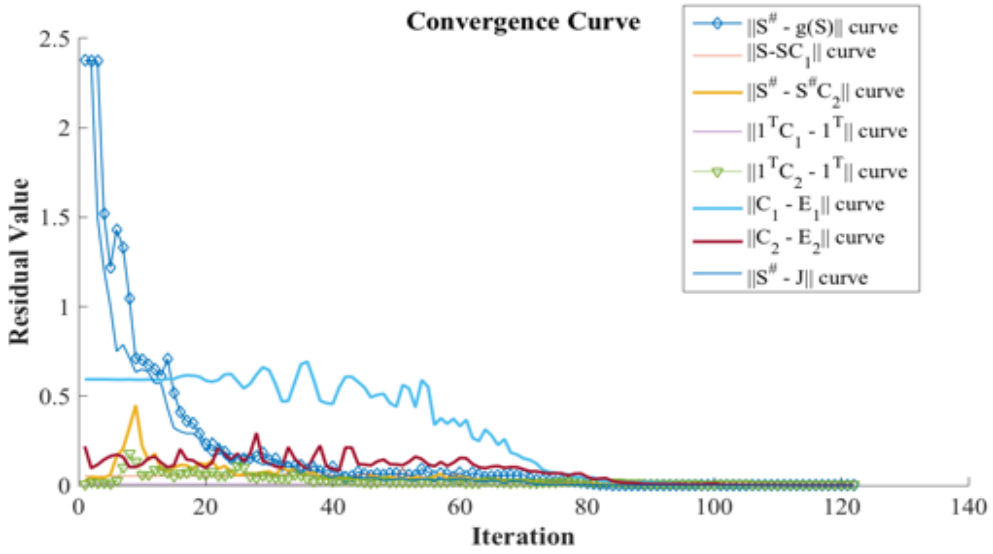
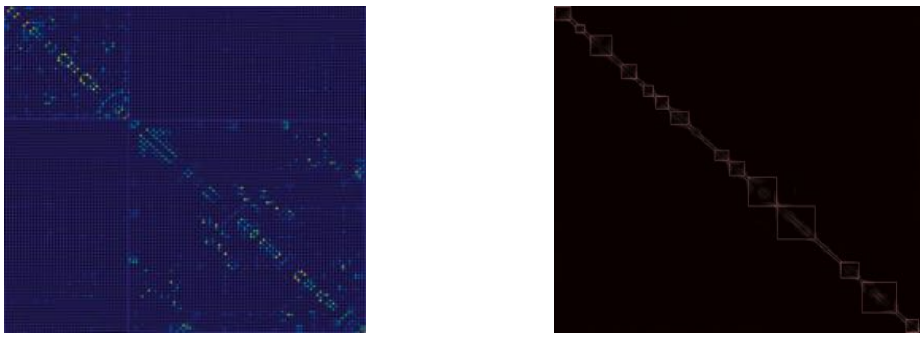


Figure 3.16: Convergence curve of the proposed optimization. Each curve represents the residual value associated with each terms shown in legends over iteration. (Best viewed in color)



(a) Spatial Affinity Matrix **(b)** Temporal Affinity Matrix

Figure 3.17: (a) Affinity matrix obtained on the "Dance + Yoga" Sequence. Clearly, it shows two block diagonal structure, corresponding to the two objects, which is an interesting observation during our experiment. Thus, number of deforming objects can be directly inferred from the affinity matrix. (b) Affinity matrix obtained with temporal clustering, it shows similar activities are encapsulated in the same block structure or captured in local subspace.

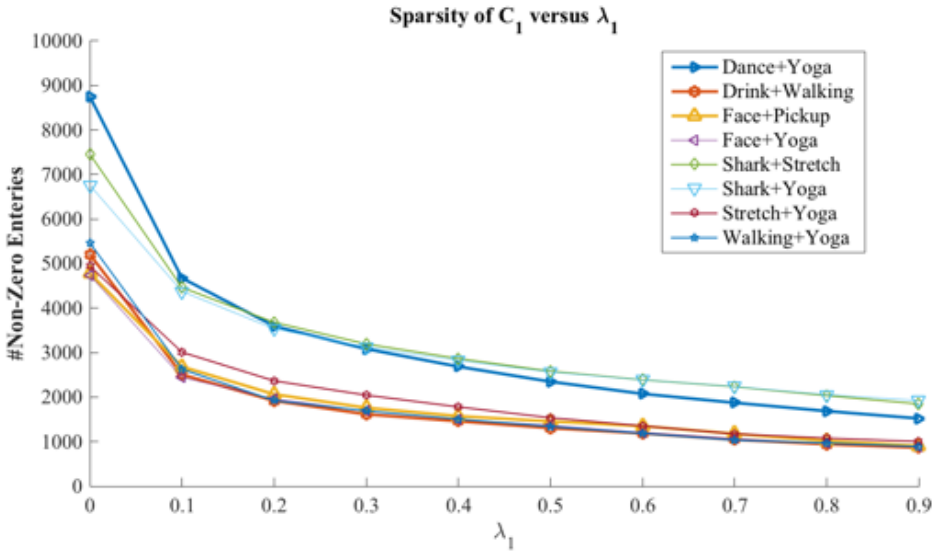


Figure 3.18: Sparsity of C_1 matrix vs λ_1 on different sparse data-set, it can be inferred that by using a proper value of λ_1 one can control the balance between sparsity and connectedness. Similar inference can be drawn for non-zero entries of C_2 with variation in λ_3. (Best viewed in color)

3.8 Closing Remarks

In this chapter, we described an algorithm to solve complex multi-body non-rigid structure from motion by exploiting spatio-temporal relation of the deforming shapes. This chapter demonstrated a new way to compactly represent deformable shapes. Despite being a non-convex formulation, we provide a solution to the resultant optimization using ADMM [19] which is effective, fast and easy to implement. We supply extensive experimental results on both synthetic and real benchmark datasets to test the method. The result demonstrate that the present approach outperforms the recent state-of-the-art non-rigid reconstruction methods by providing competitive 3D reconstruction and reliable object segmentation. Even though methods such as [42], [142], [176], [109] can handle simple variations of non-rigid deformation well, our approach provides robust reconstruction for both short and long-term complex multi-body deformations. In the next chapter, we will discuss the scalability issue with non-rigid structure from motion under factorization.

4

Scalable Dense Non-Rigid Structure from Motion

Contents

4.1 From sparse NRSFM to dense NRSFM

Non-rigid structure from motion algorithm discussed in the previous chapter has shown some promising results for sparse feature points. We also learned that the previous formulation suffers from the scalability issue. Additionally, the formulation proposed in chapter (3) assumes that the non-rigid object span a global linear space, and the spatial-temporal space spanned by these deforming objects lies in a union of Euclidean affine or linear subspace. However, in practice the features can be noisy and dense with underlying manifold structure [198]. These dense structure can be composed of several local linear subspace, hence single global linear subspace assumption does not hold. To overcome these limitation, this chapter addresses the task of solving a dense non-rigid structure from motion (NRSfM) problem. The algorithm introduced in this chapter can handle millions of points or does not ignore local non-linearities of surface deformation, and thus can reliably model complex non-rigid deformations. Our method propose a new approach for dense NRSfM by modeling the problem on a Grassmann manifold. Specifically, we assume the complex non-rigid deformations lie on a union of local linear subspaces both spatially and temporally. This naturally allows for a compact representation of the complex non-rigid deformation over frames. We provide experimental results on several synthetic and real benchmark datasets. The procured results clearly demonstrate that our current formulation, apart from being scalable and more accurate than state-of-the-art methods, is also more robust to noise and generalizes to highly non-linear deformations.

4.2 Introduction to dense NRSFM

Dense Non-rigid Structure from Motion (NRSfM) aims to recover 3D coordinates for every pixels of the deforming object. The existing solutions to *sparse* NRSfM cannot be employed directly to *dense* NRSFM as they do not scale to dense feature points and their resilience to noise remains unsatisfactory. Moreover, state-of-the-art algorithms [65, 41] to solve *dense* NRSfM are computationally expensive and rely on the assumption of global low-rank shape which, unfortunately, fails to cater the inherent local structure of the deforming shape over time. Consequently, to represent dense non-rigid structure under such formulations seems rather flimsy and implausible.

For many real-world applications, for instance, dense reconstruction of facial expressions from images, limitations such as scalability, timing, robustness, reliable modeling, *etc*, are of crucial concern. Despite these limitations —which are well-known to the researchers of this area, no template-free approach exists that can reliably deal with these concerns. In this chapter, we will learn a template-free dense NRSfM algorithm that overcomes these difficulties. As a first step to overcome these difficulties, we reduce the overall high-dimensional

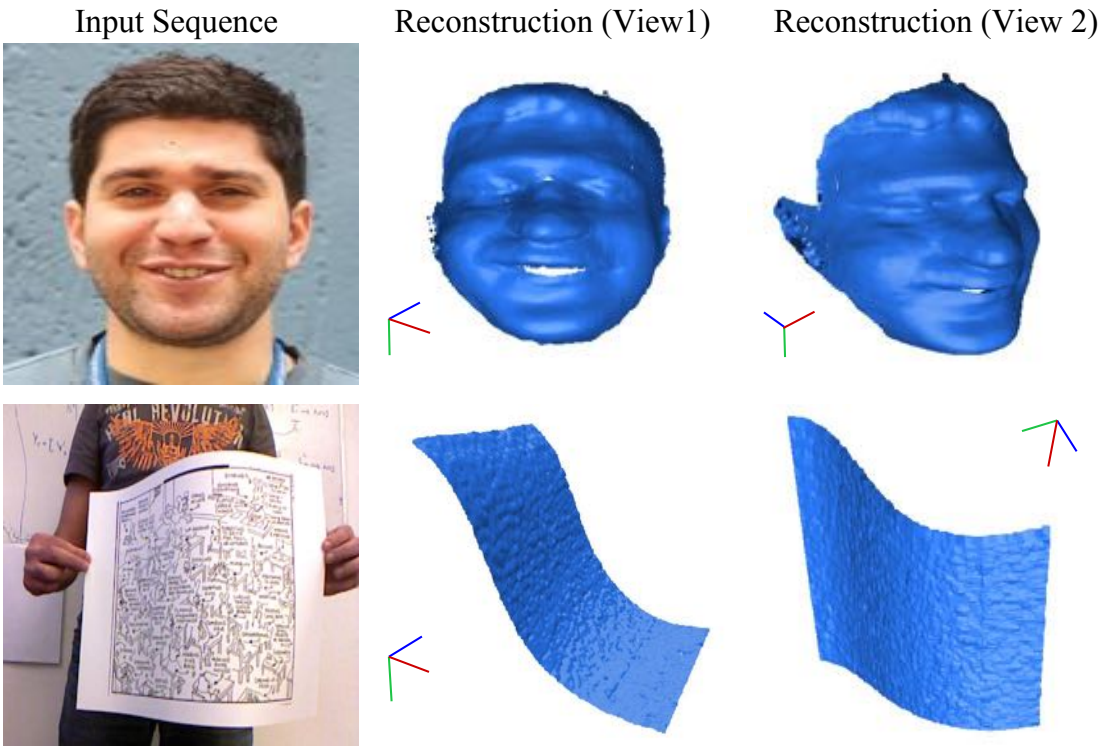


Figure 4.1: Our algorithm takes dense long-term 2D trajectories of a non-rigid deforming object as input, and provides a dense detailed 3D reconstruction of it. The reconstructed surface captures the complex non-linear motion which can be helpful for real world applications such as 3D virtual and augmented reality. Example frames are taken from publicly available real datasets: real face sequence[65] and kinect_paper sequence[183] respectively.

non-linear space spanned by the deforming shape as a union of several local low-dimensional linear subspaces. Our approach is based on a simple idea/assumption *i.e.*, any complex deforming surface can be approximated by a locally linear subspace structure [37]. We use this simple intuition in a spatio-temporal framework to solve dense NRSfM. This choice naturally leads to a few legitimate queries:

a) Why spatio-temporal framework for solving dense NRSfM? Spatio-temporal framework discussed in the previous chapter has exhibited the state-of-the-art results in NRSfM challenge [9, 93]. Even though the concept behind such a framework is elementary, no algorithm to our knowledge exists that exploit such an intrinsic idea for dense NRSfM.

b) Why the previously proposed spatio-temporal methods are unable to handle dense NRSfM? The formulation discussed in the previous chapter in inspired from SSC algorithm [52]. As

a result, the complexity of their formulations grows exponentially in the order of the number of data points. This makes it difficult to solve dense NRSfM using such formulation. Moreover, these methods ([107, 205]) use an assumption that non-rigid shape should lie on a low-dimensional linear or affine subspace globally. In reality, such an assumption does not hold for all kinds of non-linear deformations [191, 148]. Although a recent spatio-temporal method proposed by Dai *et al.*[41] can solve this task, however, it involves a series of least square problems to be solved, which is computationally very demanding.

To overcome all these issues, this work introduce a spatio-temporal dense NRSfM algorithm which is free from previous chapter limitations. In this chapter, we adhere to the assumption that the low-dimensional linear subspace spanned by a deforming shape is locally valid. Such assumptions about shapes have been well studied in topological manifold theory [1, 48]. The Grassmann manifold is a topologically rich non-linear manifold, each point of which represents the set of all right-invariant subspaces of a Euclidean space. One property of the Grassmannian that is particularly useful in our setting is that the points in it can be embedded into the space of symmetric matrices. This property has been used in several computer vision applications that deals with subspace representation of data [80, 28]. Accordingly, in our problem, to model a non-linear shape, using a Grassmannian allows us to represent the shape as a set of "smooth" low-dimensional surfaces embedded in a higher dimensional Euclidean space. Such a representation not only reduces the complexity of our task but also makes our formulation robust and scalable as described below.
c) Why Grassmann manifold? It is well-known that the complex non-rigid deformations are composed of multiple subspaces that quite often fit a higher-order parametric model [144, 154, 205]. To handle such complex models globally can be very challenging – both numerically and computationally. Consequently, for an appropriate representation of such a model, we decompose the overall non-linearity of the shape by a set of locally linear models that span a low-rank subspace of a vector space. As alluded to above, the space of all d-dimensional linear subspaces of $\mathbb{R}^N$ ($0 < d < N$) forms the Grassmann manifold [1, 2]. By modeling the deformation on this manifold allows us to operate on the number of subspaces rather than on the number of vectorial data points (on the shape), which reduces the complexity of the problem significantly. Moreover, since each local surface is a low-rank subspace, it can be faithfully reconstructed using a few eigenvalues and corresponding eigenvectors, which makes such representation scalable and robust to noise.

The aforementioned properties of the Grassmannian perfectly fit our strategy to model complex deformations, and therefore, we blend the concept of spatio-temporal representations with local low-rank linear models. This idea results in a two-stage coupled optimization problem *i.e.* local reconstruction and global grouping, which is solved efficiently using the standard ADMM algorithm [19]. As the local reconstructions are performed using a

low-rank eigen decomposition, our representation is computationally efficient and robust to noise. We demonstrate the benefit of our approach to benchmark real and synthetic sequences §4.6. Our experimental results show that our method outperforms previous state-of-the-art approaches by 1-2 % on all the benchmark datasets. Before we provide the details of our algorithm, we review some pertinent previous works in the next section.

4.3 Background

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

Preliminaries: Given 'P' feature points over 'F' frames, we represent $W \in \mathbb{R}^{2F\times P}$, $S \in \mathbb{R}^{3F\times P}$, $R \in \mathbb{R}^{2F\times 3F}$ as the measurement, the shape, and the rotation matrices, respectively. Here R matrix is composed of block diagonal $R_i \in \mathbb{R}^{2\times 3}$, representing per frame orthographic camera projection. Also, the notation $S^\sharp \in \mathbb{R}^{3P\times F}$ stands for the rearranged shape matrix, which is a linear mapping of S. We use $\|\,.\,\|_F$ and $\|\,.\,\|_*$ to denote the Frobenius norm and the nuclear norm, respectively.

(a) Dai *et al.'s* [44] $\min_{S^\sharp,E} \|S^\sharp\|_* + \lambda\|E\|_F^2$ subject to: $W = RS + E$	(b) Zhu *et al.'s* [205] $\min_{S^\sharp,C,E} \|C\|_* + \gamma\|S^\sharp\|_* + \lambda\|E\|_1$ subject to: $S^\sharp = S^\sharp C, W = RS + E$
(c) Kumar *et al.'s* [107] $\min_{S,S^\sharp,C_1,C_2} \frac{1}{2}\|W - RS\|_F^2 + \lambda_1\|C_1\|_1 + \lambda_2\|S^\sharp\|_* + \lambda_3\|C_2\|_1$ subject to: $S = SC_1, S^\sharp = S^\sharp C_2, 1^T C_1 = 1^T, 1^T C_2 = 1^T$, $diag(C_1) = 0, diag(C_2) = 0, S^\sharp = g(S)$	(d) Garg *et al.'s*[65] $\min_{S,R} \lambda\frac{\|W-RS\|_F^2}{2} + \sum_{f,i,p} \|\nabla S_f^i(p)\| + \tau\|S^\sharp\|_*$ subject to: $R \in \mathbb{SO}(3)$

Table 4.1: A brief summary of formulation used by some of the recent approaches to solve sparse and dense NRSfM which are closely related to our method. Among all these four methods only Garg *et al.'s* [65] approach is formulated particularly for solving dense NRSfM.

4.3.1 Relevant Previous Work

Dai et al.'s approach: Dai *et al.* proposed a simple and elegant solution to NRSfM [44]. The work, dubbed "prior-free", provides a practical solution as well as new theoretical insights to NRSfM. Their formulation involves nuclear norm minimization on $S^\sharp$ instead of S –see Table 4.1(a). This is enforced due to the fact that $3K$ rank bound on S is weaker than K rank bound on $S^\sharp$, where K refers to the rank of S. Although this elegant framework provides robust results for the shapes that span a single subspace, it may perform poorly on complex non-rigid motions [205].

Zhu et al.'s approach: To achieve better 3D reconstructions on complex non-rigid sequences, this work capitalized on the limitations of Dai *et al.*'s work[44] by exploiting the union of subspaces in the shape space [205]. The proposed formulation is inspired by LRR [124] in conjunction with Dai *et al.*work –see Table 4.1(b). In the formulation, $C \in \mathbb{R}^{F\times F}$, $E \in \mathbb{R}^{2F\times P}$ are the coefficient and error matrices.

Kumar et al.'s approach: The work is discussed in previous chapter. It exploits multiple subspaces both in the trajectory space and in the shape space [107]. This work demonstrated empirically that procuring multiple subspaces in the trajectory and shape spaces provide better reconstruction results. The work proposed a joint segmentation and reconstruction framework, where segmentation inherently benefits reconstruction and vice-versa –see Table 4.1(c). In their formulation $C_1 \in \mathbb{R}^{P\times P}$, $C_2 \in \mathbb{R}^{F\times F}$ are the coefficient matrices and, $g(.)$ linearly maps S to $S^\sharp$.

Dense NRSfM approach: Garg *et al.* developed a variational approach to solve dense NRSfM [65]. The optimization framework proposed by them employs total variational constraint on the deforming shape ($\nabla S_f^i(p)$) to allow edge preserving discontinuities, and trace norm constraints to penalize the number of independent shapes –see Table 4.1(d). Recently, Dai *et al.* has also proposed a dense NRSfM algorithm with a spatio-temporal formulation [41].

4.3.2 Motivation

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

1. To solve dense NRSfM using the formulation discussed in the previous chapter is nearly impractical due to complexity of the formulation §4.2. Also, the error measure used by it is composed of Euclidean norm defined on the original data (see Table 4.1), which is not proper for non-linear data with a manifold structure [1, 189].

2. The algorithm proposed by Garg *et al.* [65] results in a biconvex formulation, which is computationally expensive and needs a GPU to speed up the implementation. Similarly, Dai *et al.*'s recent work[41] is computationally expensive as well due to costly gradient term in their formulation.

3. Methods such as [200, 123] rely on the template prior for dense 3D reconstruction of the object. Other piecewise approach for solving dense NRSfM [150] require a post-processing step to stitch all the local reconstructions.

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

1. A scalable spatio-temporal framework on the Grassmann manifold to solve dense NRSfM which does not need any template prior.
2. An effective framework that can handle non-linear deformations even with noisy trajectories and provides state-of-the-art results on benchmark datasets.
3. An efficient solution to the proposed optimization based on the ADMM procedure [19].

4.4 Problem Formulation

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

4.4.1 Grassmann Manifold

The Grassmann manifold, usually denoted as $\mathcal{G}(n, r)$, consists of all r-dimensional linear subspaces of $\mathbb{R}^n$, where $n > r$. A point on the Grassmann manifold is represented by a $n \times r$ matrix (say X), whose columns are composed of orthonormal basis of the subspace spanned by X, denoted as span(X) or in short as $[X]$. Let's suppose $[X_1]$, $[X_2]$ are two such points on this manifold, then among several similarity distances known for this manifold [80], we will be using the *projection metric* distance given by $d_g([X_1], [X_2]) = \frac{1}{\sqrt{2}}\|X_1X_1^T - X_2X_2^T\|_F$, as it allows directly embedding the Grassmannian points into a Euclidean space (and the use of the Frobenius norm) using the mapping $X \to XX^T$. With this metric, $(\mathcal{G}, d_g)$ forms a metric space. Interested readers may refer to [80] for details.

4.4.2 Formulation

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

$$W_s = RS_s \tag{4.1}$$

where, $W_s \in \mathbb{R}^{2F\times P}$, $R = \text{blkdiag}(R_1, ..., R_F) \in \mathbb{R}^{2F\times 3F}$, $S_s \in \mathbb{R}^{3F\times P}$. The motive here is, given the input measurement matrix, solve for rotation (R) and 3D shape (S_s). To serve this objective, Eq.(4.1) maintains the camera motion and the shape deformation such that it

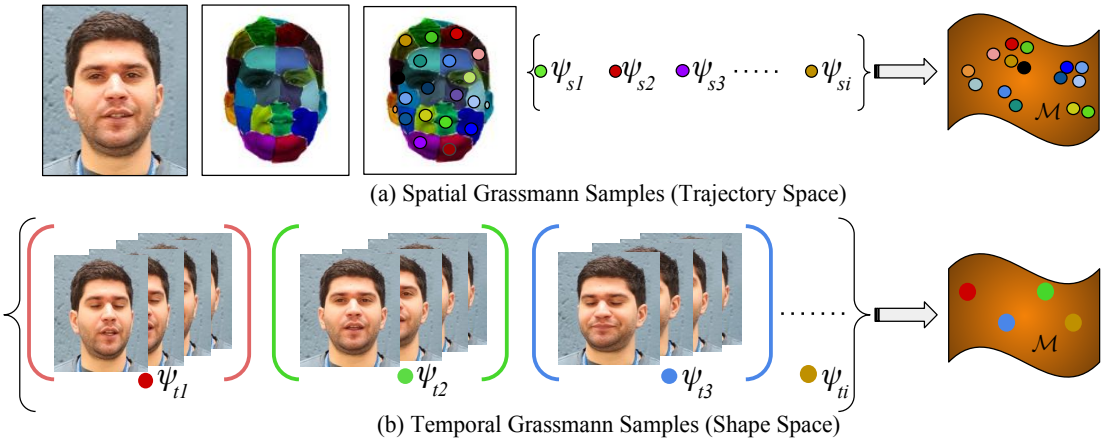


Figure 4.2: Conceptual illustration of data point representation on the grassmann manifold. Each local subspace can equivalently be represented by a single point on the manifold. Top row: Construction of grassmann samples in the trajectory space using spatial information. Bottom row: Construction of grassmann samples in the shape space by partitioning the shapes in a sequential order over frames.

complies with the image measurements. For our method, we solve for rotations using the Intersection method [44] by assuming that the multiple non-rigid motions within a single deforming object, over frames, can be faithfully approximated by per frame single relative camera motion with a higher rank*. Accordingly, our goal reduces to develop a systematic approach that can reliably explain the non-rigid shape deformations and provides better 3D reconstruction. We use subscript 's' in Eq.(4.1) to indicate that the column permutations of S_s and W_s matrix are allowed. Alternatively, the column permutations of $S_t^\sharp$ is inadmissible as it results in the discontinuity of the trajectories over frames.

Grassmannian Representations in Trajectory Space:

Let's suppose $\Psi_s = \{\psi_{s1}, \psi_{s2}, .., \psi_{sK_s}\}$ is the set of points on the Grassmann manifold generated using S_s matrix, then $\mathcal{T}_s = \{(\psi_{s1})(\psi_{s1})^T, (\psi_{s2})((\psi_{s2})^T ..., (\psi_{sK_s})(\psi_{sK_s})^T\}$ represents a tensor which is constructed by mapping all symmetric matrices of the Grassmann data points —refer Figure 4.2(a). As discussed before in §4.2, to explain the complex deformations, we reduce the overall non-linear space as a union of several local low-dimensional linear spaces which form the sample points on the Grassmann manifold. But, the notion of self-expressiveness is valid only for Euclidean linear or affine subspace. To apply self-expressiveness on the Grassmann manifold one has to adopt linearity onto the manifold. Since, Grassmann manifold is isometrically equivalent to the symmetric idempotent matrices [33], we embed

*Check the Appendix (C) for a detail discussion on rotation.

the Grassmann manifold into the symmetric matrix manifold, where the self-expression can be defined in the embedding space. This leads to the following optimization:

$$\begin{aligned} &\underset{E_s, C_s}{\text{minimize}} \; \|E_s\|_F^2 + \lambda_1 \|C_s\|_* \\ &\textit{subject to: } \mathcal{T}_s = \mathcal{T}_s C_s + E_s \end{aligned} \tag{4.2}$$

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

Grassmannian Representations in Shape Space:

Deforming object attains different state over time which adheres to distinct temporal local subspaces [107]. Assuming that the temporal deformation is smooth over-time, we express deforming shapes in terms of local self-expressiveness across frames as:

$$\begin{aligned} &\underset{E_t, C_t}{\text{minimize}} \; \|E_t\|_F^2 + \lambda_2 \|C_t\|_* \\ &\textit{subject to:} \mathcal{T}_t = \mathcal{T}_t C_t + E_t \end{aligned} \tag{4.3}$$

Similarly, $\mathcal{T}_t$ is the set of all symmetric matrices constructed using a set of Grassmannian samples Ψ_t, where Ψ_t contains the samples which are drawn from $S_t^\sharp \in \mathbb{R}^{3P \times F}$ —refer Figure 4.2(b). 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.

Spatio-Temporal Formulation:

Combining the above two objectives and their constraints with reprojection error term give us our formulation. Our representation blends the local subspaces structure along with

the global composition of a non-rigid shape. Thus, the overall objective is:

$$\begin{aligned}
&\underset{S_s, S_t^\sharp, E_s, E_t, C_s, C_t}{\text{minimize}} \; \mathrm{E} = \frac{1}{2}\|W_s - RS_s\|_F^2 + \gamma\|S_t^\sharp\|_* + \lambda_1\|E_s\|_F^2 + \lambda_2\|E_t\|_F^2 + \lambda_3\|C_s\|_* + \lambda_4\|C_t\|_* \\
&\textit{subject to:} \\
&\mathcal{T}_s = \mathcal{T}_s C_s + E_s; \mathcal{T}_t = \mathcal{T}_t C_t + E_t; \\
&\Psi_s = \xi(C_s, S_s, q); \Psi_t = \xi(C_t, S_t^\sharp, q); \\
&S_s = \zeta\big(\Psi_s, \Sigma_s, V_s, N_s\big); S_t^\sharp = \zeta\big(\Psi_t, \Sigma_t^\sharp, V_t, N_t\big); \\
&S_t^\sharp = \mathscr{T}_1(S_s); W_s = \mathscr{T}_2(W_s, S_s);
\end{aligned} \tag{4.4}$$

The re-projection error constraint performs the 3D reconstruction using W_s and R. Meanwhile, the local subspace grouping naturally enforces the union of subspace structure in S_s, $S_t^\sharp$ with corresponding low-rank representations of the coefficient matrices C_s and C_t. Here, the function $\xi(.)$ draws inference from C matrices to refine Grassmannian sample set, both in trajectory and shape spaces. The function $\zeta(.)$ reconstructs S_s and $S_t^\sharp$ matrices based on a set of local subspaces (Ψ_s, Ψ_t, V_s, V_t), singular values (Σ_s, Σ_t) and the number of top eigenvalues (N_s, N_t). The function $\mathscr{T}_1(.)$ transforms $S_s \in \mathbb{R}^{3F\times P}$ matrix to $S_t^\sharp \in \mathbb{R}^{3P\times F}$ matrix and $\mathscr{T}_2(.)$ function rearranges W_s matrix as per the recent ordering of S_s[†]. Parameters such as 'q', 'N_s' and 'N_t' provides the flexibility to handle noise and adjust computations. Note that the element of the sets Ψ_s, Ψ_t, V_s and V_t are obtained using SVD (Singular Value Decomposition). The above equation *i.e.* Eq: (4.4) is a coupled optimization problem where the solution to S matrices influence the solution of C matrices and vice-versa, and $\mathscr{T}_1()$ connects $S_t^\sharp$ to S_s.

4.5 Solution

The formulation in Eq.(4.4) is a non-convex problem due to the bilinear optimization variables ($\mathcal{T}_s C_s$, $\mathcal{T}_t C_t$), hence a global optimal solution is hard to achieve. However, it can be efficiently solved using Augmented Lagrangian Methods (ALMs) [19], which has proven its effectiveness for many non-convex problems. Introducing Lagrange multipliers ($\{Y_i\}_{i=1}^3$)

[†]It's important to keep track of column permutation of W_s, S_s.

and auxiliary variables (J_s, J_t) to Eq.(4.4) gives us the complete cost function as follows:

$$\begin{aligned}
&\underset{S_s,S_t^\sharp,C_s,C_t,J_s,J_t}{\text{minimize}}\ \mathrm{E} = \frac{1}{2}\|W_s - RS_s\|_F^2 + \frac{\beta}{2}\|S_t^\sharp - \mathscr{T}_1(S_s)\|_F^2 + < Y_1, S_t^\sharp - \mathscr{T}_1(S_s) > + \gamma\|S_t^\sharp\|_* \\
&+ \lambda_1\|\mathcal{T}_s - \mathcal{T}_s C_s\|_F^2 + \lambda_3\|J_s\|_* + \frac{\beta}{2}\|C_s - J_s\|_F^2 + < Y_2, C_s - J_s > + \lambda_2\|\mathcal{T}_t - \mathcal{T}_t C_t\|_F^2 \\
&+ \lambda_4\|J_t\|_* + \frac{\beta}{2}\|C_t - J_t\|_F^2 + < Y_3, C_t - J_t > \qquad (4.5) \\
&\textit{subject to: } \Psi_s = \xi(C_s, S_s, q); \Psi_t = \xi(C_t, S_t^\sharp, q); \\
&\qquad S_s = \zeta(\Psi_s, \Sigma_s, V_s, N_s); S_t^\sharp = \zeta(\Psi_t, \Sigma_t^\sharp, V_t, N_t); \\
&\qquad W_s = \mathscr{T}_2(W_s, S_s);
\end{aligned}$$

The function $\xi(.)$ first computes the SVD of C matrices, *i.e.*$C = [U_c, \Sigma_c, V_c]$, then forms a matrix $\mathcal{A}$ such that $\mathcal{A}_{ij} = [XX^T]_{ij}^q$, where '$q$' is set empirically based on noise levels and $X = U_c(\Sigma_c)^{0.5}$ (normalized). Secondly, it uses $\mathcal{A}_{ij}$ to form new Grassmann samples from the S matrices. Notice that $\xi(.)$ operates on C matrices whose dimensions depend on the number of Grassmann samples. This reduces the complexity of the task from exponential in the number of vectorial points to exponential in the number of linear subspaces. The later being of the order 10-50, where as the former can go more than 50,000 for dense NRSfM.

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

4.6 Experiments and Results

We compare the performance of our method against four previously reported state-of-the-art approaches, namely Dense Spatio-Temporal DS [41], Dense Variational DV [65], Trajectory Basis PTA [7] and Metric Projection MP [143]. To test the performance, we used dense NRSfM dataset introduced by Garg *et al.* [65] and Varol *et al.* [183] under noisy and noise free conditions. For quantitative evaluation of 3D reconstruction, we align the estimated shape S_{est}^t with ground-truth shape S_{GT}^t per frame using Procrustes analysis. We compute the average RMS 3D reconstruction error as $e_{3D} = \frac{1}{F}\sum_{t=1}^{F} \frac{\|S_{est}^t - S_{GT}^t\|_F}{\|S_{GT}^t\|_F}$. We used Kmeans++ algorithm [11] to initialize segments without disturbing the temporal continuity.

Algorithm 2 Scalable Dense Non-Rigid Structure from Motion: A Grassmannian Perspective

Require: W_s, R using [44], λ_1, λ_2, λ_3, λ_4, γ, $\varrho = 1.1$, $\beta = 1e^{-3}$, $\beta_m = 1e^{6}$, $\varepsilon = 1e^{-12}$, K_s, K_t.

Initialize: S_s = pseudoinverse(R)W_s and $S_t^\sharp = \mathscr{T}_1(S_s)$.

Initialize: 'K_t' temporal Grassmannians using '$S_t^\sharp$' matrix, $\Psi_t = \{\psi_{ti}\}_{i=1}^{K_t}$.

Initialize: 'K_s' spatial Grassmannians using 'S_s' matrix, $\Psi_s = \{\psi_{si}\}_{i=1}^{K_s}$.

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

Initialize: $\Omega_{ij}^s = trace[\left(\psi_{sj}^T\psi_{si}\right)\left(\psi_{si}^T\psi_{sj}\right)]$, $\Omega_{ij}^t = trace[\left(\psi_{tj}^T\psi_{ti}\right)\left(\psi_{ti}^T\psi_{tj}\right)]$, $\Omega_s = (\Omega_{ij}^s)_{i,j=1}^{K_s}$, $\Omega_t = (\Omega_{ij}^t)_{i,j=1}^{K_t}$, $L_sL_s^T$ = Cholesky(Ω_s), $L_tL_t^T$ = Cholesky(Ω_t)

while not converged do

1: $S_s \leftarrow (R^TR + \beta I)^{-1}\Big(\beta\big(\mathscr{T}_1^{-1}(S_t^\sharp) + \mathscr{T}_1^{-1}(Y_1)/\beta\big) + R^TW_s\Big)$

2: $C_s \leftarrow \Big(2\lambda_1 L_sL_s^T + \beta(J_s - Y_2/\beta)\Big)\left(2\lambda_1 L_sL_s^T + \beta I_s\right)^{-1}$

3: $\Psi_s \leftarrow \xi(C_s, S_s, q)$ {Update spatial Grassmann points}

4: $S_s \leftarrow \zeta(\Psi_s, \Sigma_s, V_s, \mathcal{N}_s)$; {refine based on top $\mathcal{N}_s$ eigen value}

5: $J_s \leftarrow U_{J_s}\mathcal{S}[\lambda_3/\beta](\Sigma_{J_s})V_{J_s}$, where $[U_{J_s}, \Sigma_{J_s}, V_{J_s}] = svd(C_s + Y_2/\beta)$ and $\mathcal{S}[\tau](x)$ = sign(x)max(|x|-τ, 0)

6: $S_t^\sharp \leftarrow U_t\mathcal{S}[\gamma/\beta](\Sigma_t)V_t$, where $[U_t, \Sigma_t, V_t] = svd(\mathscr{T}_1(S_s) - Y_1/\beta)$ and $\mathcal{S}[\tau](x)$ = sign(x)max(|x|-τ, 0)

7: $C_t \leftarrow \Big(2\lambda_2 L_tL_t^T + \beta(J_t - Y_3/\beta)\Big)\left(2\lambda_2 L_tL_t^T + \beta I_t\right)^{-1}$

8: $\Psi_t \leftarrow \xi(C_t, S_t^\sharp, q)$ {Update temporal Grassmann points}

9: $S_t^\sharp \leftarrow \zeta(\Psi_t, \Sigma_t, V_t, \mathcal{N}_t)$; {refine based on top $\mathcal{N}_t$ eigen value}

10: $J_t \leftarrow U_{J_t}\mathcal{S}[\lambda_4/\beta](\Sigma_{J_t})V_{J_t}$, where $[U_{J_t}, \Sigma_{J_t}, V_{J_t}] = svd(C_t + Y_3/\beta)$ and $\mathcal{S}[\tau](x)$ = sign(x)max(|x|-τ, 0)

11: $\Omega_{ij}^s \leftarrow trace[\left(\psi_{sj}^T\psi_{si}\right)\left(\psi_{si}^T\psi_{sj}\right)]$, $\Omega_{ij}^t \leftarrow trace[\left(\psi_{tj}^T\psi_{ti}\right)\left(\psi_{ti}^T\psi_{tj}\right)]$;

12: $\Omega_s \leftarrow (\Omega_{ij}^s)_{i,j=1}^{K_s}$, $\Omega_t \leftarrow (\Omega_{ij}^t)_{i,j=1}^{K_t}$; {$\Omega_s \succeq 0$, $\Omega_t \succeq 0$, if $\Omega_s||\Omega_t = 0$ add $\delth$I to make it $\succ 0$ (see Appendix (C))}

13: $L_sL_s^T$ = Cholesky(Ω_s), $L_tL_t^T$ = Cholesky(Ω_t);

14: $W_s \leftarrow \mathscr{T}_2(W_s, S_s)$ {Note: Column permutation for W_s and S_s should be same.}

15: $Y_1 := Y_1 + \beta(S_t^\sharp - \mathscr{T}_1(S_s))$, $Y_2 := Y_2 + \beta(C_s - J_s)$, $Y_3 := Y_3 + \beta(C_t - J_t)$; {Update Lagrange multipliers}

16: $\beta \leftarrow$ minimum($\varrho\beta, \beta_m$)

17: maxgap := maximum($[\|S_t^\sharp - \mathscr{T}_1(S_s)\|_\infty, \|C_s - J_s\|_\infty, \|C_t - J_t\|_\infty]$)

if (maxgap $< \varepsilon$ || $\beta > \beta_m$) then

break;

end if{check for the convergence}

end while{Note: $\delth$ is a very small positive number and I symbolizes identity matrix}.

Ensure: S_s, S_t, C_s, C_t. {Note: Kindly use economical version of *svd* on a regular desktop.}

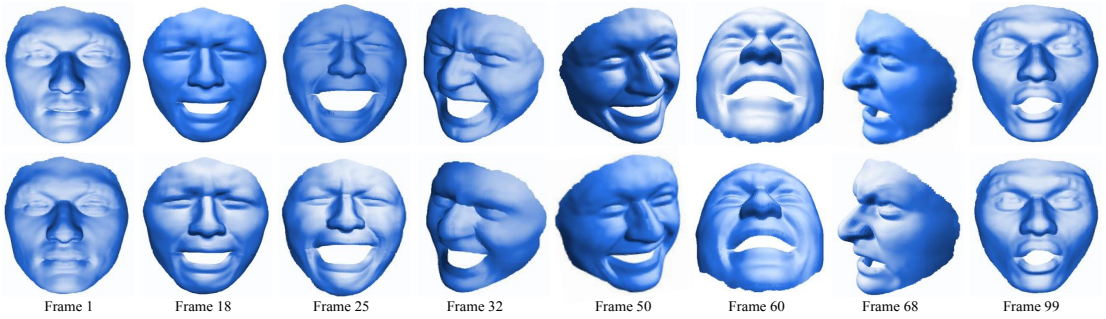


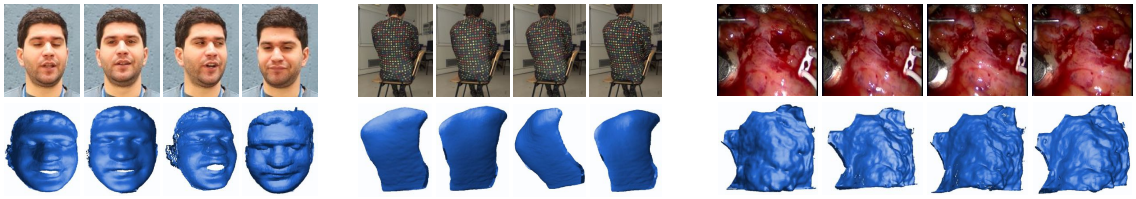
Figure 4.3: Reconstruction results obtained on synthetic dense face dataset (face sequence 4). Top row : Ground-truth 3D points, Bottom row : Recovered 3D points using our approach.

Experiments on Synthetic Face Sequences: This dataset consists of 4 different face sequence with 28,880 feature points tracked over multiple frames. The face sequence 1, 2 is a 10 frame long video, whereas, face sequence 3, 4 is a 99 frame long video. It's a challenging dataset mainly due to different rotation frequencies and deformations in each of the sequence. Figure 4.3 shows the qualitative reconstruction results obtained using our approach in comparison to the ground-truth for face sequence 4. Table (4.2) lists the performance comparisons of our method with other competing methods. Clearly, our algorithm outperforms the other baseline approach, which helps us to conclude that holistic approaches to rank minimization without drawing any inference from local subspace structure is a less effective framework to cope up with the local non-linearities.

Method	DS [41]	DV [65]	PTA [7]	MP [143]	Ours
Seq. 1	0.0636	0.0531	0.1559	0.2572	0.0443
Seq. 2	0.0569	0.0457	0.1503	0.0640	0.0381
Seq. 3	0.0374	0.0346	0.1252	0.0611	0.0294
Seq. 4	0.0428	0.0379	0.1348	0.0762	0.0309

Table 4.2: Average 3D reconstruction error (e_{3D}) comparison on dense synthetic face sequence[65]. Note: The code for DV [65] is not publicly available, we tabulated its results from DS [41] work.

Experiments on face, back and heart sequence: This dataset contains monocular videos of human facial expressions, back deformations, and beating heart under natural lighting conditions. The face sequence, back sequence, and heart sequence are composed of 28332, 20561, and 68295 feature points tracked over 120, 150, and 80 images, respectively. Unfortunately, due to the lack of ground-truth 3D data, we are unable to quantify the performance of these sequences. Fig. (4.4) show some qualitative results obtained using our algorithm on this real dataset.



(a) Face sequence **(b)** Back sequence **(c)** Heart sequence

Figure 4.4: Qualitative reconstruction results procured on benchmark real dense dataset [65] a) Face sequence (28,332 feature points over 120 frames) b) Back sequence (20,561 feature points over 150 frames) c) Heart sequence (68,295 feature points over 80 frames).

Figure 4.5: Reconstruction results on benchmark kinect_tshirt (74,000 points, 313 frames) and kinect_paper(58,000 points, 193 frames) dataset [183]. Top row: Input image frame. Bottom row: Dense 3D reconstruction for the corresponding frame using our approach.

Experiments on kinect_paper and kinect_tshirt sequence: To evaluate our performance on the real deforming surfaces, we used kinect_paper and kinect_tshirt dataset[183]. This dataset provides sparse SIFT[127] feature tracks along with dense 3D point clouds of the entire scene for each frame. Since, dense 2D tracks are not directly available with this dataset, we synthesized it. To obtain dense feature tracks, we considered the region within a window containing the deforming surface. Precisely, we considered the region within $xw = (253, 253, 508, 508)$, $yw = (132, 363, 363, 132)$ across 193 frames for paper sequence, and $xw = (203, 203, 468, 468)$, $yw = (112, 403, 403, 112)$ across 313 frames for tshirt sequence to obtain the measurement matrix [67, 63]. Fig.(4.5) show some qualitative results obtained using our method on this dataset. Table (4.3) lists the numerical comparison of our approach with other competing dense NRSfM approaches on this dataset.

Experiments on noisy data: To evaluate the robustness of our method to noise levels, we performed experiments by adding Gaussian noise under different standard deviations to the measurement matrix. Similar to DS [41] the standard deviations are incorporated as $\sigma_n = r \max\{|W_s|\}$ by varying r from 0.01 to 0.05. We repeated the experiment 10 times. Fig.(4.6a)

Method	DS [41]	DV [65]	PTA [7]	MP [143]	Ours
paper	0.0612	-	0.0918	0.0827	0.0394
tshirt	0.0636	-	0.0712	0.0741	0.0362

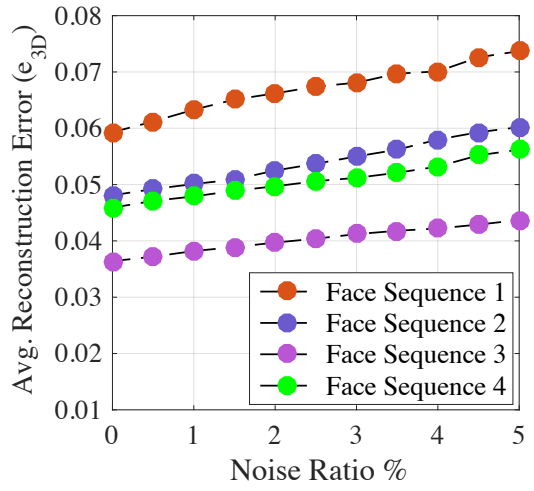
Table 4.3: Average 3D reconstruction error (e_{3D}) comparison on kinect_paper and kinect_tshirt [183] sequence. Note: The code for DV [65] is not publicly available. The pixels with no 3D data available were discarded for the experiments and the evaluation.

and Fig.(4.6b) show the variation in the performance of our method under different noise ratio's on synthetic face sequences[65] and kinect sequences[183] respectively. It can be inferred from the plot that even with large noise ratios, the average reconstruction error does not fluctuate significantly. This improvement is expected from our framework as it is susceptible only to top eigen values.

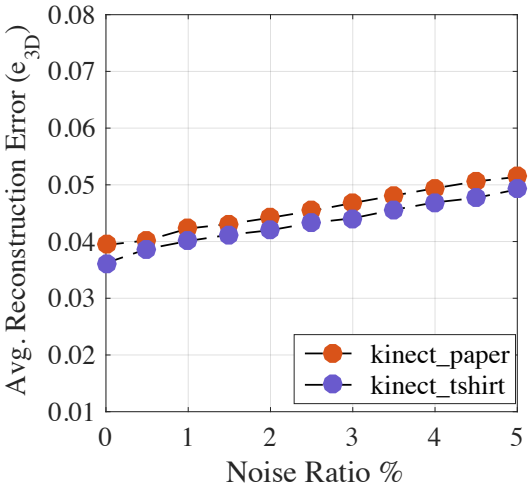
Effects of variable initialization on the overall performance: We performed several other experiments to study the behavior of the algorithm under different variable initializations. For easy exposition, we conducted this experiment on noise free sequences. We mainly investigated the behavior of N_s, N_t, K_s, K_t on the overall performance of our algorithm. Fig.(4.6c) and Fig.(4.6d) show the variations in the reconstruction errors with respect to N_s and K_s respectively. A similar trend in the plots is observed for changes on N_t and K_t values. These plots clearly illustrate the usefulness of our local low-rank structure *i.e.*, considering a small number of eigenvalues for every local structure is as good as considering all eigenvalues. Similarly, increasing the number of local subspaces after a certain value has negligible effect on the overall reconstruction error. Furthermore, we examined the form of C_s and C_t after convergence as shown Fig.(4.7a) and Fig.(4.7b). Due to the lack of ground-truth data on local subspaces, we could not quantify C_s and C_t. For qualitative analysis on the observation, kindly refer to the Appendix(C).

Ablation Analysis: This test is performed to evaluate the importance of spatial and temporal constraints in our formulation. To do this, we observe the performance of our formulation under four different setups: a) without any spatio-temporal constraint (NC), b) with only spatial constraint (SP), c) with only temporal constraint (TP), and d) with spatio-temporal constraint (Both). Fig.(4.7c) shows the variations in reconstruction errors under these setups on four synthetic face sequence. The statistics clearly illustrate the importance of both constraints in our formulation.

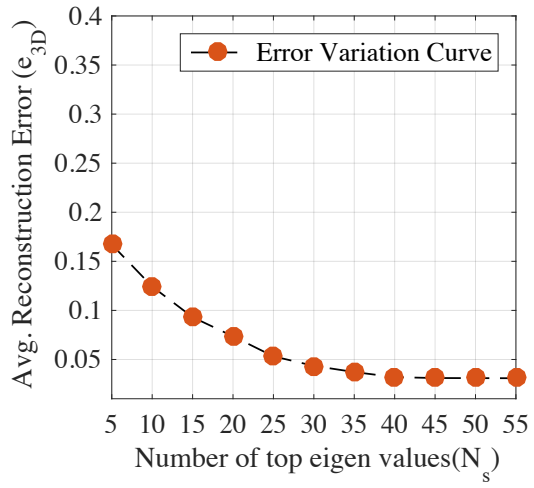
Runtime Analysis: To analyze the runtime performance of our approach, we used synthetic face, real paper, and tshirt sequence. This experiment is performed on a computer with an Intel core i7 processor and 16GB RAM. The script to compute the runtime is written in MATLAB 2016b. Fig.(4.7d) show the runtime comparisons of our approach with other dense NRSfM methods. The runtime reported in Fig.(4.7d) corresponds to the results



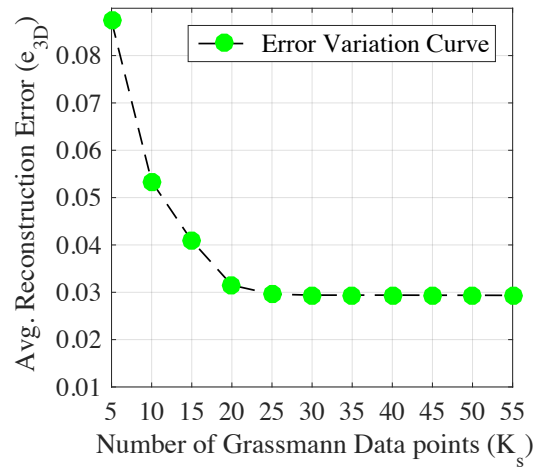
(a) Results on Noisy Trajectory for Face Sequence



(b) Results on Noisy Trajectory for kinect Sequence

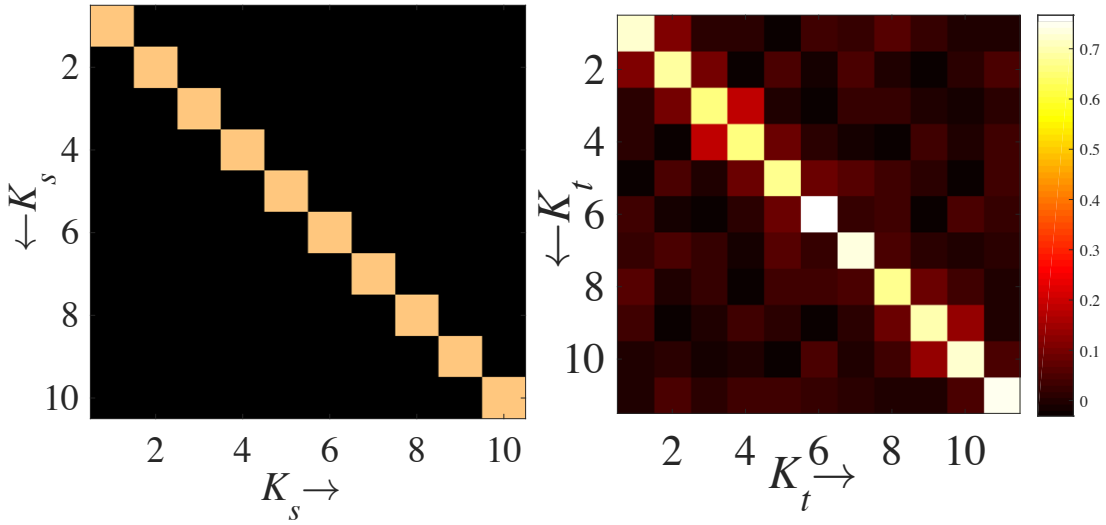


(c) Result with variation in no. of singular values



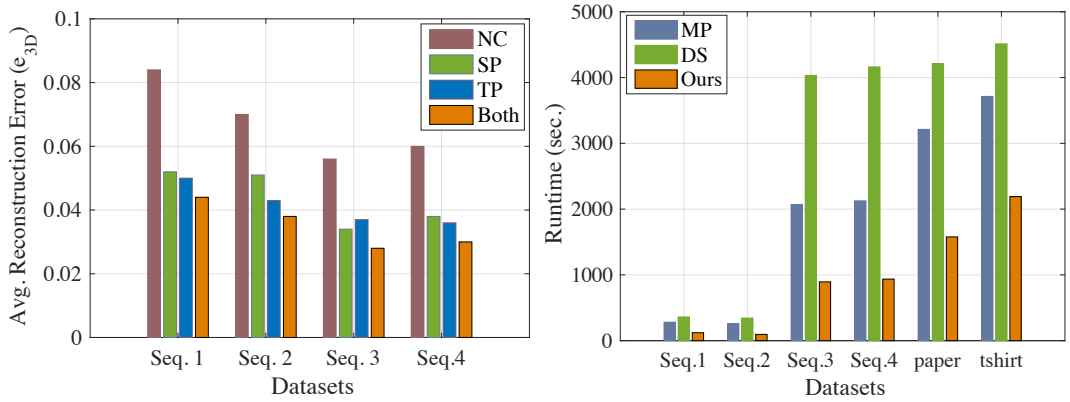
(d) Result with variation in no. of Grassmannians

Figure 4.6: (a)-(b) Avg. 3D reconstruction error (e_{3D}) variation with the change in the noise ratio for synthetic face sequence and kinect sequence respectively. (c)-(d) Variation in e_{3D} with the number of top eigen value and number of grassmann data points for Face Seq3.



(a) Spatial Coefficient Matrix

(b) Temporal Coefficient Matrix



(c) SP(Spatial Constraint), TP(Temporal Constraint)

(d) Computation time comparison

Figure 4.7: (a)-(b) A typical structure of $C_s \in \mathbb{R}^{K_s \times K_s}$, $C_t \in \mathbb{R}^{K_t \times K_t}$ after convergence. (c) Ablation test performance on the synthetic face sequence [65], NC(No spatial or temporal constraint), SP(only spatial constraint), TP(only temporal constraint), Both (both spatial and temporal constraint). (d) Runtime comparison of our method with MP [143] and a recent state-of-the-art dense NRSfM algorithm DS[41].

listed in Table (4.2), 4.3. The results clearly show the *scalability* of our method on datasets with more than 50,000-70,000 points. Despite PTA [7] is faster than our approach, its reconstruction accuracy suffers by a large margin for dense NRSfM (see Table 4.2, 4.3).

4.7 Chapter Outcome

In this chapter, we have introduced a scalable dense NRSfM algorithm which efficiently models the complex non-linear deformations. We achieved this by exploiting the non-linearity on the Grassmann manifold via spatiotemporal formulation. Moreover, we provided an efficient ADMM [19] based solution for solving our optimization. Several experiments on benchmark datasets are provided which clearly show the usefulness of our method. The proposed algorithm provides a new insight to model dense NRSfM which previously seems inconceivable under spatiotemporal formulation. We believe that in practice such a framework will be helpful to interesting 3D-vision applications.

5

Geometry Aware Dense Non-Rigid Structure from Motion

Contents

5.1 Motivation

Given dense image feature correspondences of a non-rigidly moving object across multiple frames, the goal is to develop an algorithm that provide accurate 3D shape for each frame. In the previous chapter, we developed a foundation to solve this task using Grassmannian representation. Unfortunately, the method we proposed has some minor practical issues associated with the modeling of surface deformations, for e.g., we ignored the inherent dependence of a local surface deformation on its neighbors. Furthermore, our representation to group high dimensional data points inevitably introduce the drawbacks of categorizing samples on the high-dimensional Grassmann manifold [91, 83]. Hence, to deal with such limitations with our previous algorithm [106], we propose an algorithm that jointly exploits the benefit of high-dimensional Grassmann manifold to perform reconstruction, and its equivalent lower-dimensional representation to infer suitable clusters. To accomplish this, we project each Grassmannians onto a lower-dimensional Grassmann manifold which preserves and respects the deformation of the structure w.r.t its neighbors. These Grassmann points in the lower-dimension then act as a representative for the selection of high-dimensional Grassmann samples to perform each local reconstruction. In practice, our algorithm provides a geometrically efficient way to solve dense NRSfM by switching between manifolds based on its benefit and usage. Experimental results show that the proposed algorithm is very effective in handling noise with reconstruction accuracy as good as or better than the other competing methods.

5.2 Introduction: Manifold View

Non-rigid Structure-from-Motion (NRSfM), a problem where the task is to recover the three-dimensional structure of a deforming object from a set of image feature correspondences across frames. Any solution to this problem depends on the proper modeling of *structure* $\in \mathcal{M}$ and an efficient estimation of *motion* $\in \mathbb{SE}(3)$, where $\mathcal{M}$ denotes some structure manifold and $\mathbb{SE}(3)$ denotes special Euclidean group which is a differentiable manifold [56]. Though, after Bregler *et al.* factorization framework to NRSfM [170], motion estimations are mostly relaxed to rotation estimation $\in \mathbb{SO}(3)$. Even after such relaxation, the problem still remains unsolved for any arbitrary motion. The main difficulty in NRSfM comes from the fact that both the camera and the object are moving and, along with it the object themselves are deforming, hence, it becomes difficult to distinguish camera motion from object motion using only image data. Despite such difficulties, many efficient and reliable solutions based on the priors are proposed to solve NRSfM. A reliable solution to this problem is important as it covers a wide range of applications from medical industry to the entertainment industry and many more.

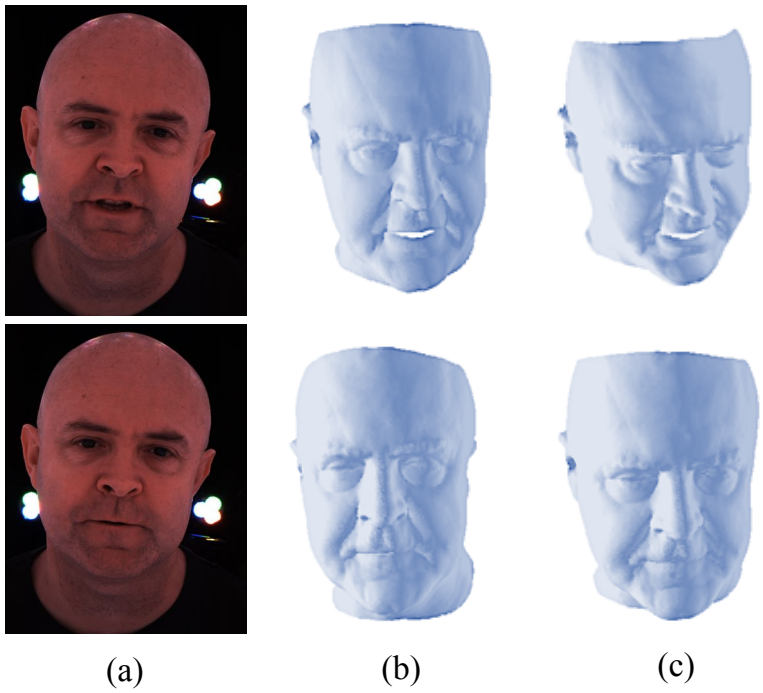


Figure 5.1: Dense 3D reconstruction of facial expression using our algorithm. The result show the 3D reconstruction of 73,765 points of a complex non-rigidly deforming surface. These results can be useful for real world applications such as 3D modeling, virtual reality etc. The example sequence is taken from Actor dataset [14].

To solve NRSfM, the algorithms proposed in the past can broadly be divided into two major classes 1) *sparse* NRSfM and 2) *dense* NRSfM. This classification is based on the number of feature points that the algorithm can efficiently process to model the deformation of the object. Although many reliable solution to this problem exists for sparse NRSfM [44, 107, 8, 166, 143, 119, 73, 81], very few work have been done towards solving the dense NRSfM reliably and efficiently [65, 41, 106, 10]. Also, the existing solutions to dense NRSfM are computationally expensive and are mostly constrained to analyze the global deformation of the non-rigid shape [65, 41]. The basis for this gradual progress in dense NRSfM is perhaps due to its dependence on per pixel reliable correspondences across frames, and the absence of a resilient structure modeling framework to capture the local non-linearities. One may argue on the efficient motion estimation, however, from image correspondences, we can only estimate relative motion and reliable algorithms with solid theory exists to perform this task well [44, 119]. Also, with the recent progress in deep learning algorithms, per pixel correspondences can be achieved with a remarkable accuracy [164], which leaves structure modeling as a potential gray area in dense NRSfM to focus.

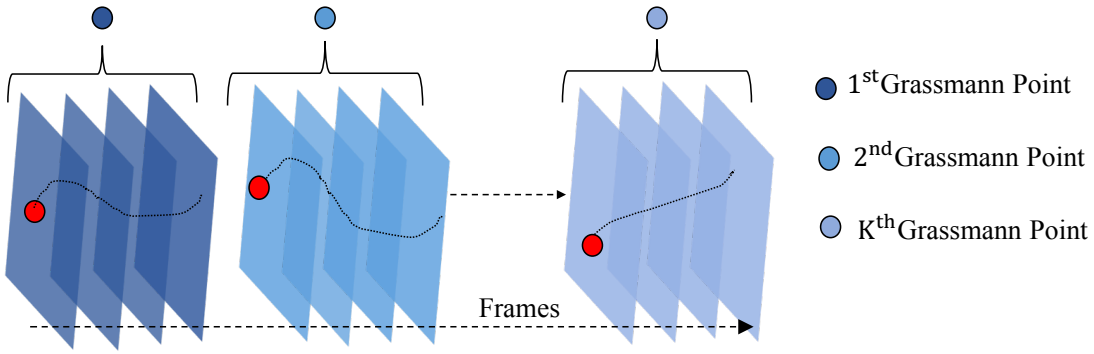


Figure 5.2: Temporal representation using Grassmannians in the shape space introduces discontinuity in the overall trajectory of the feature point. Also, to define neighboring subspace dependency graph in the time domain seems very challenging keeping in mind that the activity/expression may repeat. Red circle shows the feature point with its trajectory over frames (Black).

In the previous chapter, we exploited the Grassmann manifold to model non-rigid surfaces in dense NRSfM. The key insight in that work is; even though the overall complexity of the deforming shape is high, each local deformation may be less complex [37, 38, 39, 40]. Using this idea, we proposed a union of local linear subspace approach to solve dense NRSfM problem. Nevertheless, that work overlooked on some of the intrinsic issues associated with the modeling of non-rigidly deforming surface. *Firstly*, we represents each local linear subspace independently via a high-dimensional Grassmannian representation. Now, such representation may help reconstruct complex 3D deformation but can lead to wrong clustering, and it's very important in joint reconstruction and clustering framework to have suitable clustering of subspaces, else reconstruction may suffer. *Secondly*, the previous approach to represent local non-linear deformation completely ignored the neighboring surfaces, which may result in an inefficient representation of the Grassmannians in the trajectory space. *Thirdly*, the representation of Grassmannians in the shape space adopted in the previous work results in *irredeemable* discontinuity of the trajectories (see Fig.(5.2)). Hence, temporal representation of the set of shapes using Grassmannians seems not an extremely beneficial choice for modeling dense NRSfM on Grassmannian manifold*. *Lastly*, although the dense NRSfM algorithm proposed in last chapter works better and faster than the other methods, it depends on several manual parameters which are inadmissible for a practical application.

This chapter introduce an algorithm that overcomes the aforementioned limitations with previous formulation. The main point we are trying to make is that; reconstruction and

*Purpose behind NRSfM is not the same as activity/action recognition. See Appendix (D) for a discussion on this.

grouping of subspace on the same high dimensional Grassmann manifold seem an unreasonable choice. Even recent research in the Riemannian geometry has shown that the low-dimensional representation of the corresponding high dimensional Grassmann manifold is more favorable for grouping Grassmannians [91, 83]. So, we formulate dense NRSfM in a way that it takes advantage of both high and low dimensional representation of Grassmannians *i.e.*, perform reconstruction in the original high-dimension and cluster subspace in its lower-dimension representation.

We devise an unsupervised approach to efficiently represent the high-dimensional non-rigid surface on a lower dimensional Grassmann manifold. These low-dimensional Grassmannians are represented in such a way that it preserves the local structure of the surface deformation in accordance with its neighboring surfaces when projected. Now, these low-dimensional Grassmannians serves as a potential representative for its high-dimensional Grassmannians for suitable grouping, which subsequently help improve the reconstruction and representation of the Grassmannians on the high-dimensional Grassmann manifold, hence, the term J*umping* M*anifolds* (MoJu). Further, we drop the temporal grouping of shapes using Grassmannians to discourage the discontinuity of the trajectories (see Fig.(5.2)).

In essence, our work is inspired from the last chapter and is oriented towards settling its important limitations. Moreover, in contrast to last algorithm, we capture the notion of dependent local subspace in a union of subspace algorithm [118] via Grassmannian modeling. The algorithm we proposed is an attempt to supply a more efficient, reliable and practical solution to this problem. Our formulation gives an efficient framework for modeling dense NRSfM on the Grassmann manifold than [106]. Experimental results show that our method is as accurate as other algorithms and is numerically more efficient in handling noise. The main contributions of this work are as follows:

- An efficient framework for modeling non-rigidly deforming surface that exploits the advantage of Grassmann manifold representation of different dimensions based on its geometry.
- A formulation that encapsulates the local non-linearity of the deforming surface w.r.t its neighbors to enable the proper inference and representation of local linear subspaces.
- An iterative solution to the proposed cost function based on ADMM [19], which is simple to implement and provide results as good as the best available methods. Additionally, it helps improve the 3D reconstruction substantially, in the case of noisy trajectories.

Next, we will briefly discuss some previous work that solves dense NRSfM. Although in the last chapter we mentioned some of it, we reference it concisely for easy follow up.

5.3 Relevant Previous Work

Earlier attempts to solve dense NRSfM used piecewise reconstruction of the shape parts which were further processed via a stitching step to get a global 3D shape [35, 150]. To our knowledge, Garg *et al.* variational approach [67] was the first to propose and demonstrate per pixel dense NRSfM algorithm without any 3D template prior. This method introduced a discrete total variational constraint with trace norm constraint on the global shape, which resulted in a biconvex optimization problem. Despite the algorithm outstanding results, it's computationally very expensive and needs a GPU to provide the solution.

In contrast, Dai *et al.* extended his simple prior free approach to solve dense NRSfM problem [44, 41]. The algorithm proposed a spatial-temporal formulation to solve the problem. The author revisits the temporal smoothness term from [44] and integrate it with a spatial smoothness term using the Laplacian of the non-rigid shape. The resultant optimization leads to a series of least squares to be minimized which makes it extremely slow to process. In the previous chapter, we modeled this problem on the Grassmann manifold [106]. The work extended the spatiotemporal multi-body framework to solve dense NRSfM [107]. The algorithm demonstrated that such an approach is more efficient, faster and accurate than all the other recent approaches to solve dense NRSfM task [67, 41, 10].

Consecutive frame-based approach has recently shown some promising results to solve dense 3D reconstruction of a general dynamic scene including non-rigid object [111, 149]. Nevertheless, motion segmentation, triangulation, as rigid as possible constraint and scale consistency quite often breaks down for the deforming object over frames. Therefore, dense NRSfM becomes extremely challenging for such algorithms. Not long ago, Gallardo *et al.* combined shading, motion and generic physical deformation to model dense NRSfM [61].

5.4 Preliminaries

In this chapter, $\|.\|_F$, $\|.\|_*$ denotes the Frobenius norm and nuclear norm respectively. $\|.\|_{\mathcal{G}}$ represent the notion of norm on the Grassmann manifold. Single angle bracket $< .,. >$ denotes the Euclidean inner product. Despite we discussed some of the manifold preliminaries in the last chapter, for ease of understanding and completeness, in this section, we briefly review few important definitions related to the Grassmann manifold. Firstly, a manifold is a topological space that is locally similar to the Euclidean space —this is a loose definition and may not be completely satisfactory to a mathematician but its helpful for building intuition. Out of several manifolds, the Grassmann manifold is a topologically rich non-linear manifold, each point of which represent the set of all right invariant subspace of the Euclidean space [48, 2, 106].

Definition 3. *The Grassmann manifold, denoted by $\mathcal{G}(p,d)$, consists of all the linear 'p' dimensional subspace embedded in a 'd' dimensional Euclidean space $\mathbb{R}^d$ such that $0 \leq p \leq d$ [Absil et al., 2009] [2].*

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

Definition 4. *Grassmann manifold can be embedded into the space of symmetric matrices via mapping $\Pi : \mathcal{G}(p,d) \mapsto Sym(d), \Pi(\Phi) = \Phi\Phi^T$, where Φ is a Grassmann point [80, 82]. Given two Grassmann points Φ_1 and Φ_2, then the distance between them can be measured using the projection metric $d_g^p(\Phi_1,\Phi_2) = 0.5\|\Pi(\Phi_1) - \Pi(\Phi_2)\|_F^2$ [80].*

In the past, these two properties of Grassmann manifold has been used in many computer vision applications [80, 28, 106]. Second definition is very important as it allows to measure the distance on the Grassmann manifold, hence, $(\mathcal{G}, d_g)$ forms a metric space. We used these properties in the construction of our formulation. For comprehensive details on this topic readers may refer to [80].

5.5 Problem Formulation

Let 'P' be the total number of feature points tracked across 'F' frames. Concatenating these 2D coordinates of each feature points for all frames across the columns of a matrix gives 'W' $\in \mathbb{R}^{2F\times P}$ matrix. This matrix is popularly known as *measurement matrix* [170]. Our goal is, given the image measurement matrix, estimate the camera motion and 3D coordinates of every 2D feature points across all frames.

We start our formulation with the classical representation to NRSfM *i.e.*$W = RS$, where, $R \in \mathbb{R}^{2F\times 3F}$ is a block diagonal rotation matrix with each block as a 2×3 orthographic rotation matrix, and $\in \mathbb{R}^{3F\times P}$ as the 3D structure matrix. With such a representation, the entire problem simplifies to the estimation of correct rotation matrix 'R' and structure matrix 'S' such that the above relation holds. Following the assumption of the previous work [106], we estimate the rotation using Intersection method [44]. As a result, the task reduces to composing of an efficient algorithm that correctly models the surface deformations and provide better reconstruction results. Recent algorithms in NRSfM have demonstrated that clustering benefits reconstruction and vice-versa, however, the existing framework to employ this idea is not scalable to millions of points. To establish this idea for dense NRSfM, Kumar *et al.*[106] used LRR on Grassmannian manifold. Using the similar notions, we model dense deforming surface using Grassmannian representation to provide more reliable and accurate solution.

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

5.5.1 Grassmannian representation

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

$$\varphi_i = \Delta^T \Phi_i \tag{5.1}$$

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

$$\Theta_i U_i = \text{qr}(\varphi_i) = \Delta^T \Phi_i \tag{5.2}$$

Here, 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[†]. Using Eq.(5.2), we represent the equivalence of Φ_i in low dimension as

$$\begin{aligned} \Theta_i &= \Delta^T (\Phi_i U_i^{-1}) \\ \Theta_i &= \Delta^T \Omega_i \end{aligned} \tag{5.3}$$

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.(5.3), Fig.(5.4)), we further constrain our representation as:

$$E(\Delta) = \underset{\Delta}{\text{minimize}} \sum_{(i,j)}^{K} w_{ij} \frac{1}{2} \|\Pi(\Theta_i) - \Pi(\Theta_j)\|_2^F \tag{5.4}$$

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

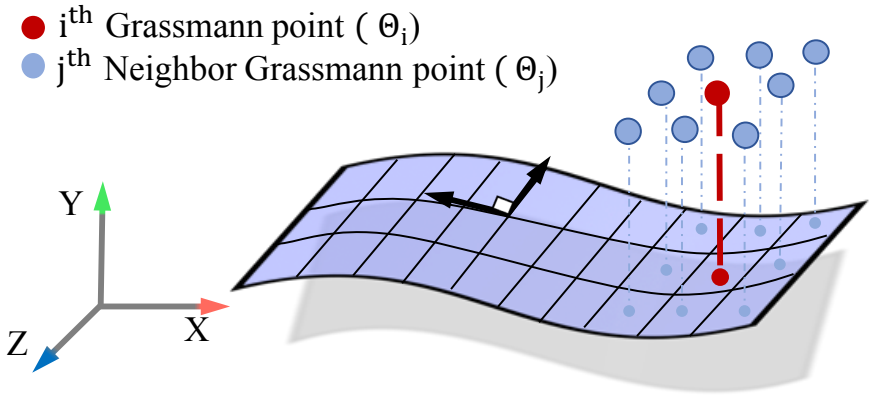


Figure 5.3: In contrast to [106], our modeling of surface using Grassmannians considers the similarity between the neighboring Grassmannians while representing it in the lower dimension. Based on the assumption that spatially neighboring surface tend to span similar subspace, defining neighboring subspace dependency graph is easy and most of the real-world examples follows such assumption. However, building such graph in shape space can be tricky.

The parameter 'w_{ij}' accommodate the similarity knowledge between the two Grassmannians. Using the Definition(4) and Eq.(5.3), we further simplify Eq.(5.4) as

$$\begin{aligned} E(\Delta) &\equiv \underset{\Delta}{\text{minimize}} \sum_{(i,j)} w_{ij} \frac{1}{2} \|\Delta^T \Omega_i \Omega_i^T \Delta - \Delta^T \Omega_j \Omega_j^T \Delta\|_F^2 \\ E(\Delta) &\equiv \underset{\Delta}{\text{minimize}} \sum_{(i,j)}^{K} w_{ij} \frac{1}{2} \|\Delta^T (\Omega_i \Omega_i^T - \Omega_j \Omega_j^T) \Delta\|_F^2 \\ E(\Delta) &\equiv \underset{\Delta}{\text{minimize}} \sum_{(i,j)}^{K} w_{ij} \frac{1}{2} \|\Delta^T (\Lambda_{ij}) \Delta\|_F^2 \end{aligned} \tag{5.5}$$

where, $\Lambda_{ij} \in Sym(d)$. The parameter 'w_{ij}' (similarity graph) is set as $exp(-d_g^2(\Phi_i, \Phi_j))$ with d_g as the projection metric (see Definition (4)). Eq.(5.5) is an unconstrained optimization problem and its solution may provide a trivial solution. To estimate the useful solution, we further constrain the problem. Using i^{th} Grassmann point 'Ω_i' and its neighbors, expand Eq.(5.5). By performing some simple algebraic manipulation, Eq.(5.5) reduces to

$$\text{trace}\Big(\Delta^T \big(\sum_{i=1}^{K} \lambda_{ii} \Omega_i \Omega_i^T\big) \Delta\Big) \tag{5.6}$$

where, $\lambda_{ii} = \sum_{j=1}^{K} w_{ij}$. Constraining the value of Eq.(5.6) to 1 provides the overall optimization for an efficient representation of the local non-rigid surface on the Grassmann manifold.

$$
\begin{aligned}
E(\Delta) \equiv \underset{\Delta}{\text{minimize}} & \sum_{(i,j)}^{K} w_{ij} \frac{1}{2} \|\Delta^T(\Lambda_{ij})\Delta\|_F^2 \\
& \text{subject to:} \\
& \text{trace}\Big(\Delta^T\big(\sum_{i=1}^{K} \lambda_{ii}\Omega_i\Omega_i^T\big)\Delta\Big) = 1
\end{aligned}
\tag{5.7}
$$

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

5.5.2 Dense NRSfM formulation

To solve the dense non-rigid structure from motion with the representation formulated in the previous sub-section §5.5.1, we propose to jointly optimize the objective function over the 3D structure and its local group representation. In order to build the overall objective function, we introduce each constraint equation one by one for clear understanding of our overall cost function.

$$
\mathrm{E}_p(S) = \underset{S}{\text{minimize}} \ \frac{1}{2}\|W - RS\|_F^2 \tag{5.8}
$$

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

$$
\mathrm{E}_s(S^\sharp) = \underset{S^\sharp}{\text{minimize}} \ \|S^\sharp\|_* \tag{5.9}
$$

The *second* term caters the global assumption about the non-rigid object; that is the overall shape matrix is low-rank. To establish this assumption, we perform rank minimization of the shape matrix. Although the rank minimization of a matrix is NP-hard, it's relaxed to nuclear norm minimization to find an approximate solution. This term mainly penalizes the total number of independent shape required to represent the shape. The choice of minimizing $S^\sharp \in \mathbb{R}^{3P\times F}$ instead to $S \in \mathbb{R}^{3F\times P}$ is inspired from Dai *et al.*'s work [44]. Since the dense deforming shape is composed of several local linear low-dimensional subspace, the global constraint (Eq.(5.9)) may not reflect their local dependency. Therefore, in order to introduce the local subspace constraint on the shape, we use the notion of self-expressiveness

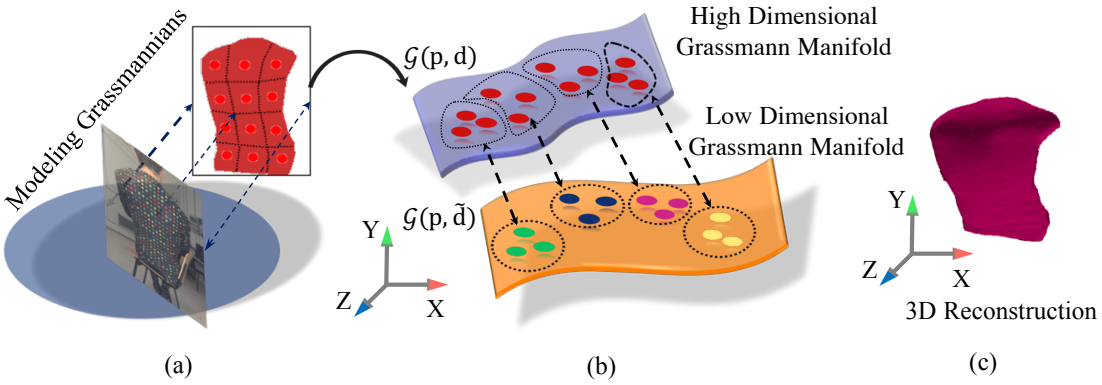


Figure 5.4: Conceptual illustration of our modeling (a) Modeling of 3D trajectories to Grassmann points (b) The two grassmann manifold and mapping of the points between them to infer better cluster index that leads to better reconstruction (c) The 3D reconstruction of the non-rigid deforming object.

on the non-linear Grassmann manifold space.

$$\begin{aligned} &\underset{E,C,S^\sharp}{\text{minimize}} \ \|E\|_{\mathcal{G}}^2 + \beta_2 \|S^\sharp\|_* + \beta_3 \|C\|_* \\ &\text{subject to: } S^\sharp = f(S), S = SC + E \end{aligned} \tag{5.10}$$

Here, we define $f : S \in \mathbb{R}^{3F \times P} \mapsto S^\sharp \in \mathbb{R}^{3P \times F}$ and $C \in \mathbb{R}^{P \times P}$ as the coefficient matrix. We know from the literature that the Grassmann manifold is isometrically equivalent to the symmetric idempotent matrix [33]. So, we embed the Grassmann manifold into symmetric matrix manifold to define the self-expressiveness. Let $\tilde{\xi} = \{\Theta_1, \Theta_2, ..., \Theta_K\}$ be the set of Grassmannians on a low-dimensional Grassmann manifold. The elements of $\tilde{\xi}$ are the projection of high-dimensional Grassmannian representation of the columns of '' matrix. Let $\chi = \{(\Theta_1\Theta_1^T), (\Theta_2\Theta_2^T), ..., (\Theta_K\Theta_K^T)\}$ be its embedding onto symmetric matrix manifold. Using such embedding techniques we re-write Eq.(5.10) as

$$\begin{aligned} &\underset{E,\tilde{C},\sharp}{\text{minimize}} \ \|E\|_F^2 + \beta_2 \|S^\sharp\|_* + \beta_3 \|\tilde{C}\|_* \\ &\text{subject to:} \\ &S^\sharp = f(S), \chi = \chi\tilde{C} + E \end{aligned} \tag{5.11}$$

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

P, which makes such representation scalable.

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

$$\begin{aligned}&\Big\{\big(\xi, f_g(\mathrm{P}, S)\big) : \text{order } \{S_i\}_{i=1}^{K} \text{ columns of } S \text{ of using P},\\ &\xi := \{\Phi_i\}_{i=1}^{K} \text{where, } [\,\Phi_i, \Sigma_i, \xi_{\text{vi}}] = \text{svds}(S_i, p)\Big\}\end{aligned} \tag{5.12}$$

$$\begin{aligned}&\Big\{\big(\tilde{\xi}, f_h(\Delta, \xi)\big) : \tilde{\xi} = \{\Theta_i\}_{i=1}^{K}, \Theta_i = \Delta^T(\Phi_i U_i^{-1}),\\ &\text{where, } \Delta = \text{solution to the minimization of Eq.(5.7)}\Big\}\end{aligned} \tag{5.13}$$

$$\Big\{(\mathrm{P}, f_p(\tilde{\xi}, \tilde{C}, \mathrm{P}_o) : \mathrm{P} = \text{spectral\_clustering}(\tilde{\xi}, \tilde{C}, \mathrm{P}_o)\Big\} \tag{5.14}$$

$$\Big\{\big(S, f_s(\xi, \Sigma, \xi_v)\big) : {}_i = [\xi_i\, \Sigma_i\, \xi_{vi}], \text{where } \Sigma_i \in \mathbb{R}^{p\times p}\Big\} \tag{5.15}$$

Intuitively, the first function (f_g) uses the ordering vector $\mathrm{P} \in \mathbb{R}^{1\times P}$ to refine the grouping of the trajectories for suitable Grassmannian representation. The second function (f_h) projects the Grassmannians to a lower dimension in accordance with the neighbors using Eq.(5.7). The third function (f_p) uses the projected Grassmannians to assign proper labeling to the Grassmann points and update the given ordering vector P using spectral clustering. The fourth function (f_s) uses the group of trajectories to reconstruct back the set of local surface. Σ, ξ_v are the singular values and right singular vector matrices in the high-dimension.

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

$$\begin{aligned}&\underset{E,\tilde{C},S,S^\sharp}{\text{minimize}} \frac{1}{2}\|W - RS\|_F^2 + \beta_1\|E\|_F^2 + \beta_2\|S^\sharp\|_* + \beta_3\|\tilde{C}\|_*\\ &\text{subject to:}\\ &S^\sharp = f(S), \chi = \chi\tilde{C} + E,\\ &\xi = f_g(\mathrm{P}, S), \tilde{\xi} = f_h(\Delta, \xi),\\ &S = f_s(\xi, \Sigma, \xi_v), \mathrm{P} = f_p(\tilde{\xi}, \tilde{C}, \mathrm{P}_o)\end{aligned} \tag{5.16}$$

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

5.6 Solution

The optimization proposed in Eq.(5.16) is a coupled optimization problem. Several methods of Bi-level optimization can be used to solve such minimization problem [13, 74]. Nevertheless, we propose ADMM [19] 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.(5.7), and therefore, we cannot directly embed the constraint to the main objective function. Instead, we only introduce two Lagrange multiplier L_1, 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 $\tilde{C}$ from $\mathcal{X}$, we introduce auxiliary variable $\tilde{C} = Z$. We apply these operations to our optimization problem to get the following Augmented Lagrangian form:

$$\begin{aligned}\underset{Z,\tilde{C},S,S^\sharp}{\text{minimize}} & \frac{1}{2}\|W - RS\|_F^2 + \beta_1\|\mathcal{X} - \mathcal{X}\tilde{C}\|_F^2 + \beta_2\|S^\sharp\|_* + \\ & \frac{\varrho}{2}\|S^\sharp - f(S)\|_F^2 + < L_1, S^\sharp - f(S) > + \beta_3\|Z\|_* + \\ & \frac{\varrho}{2}\|\tilde{C} - Z\|_F^2 + < L_2, \tilde{C} - Z > \end{aligned} \tag{5.17}$$

subject to:

$$\xi = f_g(P, S), \tilde{\xi} = f_h(\Delta, \xi),$$
$$S = f_s(\xi, \Sigma, \xi), P = f_p(\tilde{\xi}, \tilde{C}, P_o)$$

Note that $\tilde{C}$ 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 $\tilde{C}$, $f_g(.)$ provides new Grassmann sample corresponding to each group. The definition of $f_h(.)$ and $f_s(.)$ is provided in Eq.(5.7) and Eq.(5.14) respectively. More generally, the solution to the optimization in Eq.(5.7) is obtained by solving it as a generalized eigenvalue problem. To keep the order of columns of 'S' matrix consistent with 'W' matrix $f_p(.)$ provides the ordering index. We provide the implementation details of our method with suitable MATLAB commands in the Algorithm (3). For details on the derivation to each sub-problem, kindly refer to the Appendix (D).

5.7 Initialization and Evaluation

We performed experiments and evaluation on the available standard benchmark datasets [65, 183, 14]. To keep our evaluations consistent with the previous methods, we compute the mean

Algorithm 3 Dense Non-rigid Structure from Motion (MoJu)

Require: W, R, $\{\beta_i\}_{i=1}^{3}$, ϱ=e^{-2}, ϱ_m=e^{8}, ε=e^{-10}, c =1.1, K;
Initialize: S=pinv$(R)W$, $S^{\sharp}$=$f(S)$, Z=0, $\{\mathrm{L}_i\}_{i=1}^{2}$=0, $\tilde{d}$;
Δ = [$\mathrm{I}_{\tilde{d}\times\tilde{d}}$; random values], p %top singular values
P_o = kmeans++(S, K), iter = 1, $\mathrm{P}_{\text{store}}$(iter, :) = P_o,
P = P_o
while not converged do
1. S := mldivide$\left(R^TR + \varrho I, \varrho(f^{-1}(S^{\sharp}) + f^{-1}(\mathrm{L}_1)/\varrho) + R^TW\right)$;
2. $\xi := f_g(\mathrm{P}, S)$; see Eq.(5.12)
3. W := arrange_column(P, W)
4. Update the similarity matrix 'w_{ij}' using ξ. §5.5.1
5. $\tilde{\xi} := f_h(\xi, \Delta)$; s.t, $\Delta \equiv \underset{\Delta}{\text{minimize}}\ E(\Delta)$; see Eq.(5.13)
6. $\Gamma_{ij} = \mathrm{Tr}[(\Theta_j^T\Theta_i)((\Theta_i^T\Theta_j)]$; $\Gamma = (\Gamma_{ij})_{ij=1}^{K}$; $L = \mathrm{chol}(\Gamma)$
7. $\tilde{C} := \left(2\beta_1 LL^T + \varrho(Z - \mathrm{L}_2/\varrho)\right)\left(2\beta_1 LL^T + \varrho I\right)^{-1}$;
8. P := $f_p(\tilde{\xi}, \tilde{C}, \mathrm{P})$;
9. $S := f_s(\xi, \Sigma, \xi_v)$; see Eq.(5.14)
10. $S^{\sharp} := U_s\mathcal{S}_{-2}(\Sigma_s)V_s$; s.t, $[U_s, \Sigma_s, V_s] := \mathrm{svd}(f(S) - \mathrm{L}_1/\varrho)$
11. $Z := U_z\mathcal{S}_{-3}(\Sigma_z)V_z$; s.t, $[U_z, \Sigma_z, V_z] := \mathrm{svd}(\tilde{C} + \mathrm{L}_2/\varrho)$;
12. $\mathrm{L}_1 := \mathrm{L}_1 + \varrho(S^{\sharp} - f(S))$; $\mathrm{L}_2 := \mathrm{L}_2 + \varrho(\tilde{C} - Z)$
13. iter := iter + 1; $\mathrm{P}_{\text{store}}$(iter, :) := P;
14. $\varrho := \min(\varrho_m, c\varrho)$;
15. gap := $\max\{\|S^{\sharp} - f(S)\|_{\infty}, \|\tilde{C} - Z\|_{\infty}\}$;
(gap $< \varepsilon$) $\vee$ ($\varrho > \varrho_m$) $\rightarrow$ break; %convergence check
end while
return S;
e_{3D} = Estimate_error (S, S_{GT}, $\mathrm{P}_{\text{store}}$); %use Eq.(5.18)

normalized 3D reconstruction error of the estimated shape 'S_{est}' after convergence as

$$e_{3D} = \frac{1}{F}\sum_{i=1}^{F}\frac{\|S^i_{\text{est}} - S^i_{GT}\|_F}{\|S^i_{GT}\|_F} \tag{5.18}$$

here 'S_{GT}' denotes the ground-truth 3D shape matrix.

Initialization: We used Intersection method [44] to estimate the rotation matrix and initialize S = pinv$(R)W$. The initial grouping of the trajectories or columns of S is done

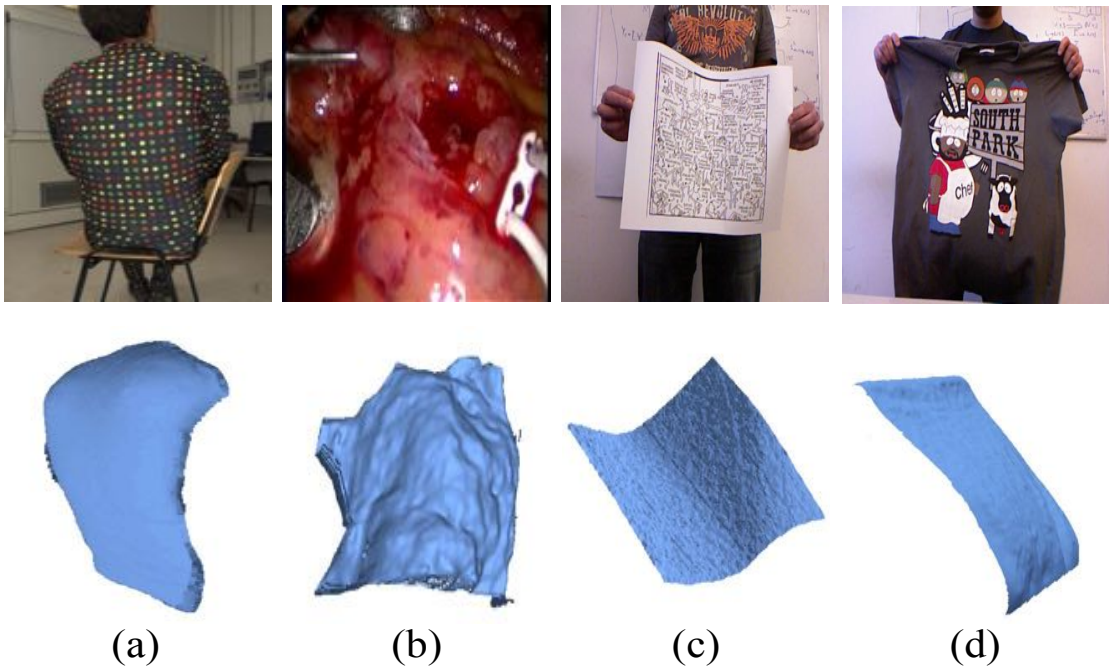


Figure 5.5: From left: 3D reconstruction results on Back [65], Heart [65], Paper[183] and T-shirt [183] data sequence respectively.

using k-means++ algorithm [11]. These groups are then used to initialize P_o, P and the Grassmann points $\{\Phi_i\}_{i=1}^{K} \in \xi$ via subset of singular vectors. To represent the Grassmannians in the lower-dimension, we solve Eq.(5.7) to initialize $\tilde{\xi}$ and store corresponding singular values. The similarity matrix or graph in Eq.(5.7) is build using the distance measure between the Grassmannians in the embedding space §5.5.1.

1. Results on synthetic Face dataset: The synthetic face dataset is composed of four distinct sequence [65] with 28,880 feature points tracked over multiple frames. Each sequence captures the human facial expression with a different range of deformations and camera motion. Sequence 1 and Sequence 2 are 10 frame long video with rotation in the range $\pm 30^\circ$ and $\pm 90^\circ$ respectively. Sequence 3 and Sequence 4 are 99 frame long video that contains high frequencies and low frequencies rotation respectively which captures real human facial deformations. Table (5.1) shows the statistical results obtained on these sequences using our algorithm. For qualitative results on these sequences kindly refer to Appendix (D).

Dataset	MP	PTA	CSF1	CSF2	DV	DS	SMSR	SDG	Ours
Face1	0.2572	0.1559	0.5325	0.4677	0.0531	0.0636	0.1893	0.0443	0.0404
Face2	0.0644	0.1503	0.9266	0.7909	0.0457	0.0569	0.2133	0.0381	0.0392
Face3	0.0682	0.1252	0.5274	0.5474	0.0346	0.0374	0.1345	0.0294	0.0280
Face4	0.0762	0.1348	0.5392	0.5292	0.0379	0.0428	0.0984	0.0309	0.0327
Actor1	0.5226	0.0418	0.3711	0.3708	-	0.0891	0.0352	0.0340	0.0274
Actor2	0.2737	0.0532	0.2275	0.2279	-	0.0822	0.0334	0.0342	0.0289
Paper	0.0827	0.0918	0.0842	0.0801	-	0.0612	-	0.0394	0.0338
T-shirt	0.0741	0.0712	0.0644	0.0628	-	0.0636	-	0.0362	0.0386

Table 5.1: Statistical comparison of our method with competing approaches namely MP [143], PTA [7], CSF1 [71], CSF2[73], DV [65], DS [41], SMSR [10] and SDG[106]. Quantitative evaluations for SMSR [10] and DV [65] are not performed by us due to the unavailability of their code, and therefore, we tabulated their reconstruction error from their published work. Codes for DS [41] and SDG [106] are obtained through personal communication.

2. Results on Paper and T-shirt dataset: Varol *et al.*introduced 'kinect_paper' and 'kinect_tshirt' datasets to test the performance of NRSfM algorithm under real conditions [183]. This dataset provides sparse SIFT [127] 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.*[63] algorithm to estimate the measurement matrix. To keep the numerical comparison consistent with the previous work in dense NRSfM [106], we used the same coordinate range for tracking the features. Numerically, its $x_w = (253, 253, 508, 508)$, $y_w = (132, 363, 363, 132)$ rectangular window across 193 frames for kinect_paper sequence. For kinect_tshirt sequence, we considered rectangular window of $x_w = (203, 203, 468, 468)$, $y_w = (112, 403, 403, 112)$ across 313 frames, same as used in Kumar *et al.*work [106]. Fig.(5.5) shows the reconstruction results on these sequence with comparative results provided in Table (5.1).

3. Results on Actor dataset: Beeler *et al.* [14] 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 [10]. Fig.(5.6) show the dense detailed reconstruction that is achieved using our algorithm. Table (5.1) clearly indicates the benefit of our approach to reconstruct such complex deformations.

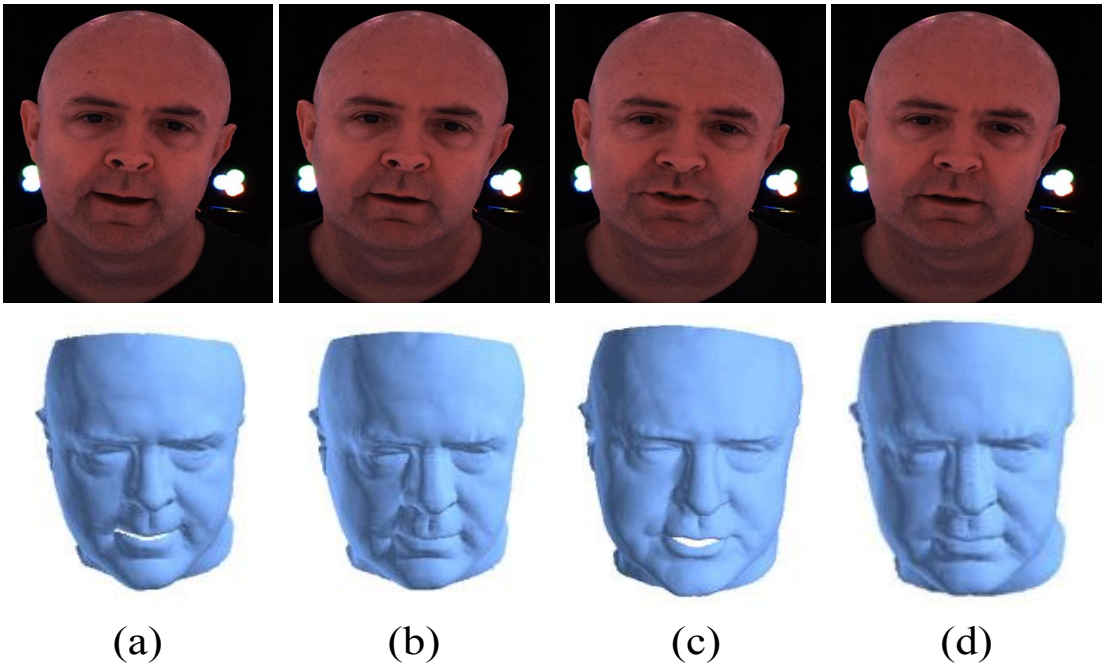


Figure 5.6: 3D reconstruction results on the Actor sequence [14].

4. Results on Face, Heart, Back dataset: To evaluate the variational approach to dense NRSfM [65] Garg *et al.* introduced these datasets. Its sequences are composed of monocular video's captured in a natural environment with varying lighting condition and large displacements. It consists of three different videos with 120, 150 and 80 frames for face sequence, back sequence and heart sequence respectively. Additionally, it provides dense 2D feature track for the same with 28332, 20561, and 68295 features track over the frames for face, back and heart sequence. No ground-truth 3D is available with this dataset for evaluation. Fig.(5.5) show reconstruction results on back and heart sequence. Fig.(5.7) and Fig.(5.8) show the 3D reconstruction results of our algorithm on real face and synthetic face sequence.

5.7.1 Algorithmic Analysis

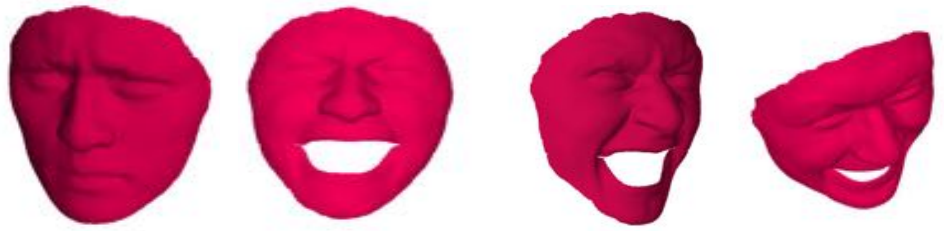
We performed some more experiments to understand the behavior of our algorithm under different input parameters and evaluation setups. In practice these experiment help analyze the practical applicability of our algorithm.

Figure 5.7: 3D reconstruction results on real face sequence [65]

1. Performance over noisy trajectories: We utilized the standard experimental procedure to analyze the behavior of our algorithm under different noise levels. Similar to the previous work [106], we added the Gaussian noise to the input trajectories. The standard deviation of the noise are adjusted as $\sigma_g = \lambda_g \max\{|W|\}$ with λ_g varying from 0.01 to 0.055. Fig.(5.9a) show the quantitative comparison of our approach with recent algorithms namely DS [41] and SDG [106]. The graph is plotted by taking the average reconstruction error of all the four synthetic face dataset [65]. The procured statistics indicate that our algorithm is more resilient to noise than other competing methods.

2. Performance with change in the number of singular values: The selection of 'p' in $\mathcal{G}(,)$ *i.e.* the number of top singular vectors for Grassmannian representation and its corresponding singular values to perform reconstruction can directly affect the performance of our algorithm. However, it has been observed over several experiments that we need very few singular value and singular vectors to recover dense detailed 3D reconstruction of the deforming object. Fig.(5.9b) show the variation in average 3D reconstruction with the values of 'p' for synthetic face dataset [65].

3. Dependence of the algorithm on variable $\tilde{d}$: While reducing the dimension for grouping the grassmann points, one of the critical aspect is to determine the dimension to



(a) Reconstructed Shape

(b) Ground-Truth Shape

Figure 5.8: 3D reconstruction results on synthetic face sequence [65].

which we should project for better results. We used well-known procedure of cumulative energy of eigen vectors to get the value of $\tilde{d}$. Mathematically, let Ω be the set that stack all the Grassmannians and σ_i be the i^{th} singular value of $\Omega\Omega^T$, then

$$\tilde{d} = \underset{d_{\text{opt}}}{\text{argmin}} \frac{\sum_{i=1}^{d_{\text{opt}}} \sigma_i}{\sum_{i=1}^{d} \sigma_i} \geq \tau \tag{5.19}$$

where τ can vary from 0 to 1 and d_{opt} (optimal dimension) is a positive integer. We put $\tau = 0.97$ for all our experiment. Fig.(5.10) show the variations in the reconstruction error with the value of τ. It is observed that for different dataset the value of suitable $\tilde{d}$ 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 Grassmannians.

4. Processing Time and Convergence: Our algorithm execution time is almost at par or a bit slower than SDG [106]. Fig.(5.9c) show the processing time taken by our method on different datasets. Fig.(5.9d) show a typical convergence curve of our algorithm. Ideally, it takes 120-150 iteration to provide an optimal solution to the problem.

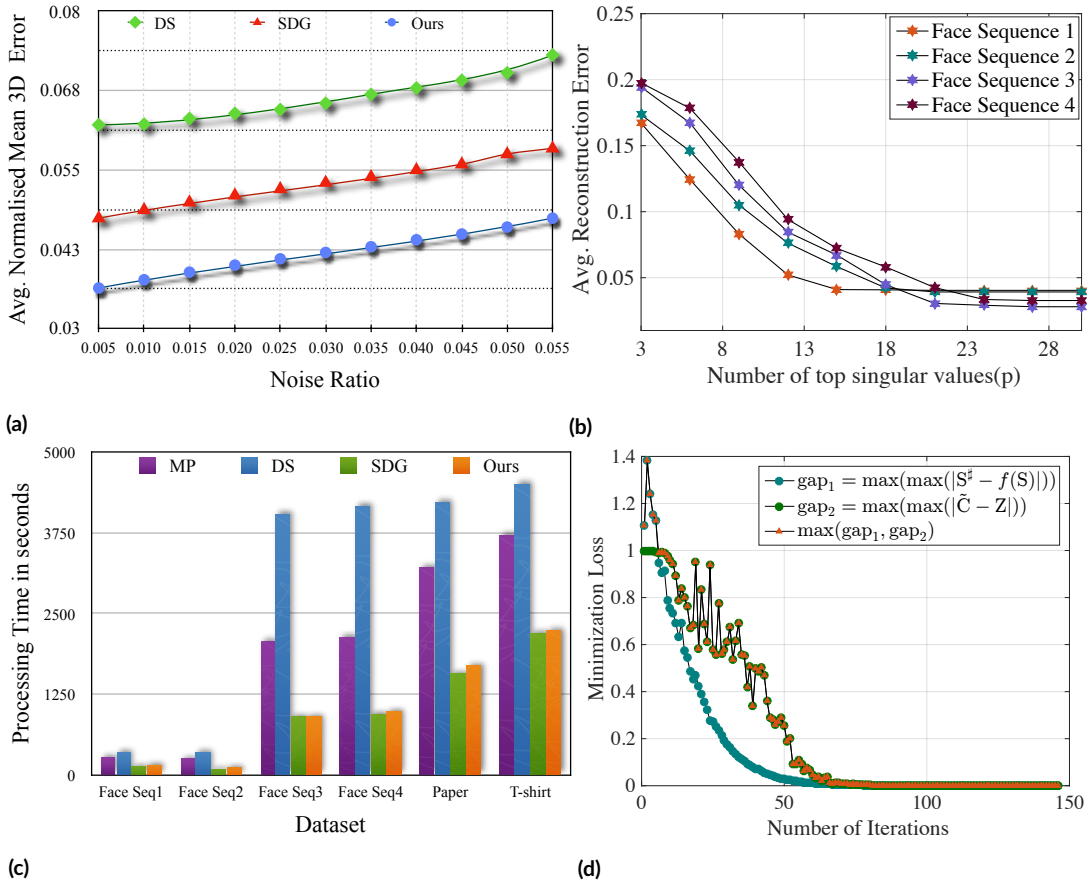


Figure 5.9: (a) Variation in the average 3D reconstruction error with change in the noise ratio's for face dataset[65]. (b) Fluctuation in the 3D reconstruction accuracy with change number of top singular values and corresponding singular vectors used by our algorithm for face sequence[65]. (c) Processing time againt other competing algorithm's on Intel Core i7-4790 CPU @3.60GHz x 8 Desktop with MATLAB 2017b, our method show comparable execution timing to SDG[106]. (d) A typical ADMM optimization convergence curve of our algorithm.

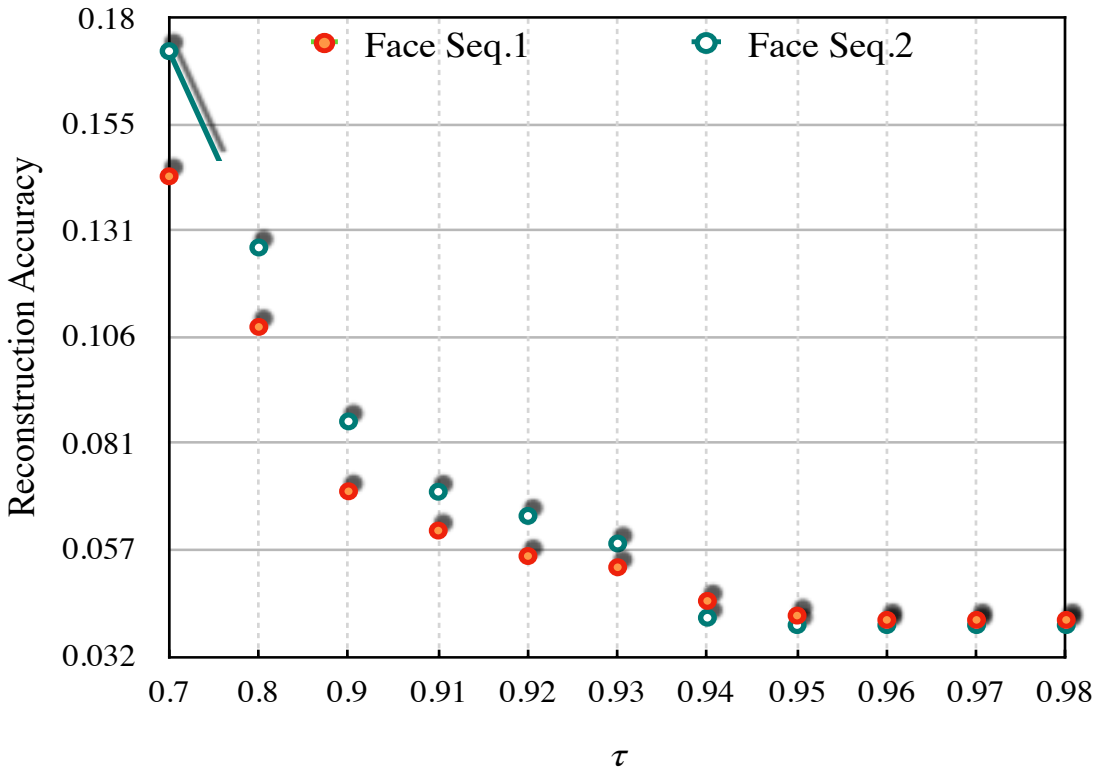


Figure 5.10: Accuracy variation with respect to τ.

5.8 Closing Remarks

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

6

Dense monocular 3D reconstruction of a complex dynamic scene.

Contents

In this chapter we will deviate from factorization approach to solve dense NRSFM. We introduce a dense 3D reconstruction algorithm which is more applicable to real world scenes

and is free from orthographic camera assumption. The algorithm can supply dense 3D reconstruction of both the background and foreground irrespective of the rigidity type.

6.1 Introduction

To set the stage for this topic, we reiterate that "The task of reconstructing 3D scene geometry from images –popularly known as structure-from-motion (SfM) is a fundamental problem in computer vision". An initial introduction and working solution to this problem can be found as early as 1970's and 1980's [178] [78] [125], which were further discussed comprehensively in Blake *et al.* seminal work [18]. While this field of study was largely dominated by sparse feature based reconstruction of rigid [88] [87] [85] [168] [170] and non-rigid objects [22] [44] [119] [109] [107], in recent years with the surge in computational resources dense 3D reconstruction of a complex dynamic scene have been introduced and successfully demonstrated [134] [149].

A dense solution to this inverse problem is required due to its increasing demands in the real-world application —from animation and entertainment industry to robotics industry (VSLAM). In particular with the proliferation of monocular camera in almost all modern mobile devices has elevated the demand for sophisticated dense reconstruction algorithm. When a 3D scene is rigid, the reconstruction can be easily done by conventional rigid-SfM techniques [85]. However, real-world scenes are more complex containing not only rigid motions but also non-rigid deformations as well as their combination. For example, a typical outdoor traffic scene consists of both multiple rigid motions of vehicles, and non-rigid motions of pedestrians etc. Therefore, it is highly desirable to develop a 3D reconstruction framework that can handle generic (complex and dynamic) scenes.

As stated earlier, when only camera is moving and the scene is static under such situation a dense 3D reconstruction can be faithfully recovered using well known geometry approach [85], upto an unknown global scale. Now, imagine a situation when there are multiple rigidly moving objects in the same scene observed by a moving camera. Although each of the individual rigid objects can be reconstructed up to an arbitrary scale (and assuming motion segmentation is done), the reconstruction of the shape of the whole dynamic scene is generally impossible, simply because the relative scales among all the moving shapes cannot be determined in a globally consistent way in general. Furthermore, since all the estimated motions are relative to each other, one cannot distinguish camera motion from object motions. Therefore, prior information about the objects or the scene and their connection in the real-world are used to fix the placement of these objects in the environment. This is precisely the pipeline adopted by Ranftl *et al.* in his recent work [149].

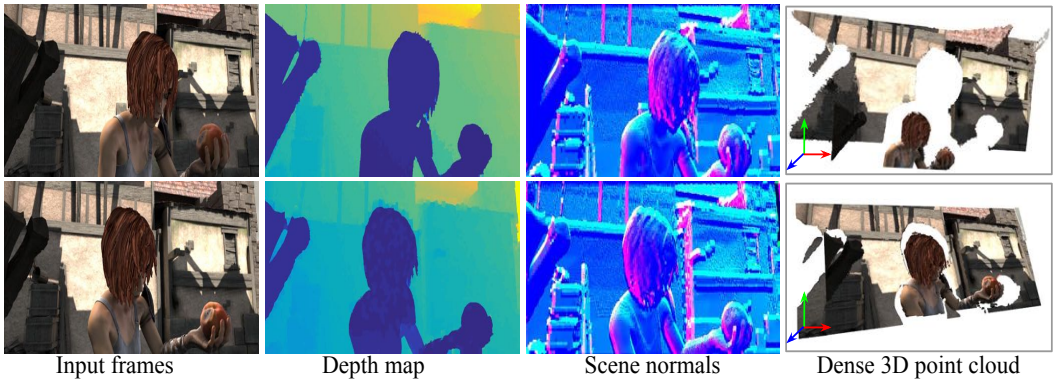


Figure 6.1: Dense 3D reconstruction of a complex dynamic scene from two frames using our method. Here, both the subject and the camera were moving dynamically. (MPI Sintel [24] *alley_1 frame 10 and 20*).

In this chapter, we present an approach to solve this problem which neither perform any object level motion segmentation nor assumes any prior knowledge about the scene rigidity type and still able to recover scale consistent dense reconstruction of a complex dynamic scene. Our formulation instinctively encapsulates the solution to inherent scale ambiguity in perspective structure from motion which is a very challenging problem to solve in general. However, we show that by using two prior assumptions –about the 3D scene and about the deformation, we can effectively pin down the unknown relative scales, and obtain a globally consistent dense 3D reconstruction of a dynamic scene from its two perspective views.

6.2 Motivation and Contribution

The formulation proposed in this work is motivated by the following endeavor in dense structure from motion of a dynamic scene.

1. Object level motion segmentation

To solve dense reconstruction of a complex dynamic scene from perspective images, a straightforward idea is:

1. Implement object level motion segmentation to infer distinct motion models of multiple rigidly moving object.
2. Execute existing rigid reconstruction algorithm [88] to retrieve per object reconstruction.

3. Use the prior information about the object and the environment to procure scale consistent reconstruction upto unknown global scale.

The main concern with such a framework is: In a general dynamic setting, the task of densely segmenting rigidly moving objects or part is not trivial. Consequently, inferring motion models for deforming shapes becomes very challenging. Furthermore, the success of object-level segmentation build upon the assumption of multiple rigid motions, fails to describe more general scenarios such as "when the objects themselves are deforming". Subsequently, reconstruction dependent on motion segmentation of objects suffers. This motivate us to develop an algorithm that is able to recover a dense-detailed 3D model of a complex dynamic scene, from its two perspective images, *without object-level motion segmentation* as an essential intermediate step.

2. Separate treatment for rigid SfM and non-rigid SfM

Our investigation shows that the framework for reconstructing deformable object often differs from rigidly moving object. Not only solutions, but even the assumptions varies significantly e.g orthographic projection, low-rank shape [22] [44] [119] [107]. The reason for such inadequacy is perfectly valid due to the under-constraint nature of the problem itself. This motivate us to our next goal *i.e* "*To achieve 3D reconstruction of deformable object and complex dynamic scene under similar assumptions and same formulation.*"

To accomplish this goal for any arbitrary non-rigid deformation still remains an open problem. However, experiments suggest that our framework under some basic assumptions about the scene and the deformation, can reconstruct a general dynamic scene irrespective of the scene rigidity type. Thanks to the recent advancements in the dense optical flow algorithm's [12] [31] which are able to capture smooth non-rigid deformation over frames. These robust dense feature correspondences gives us the opportunity to exploit local motion. Thus, makes our formulation competent enough to bridge this gap between rigid and non-rigid SfM.

Assumptions: The two basic assumptions we used about the scene are: 1) the deformation of the scene between two frame is *locally-rigid,* but *globally as-rigid-as-possible*. 2) the structure of the scene in each frame can be approximated by a *piecewise planar smooth surface*.

We call our new algorithm the *SuperPixelSoup* algorithm, for reasons discussed in Section §6.4.1. Fig. (6.1) show sample reconstruction results obtained using our algorithm. The main contributions of this work are:

- a) A framework which disentangle object level motion segmentation for dense 3D reconstruction of a complex dynamic scene.

- A common framework for dense two-frame 3D reconstruction of a complex dynamic scene (including deformable objects), which achieves state-of-the-art performance.
- A new idea to resolve the inherent relative scale ambiguity problem in monocular 3D reconstruction by exploiting the as-rigid-as-possible (ARAP) constraint [161].

6.3 Prior works

The existence of a solution to SfM can be traced back to almost four decades ago [178]. Hence, this field of research can be considered as one the most researched field in the computer vision community. Since its inception this area has undergone a prodigious development and now practical algorithms are available which facilitates live dense reconstruction of a rigid scene [133] [135].

Even after such a remarkable development in this field, the choice of algorithm depends on the complexity of the object motion and the environment. Therefore, even now researchers are actively working in this area. One can also consider our work as an attempt to bridge this gap of rigid SfM and non-rigid SfM (NRSfM) which are most often treated as separate problems. Our work utilizes the idea of rigidity (locally) to solve dense reconstruction of a general dynamic scene. This concept of rigidity is not new in structure from motion problem [178] [126] and it has been effectively applied as a global constraint to solve large scale reconstruction problem [4]. This global rigidity to solve structure and motion has also been exploited to solve reconstruction over multiple frames at the same time via factorization approach [170].

Since, the literature on structure from motion and its treatment to different scenario's is very extensive. For brevity, we give a brief review only to previous works for monocular dynamic reconstruction that are of direct relevance to our method. Linear low-rank model has been used for dense nonrigid reconstruction. E.g., Garg *et al.* [65] solved the task with an orthographic camera model assuming feature matches across multiple frames. Fayad *et al.* [55] recovered deformable surfaces with a quadratic approximation, again from multiple frames. Taylor *et al.* [166] proposed a piecewise rigid solution using locally-rigid SfM to reconstruct a soup of rigid triangles.

While Taylor *et al.* [166] method is conceptually similar to ours, there are major differences:

1. We achieve *two-view* dense reconstruction while [166] relies on multiple views ($N \geq 4$).
2. We use *perspective camera model* while they rely on an orthographic camera model..

3. We solve the scale-indeterminacy issue, which is an inherent ambiguity for 3D reconstruction under perspective projection, while Taylor *et al.* [166] method does not suffer from this at the cost of being restricted to the orthographic camera model.

Recently, Russel *et al.* [152] and Ranftl *et al.* [149] used object-level segmentation for dense dynamic reconstruction. In contrast, our method is free from object segmentation, hence circumvent the difficulty associated with motion segmentation in a dynamic setting.

Template based approach is yet another method for deformable surface reconstruction. Yu et.al. [200] proposed a direct approach to capture dense, detailed 3D geometry of generic, complex non-rigid meshes using a single RGB camera. While it works for generic surfaces, the need of a template prevent its wider application to more general scenes. Wang [192] introduced a template-free approach to reconstruct a poorly-textured, deformable surface. However, its success is restricted to a single deforming surface rather than an entire dynamic scene. Varol et.al. [184] reconstructed deformable surfaces based on a piecewise reconstruction by assuming overlapping patches to be consistent over the entire surface and its also limited to the reconstruction of a single deformable surface.

6.4 Outline of the Algorithm

Before providing the details about our algorithm, we would like to introduce some common notations that are used throughout the paper.

Notation: In our formulation, we represent two perspective images as I, $I' : \Omega \rightarrow \mathbb{R}^3$ $|\Omega \subset \mathbb{Z}^2$ also referred as reference image and next image respectively. Vectors are represented by lower case letters, such as 'x' and matrices are represented by upper case letters such as 'X'. The subscript 'a', 'b' in a vector denotes anchor point and boundary point vectors respectively, for example x_{ai}, x_{bi} represents anchor point and boundary point vector corresponding to i^{th} superpixel in the image space. For now, we are just introducing notations, exact meaning of these terms will be introduced in the later sections §6.4.2. The 1-norm, 2-norm of a vector is denoted as $|.|_1$ and $\|.\|_2$ respectively. For matrices, Frobenius norm is denoted as $\|.\|_F$.

6.4.1 Overview

Given two perspective images I, I′ of a general dynamic scene, our goal is to recover the dense 3D structure of the scene. We first over-segment the reference image into superpixels, then model the deformation of the scene by union of piece-wise rigid motions of these superpixels. Specifically, we divide the overall non-rigid reconstruction into small rigid reconstruction for

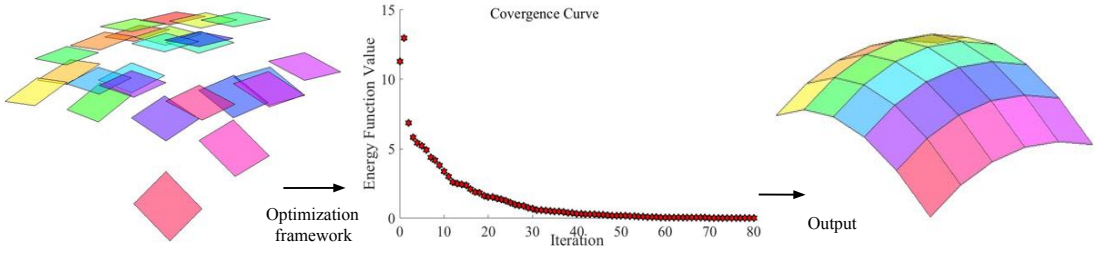


Figure 6.2: Reconstructing a 3D surface from a soup of un-scaled superpixels via solving a 3D Superpixel Jigsaw puzzle problem.

each individual superpixel, followed by an assembly process which glues all these individual local reconstructions in a globally coherent manner. While the concept of the above divide-and-conquer procedure looks simple, there is however a fundamental difficulty of *scale indeterminacy* in its implementation. Scale-Indeterminacy refers to the well-known fact that using a moving camera one can only recover the 3D structure up to an unknown scale. In our method, the individual rigid reconstruction of each superpixel can only be determined up to an unknown scale, the assembly of the entire non-rigid scene is only possible if and only if these scales among the superpixels are solved —which is however a challenging open task itself.

In this chapter, we show how this can be done, under our two very mild assumptions §6.2. Under these assumptions, our method solves the unknown relative scales and obtains a globally coherent dense 3D reconstruction of a complex dynamic scene from its two perspective views. Intuitively, our new method can be understood with the following intuition: Suppose every individual superpixel corresponds to a small planar patch moving rigidly in 3D space. Since the correct scales for these patches are not determined, they are floating in 3D space as a set of unorganized superpixel soup. Our method then starts from finding for each superpixel an appropriate scale, under which the entire set of superpixels can be assembled (glued) together coherently, forming a piecewise smooth surfaces, *as if* playing the game of "3D jigsaw puzzle". Hence, we call our method the "SuperPixel Soup" algorithm (see Fig.(6.2) for a conceptual visualization).

6.4.2 Problem Statement

To implement the above idea of piecewise rigid reconstruction, we first partition the reference image I into set of superpixels $\xi_I = \{s_1, s_2, .., s_i, .., s_N\}$, where each superpixel s_i is parametrized by its boundary pixels $\{x_{bi} = [u_{bi}, v_{bi}, 1]^T | b = 1, ..., B_i\}$ and an *anchor point*

x_{ai} corresponding to the centroid of the i^{th} superpixel in the image plane. To infer any pixel inside a superpixel, we use the operator $\Psi(.)$, for e.g $\Psi(s_i^j)$ will give the coordinates of j^{th} pixel inside s_i. Such a superpixel partition of the image plane naturally induces a piecewise-smooth segmentation of the corresponding 3D scene surface. We denote this set of 3D scene surfaces as $\Xi_W = \{\mathfrak{s}_1, \mathfrak{s}_2, ...\mathfrak{s}_i, ...\mathfrak{s}_N\}$. Although *surfel* is perhaps a better term, we nevertheless call it "3D superpixel" for the sake of easy exposition. We further assume each 3D superpixel ('$\mathfrak{s}_i$') is a small 3D *planar patch* $\Pi_{\mathfrak{s}i} = \left\{ n_i, \mathfrak{x}_{ai}, \{\mathfrak{x}_{bi}\} : (n_i, \mathfrak{x}_{ai}) \in \mathbb{R}^3 \text{ and } \{\mathfrak{x}_{bi}\} \in \mathbb{R}^{3\times B_i} \right\}$, which is parameterized by surface normal n_i, 3D anchor-point $\mathfrak{x}_{ai}$, and 3D boundary-points $\{\mathfrak{x}_{bi}\}$ (*i.e.* these are the pre-images of x_{ai} and $\{x_{bi}\}$). Assume every 3D superpixel $\mathfrak{s}_i$ moves rigidly according to

$$M_i = \begin{pmatrix} R_i & \lambda_i \hat{t}_i \\ 0 & 1 \end{pmatrix} \in SE(3) \tag{6.1}$$

where R_i represents relative rotation, $\hat{t}_i$ is the translation direction, and λ_i the unknown scale.

With the symbols and the notation preparation, we are in a position to put our idea in a more precise way: Given two intrinsically calibrated perspective images I and I′ of a generally dynamic scene and the corresponding optical flow field, our task is to reconstruct a piecewise-planar approximation of the dynamic scene surface. The deformable scene surface in the reference frame (i.e, Ξ_W) and the one in the second frame (i.e, Ξ'_W) are parametrized by their respective 3D superpixels $\{\mathfrak{s}_i\}$ and $\{\mathfrak{s}'_i\}$, where each $\mathfrak{s}_i$ is described by its surface normal n_i and an anchor point $\mathfrak{x}_{ai}$. Any 3D plane can be determined by an anchor point $\mathfrak{x}_{ai}$ and a surface normal n_i. If one is able to estimate correct placement of all the 3D anchor points and all the surface normals corresponding to the reference frame, the problem is solved, since each element of Ξ_W is related to Ξ'_W via SE(3) transformation (locally rigid).

The overall procedure of our method is presented in Algorithm 1.

6.4.3 Formulation

We begin by briefly reiterating some of our representation. We partition the reference image into a set Ξ_I, whose corresponding set in the 3D world is Ξ_W. Equivalently, Ξ'_I and Ξ'_W are the respective sets for the next frame. The mapping of each element in the reference frame and next frame differs by a rigid transformation. Mathematically, $\Xi_W \mapsto \Xi'_W$ via SE(3) transformation (also known as special euclidean group), for instance $\mathfrak{x}'_{ai} = M_i \mathfrak{x}_{ai}$ where $\mathfrak{x}'_{ai} \subset \Xi'_W$ and $\mathfrak{x}_{ai} \subset \Xi_W$. In our formulation each 3D plane is described by $\Phi_{\mathfrak{s}i} = \Big\{ (\Pi_{\mathfrak{s}i}, M_i) \mid \forall\, i \in [1, \mathcal{N}], \exists \{\Pi_{\mathfrak{s}i}\}, M_i \Big\}$, where $\mathcal{N}$ is the total number of superpixels (see Fig. 6.3). Similarly, in the image

Algorithm 4 : SuperPixel Soup

Input: Two consecutive image frames of a dynamic scene and dense optical flow correspondences between them.
Output: 3D reconstruction for both images.
1. Divide the reference image into '$\mathcal{N}$' superpixels and construct a K-NN graph to represent the entire scene as a graph $G(V, E)$ defined over these superpixels §6.4.3.
2. Employ two-view epipolar geometry to recover the rigid motion and shape for each 3D superpixel §6.4.4.
3. Optimize the proposed energy function to assemble (or glue) and align all the reconstructed superpixels ("3D Superpixel Jigsaw Puzzle") §6.4.4.
Note: The procedure of the above algorithm looks simple; there is, however, a fundamental difficulty of scale indeterminacy in its execution.

space $\xi_I \mapsto \xi'_I$ through the plane-induced homography.

$$s_i' = K\left(R_i - \frac{\lambda_i t_i n_i^T}{\lambda_i d_i}\right) K^{-1} s_i^{\star} \tag{6.2}$$

Here, K is the intrinsic camera matrix and d_i is the depth of the plane. Using these notations and definition, we define a K-NN graph.

Build K-NN graph: Using over-segmentation of the reference images which is the projection of the set of 3D planes Φ_{si}, we construct a K-NN graph G(V, E) to build the relation between anchor points. The graph vertices (V) are anchor points, which connects with each other via graph edges (E). To be precise, the distance between any two vertices ($E_i \subset E$) is defined as the Euclidean distance between them. Here, we assume euclidean distance as a valid graph metric to describe the edge length between any two local vertices. Such assumption is valid for local compactness(Euclidean spaces are locally compact). Interested reader may refer to [23] [194] [193] for comprehensive details. The intension behind constructing this graph is to constrain the motion and continuity of the space (defined in terms of optical flow, depth). To establish the constraints to be strong enough, we allow each anchor point to build its relation beyond its immediate neighbors (see Fig. 6.4).

Constructing this K-NN graph is very crucial in the establishment of local rigidity constraint which is the basis of our assumption. This assumption allows the shape to be as rigid as possible globally and rigid locally.

*scale λ_i is introduced both in the numerator and denominator for clarification that scale does not affect the homography transformation.

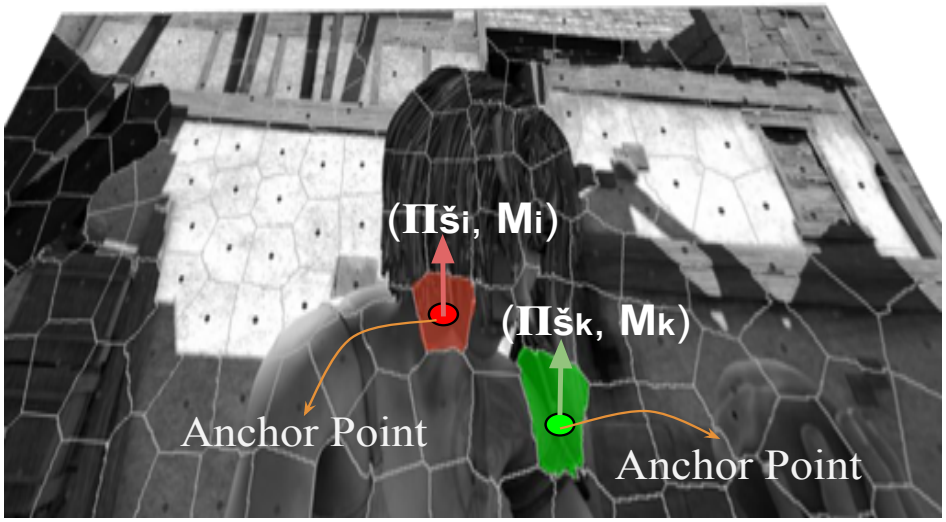


Figure 6.3: The modeling of a continuous scene with a piece wise rigid and planar assumption. Each superpixel is composed of a set $(\Pi_{\check{s}i}, M_i)$ where $\Pi_{\check{s}i}$ contains geometric parameters such as normal, anchor point, boundary points of a plane in 3D and M_i contains the motion parameters i.e rotation and translation.

As-Rigid-As-Possible (ARAP) Energy Term: Our method is built upon the idea that the correct scales of 3D superpixels can be estimated by enforcing prior assumptions that govern the deformation of the dynamic surface. Specifically, we require that, locally, the motion that each 3D-superpixel undergoes is rigid, and globally the entire dynamic scene surface must move as rigid as possible (ARAP). In other words, while the dynamic scene is globally non-rigid, its deformation must be *regular* in the sense that it deforms as rigidly as possible. To implement this idea, we define an ARAP-energy term as:

$$E_{\text{arap}} = \sum_{i=1}^{N} \sum_{k \in \mathcal{N}_i} w_1(x_{ai}, x_{ak}) \Big(\|R_i - R_k\|_F + \|\lambda_i \hat{t}_i - \lambda_k \hat{t}_k\|_2 \Big) + w_2(x_{ai}, x_{ak}) . \Big| \|\mathbf{x}_{ai} - \mathbf{x}_{ak}\|_2 - \|\mathbf{x}'_{ai} - \mathbf{x}'_{ak}\|_2 \Big|_1 . \tag{6.3}$$

Here, the first term favors smooth motion between the local neighbors, while the second term encourages inter-node distances between the anchor node and its K nearest neighbor nodes (denoted as $k \in \mathcal{N}_i$) to be preserved before and after motion (hence as-rigid-as-possible, see Fig. 6.4). We define the weighting parameters as:

$$w_1(x_{ai}, x_{ak}) = w_2(x_{ai}, x_{ak}) = \exp(-\beta \|x_{ai} - x_{ak}\|). \tag{6.4}$$

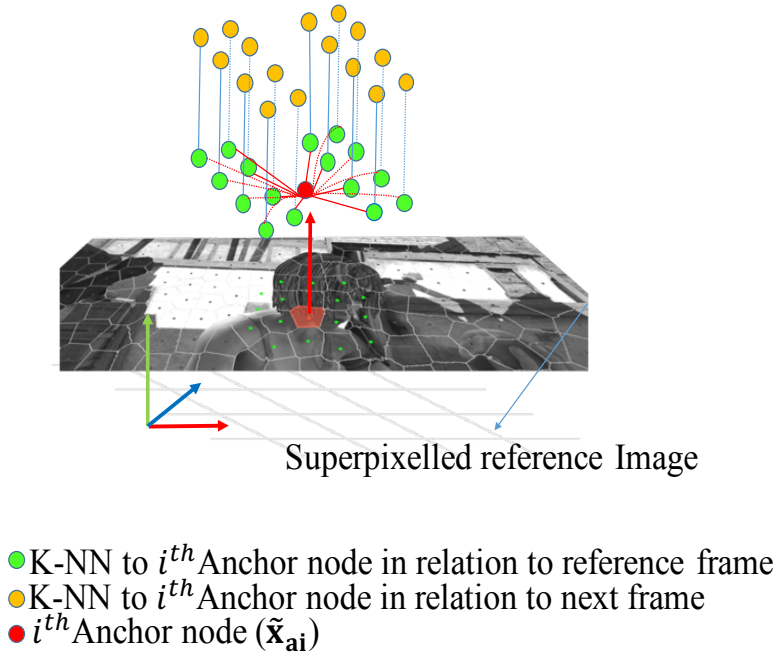


Figure 6.4: Demonstration of as rigid as possible constraint. Superpixel segmentation in the reference frame is used to decompose the entire scene as a set of anchor points. Schematic representation shows the construction of K-NN around a particular anchor point (shown in Red). We constrain the local 3D coordinate transformation both before and after motion (green shows K-NN the reference frame, yellow shows the relation in the next frame (after motion)). We want this transformation to be as rigid as possible.

These weights are set to be inversely proportional to the distance between two superpixels. This is to reflect our intuition that, the further apart two superpixels are, the weaker the E_{arap} energy is. Although there may be redundant information in these two terms w.r.t scale estimation, we keep them for motion refinement §6.4.4. Note that, this term is only defined over anchor points, hence it enforces no depth smoothness along boundaries. The weighting term in E_{arap} advocates the local rigidity by penalizing over the distance between anchor points. This allows immediate neighbors to have smooth deformation over time. Also, note that E_{arap} is generally *non-convex*. This non-convexity arises due to the second term in Eq.(6.3), where we allow for discontinuity by introducing l_1 norm on top of the difference of two l_2 norm term. In Eq. (6.4) β is a trade-off constant chosen empirically.

E_{arap} alone is good enough to provide reasonably correct scales, however, piece-wise planar composition of a continuous 3D space creates discontinuity near the boundaries of each plane. For this reason, we incorporate additional constraints to fix this depth discontinuity

and further refine motions and geometry for each superpixel via neighboring relations. We call these constraints as Planar Re-projection, 3D Continuity and Orientation Energy constraint.

Planar Re-projection Energy Term: With the assumption that each superpixel represents a plane in 3D, it must satisfy corresponding planar reprojection error in 2D image space. This reprojection cost reflects the average dissimilarity in the optical flow correspondences across the entire superpixel due to motion. Therefore, it helps us to constrain the surface normals, rotation and translation direction such that they obey the observed planar homography in the image space. Let $|\psi(s_i)| = \mathcal{Q}$

$$E_{\text{proj}} = \sum_{i=1}^{N} \frac{w_3}{\mathcal{Q}} \sum_{j=1}^{\mathcal{Q}} \|\psi(s_i^j)' - K(R_i - \frac{t_i n_i^T}{d_i})K^{-1}\psi(s_i^j)\|_2. \tag{6.5}$$

Here, $\psi(s_i^j)$, $\psi(s_i^j)'$ is the optical flow correspondence of j^{th} pixel in the reference frame and next frame of i^{th} superpixel respectively. The operator $|.|$ represent the cardinal number of a set. w_3 is a trade-off scalar chosen empirically. A natural question that may arise is: *This term is independent of scale, then what's the purpose of this constraint? How does it help?* Kindly, refer to §6.4.4 for details.

3D Continuity Energy Term: In case of a dynamic scene, where both camera and the objects are in motion, its quite apparent that the scene will undergo some changes across frames. Hence, to assume unremitting global continuity with piece-wise planar assumption, in a dynamic scene is unreasonable. Instead, local weak continuity constraint can be enforced —a constraint that can be broken occasionally [89] i.e local planes are connected to few of its neighbors. Accordingly, we want to allow local neighbors to be piece-wise continuous. To favor this continuous or smooth surface reconstruction, we require neighboring superpixels to have smooth transition at their boundaries. To do so, we define a 3D continuity energy term as:

$$E_{\text{cont}} = \sum_{i=1}^{N} \sum_{k \in \mathcal{N}_i} w_4(X_{bi}, X_{bk}) \left(\|\mathbf{X}_{bi} - \mathbf{X}_{bk}\|_F + \varrho(\|\mathbf{X}'_{bi} - \mathbf{X}'_{bk}\|_F\right) \tag{6.6}$$

where, X, $\mathbf{X}$ represents the corresponding matrices in 2D image space and 3D euclidean space ($X_{bi} \in \mathbb{R}^{2\times Bi}$, $\mathbf{X}_{bi} \in \mathbb{R}^{3\times Bi}$, where Bi is the total number of boundary pixel for i^{th} superpixel). Since in our representation, geometry and motion are shared among all pixels within a superpixel, so regularization within the superpixel is not explicitly needed. Thus, we only

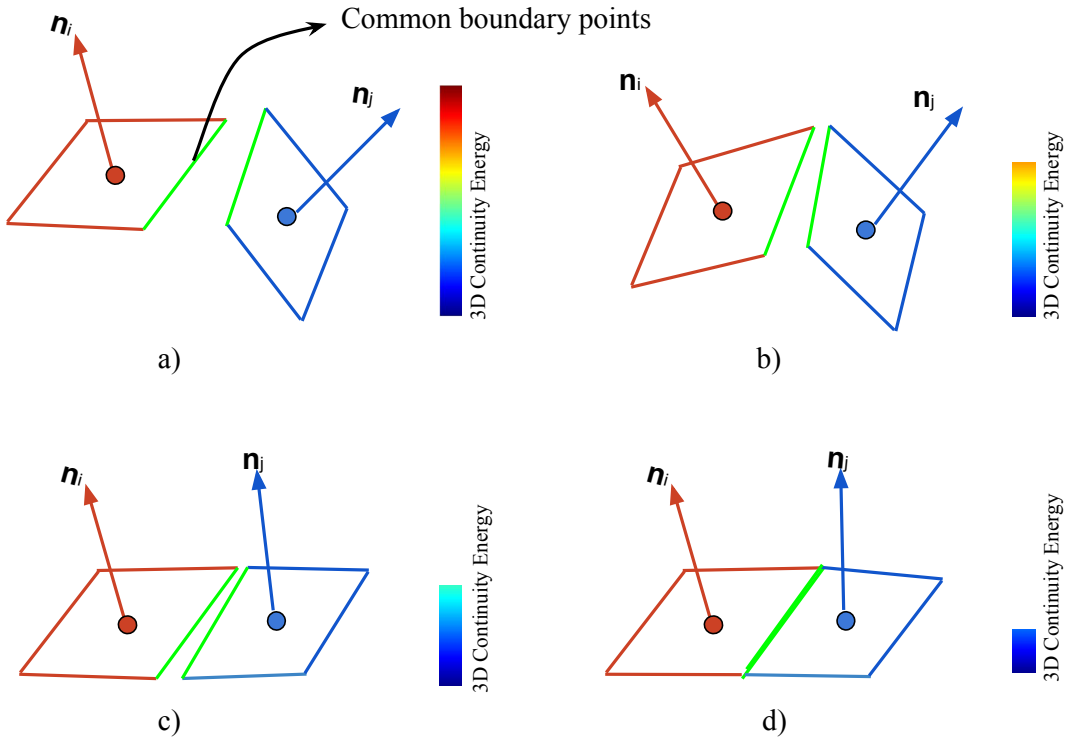


Figure 6.5: 3D Continuity energy favors continuous surface for the planes that shares the common boundary points. a)-d) The lesser the E_{cont} is, smoother the surface becomes (color bar shows the energy).

concentrate on the shared boundary pixels to regularize our energy. Note that the neighboring relationship in E_{cont} is different from E_{arap} term. Here, the neighbors share common boundaries with each other.

To encourage the geometry to be approximately smooth locally if the object has similar appearance, we color weight the energy term along the boundary pixels. For each boundary pixel of a given superpixel, we consider its 4-connected neighboring pixels to weight. Using this idea for w_4 we obtain:

$$w_4(\mathrm{X_{bi}}, \mathrm{X_{bk}}) = \sum_{j=1}^{4} \exp(-\beta \|\mathrm{I}(\mathrm{X_{bi}}) - \mathrm{I}(\zeta_j)\|_F) \tag{6.7}$$

which weigh the inter-plane transition by color difference. The symbol $\zeta_j \in \mathbb{R}^{2\times Bi}$ is a set that contains the 4 connecting pixels to each i^{th} superpixel boundary pixel shared with k^{th} super-

pixel. The color based weighting term plays an important role to allow for "weak continuity constraint"*i.e.* gradually allowing for occasional discontinuity [89] [17].

To better understand the implication of E_{cont} constraint, consider two boundary points in the image space $a, b \in \mathbb{R}^2$. Generally, if these two points lie on a different plane, it will not coincide in the 3D space before and after motion. Hence, we compute the 3D distance between boundary pixels corresponding to both reference frame and next frame, which leads to our goal of penalizing distance along shared edges (see Fig. 6.5). Therefore, this term ensures the 3D coordinates across superpixel boundaries to be continuous in both frames. The challenge here is to reach a satisfactory solution of overall scene continuity, almost everywhere in both the frames [18]. In the Eq.(6.6) ϱ is a truncation function defined as $\varrho = \min(., \sigma)$ and similar to Eq.(6.4) β in Eq.(6.7) a trade-off constant, chosen empirically.

Orientation Energy Term: To encourage the smoothness in the orientation of the neighboring planes, we added one more geometric constraint *i.e,* E_{orient} defined as follows.

$$E_{\text{orient}} = \sum_{i=1}^{N} \sum_{k \in \mathcal{N}_i} \varrho_n \left(1 - n_i{}^T n_k\right) \tag{6.8}$$

Here neighbor index are same as 3D continuity term. ϱ_n is truncated l_1 penalty function.

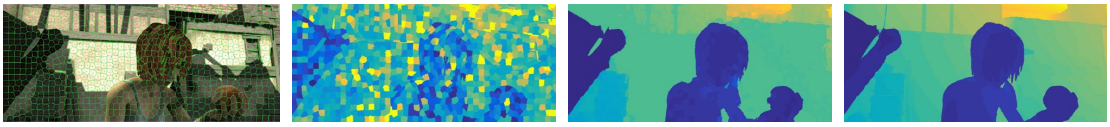
Combined Energy Function: Equipped with all these constraints, we define our overall cost function or energy function to obtain a scale consistent 3D reconstruction of a complex dynamic scene. Our goal is to estimate depth (d_i), surface normal (n_i) and scale λ_i for each 3D planar superpixel. The key is to estimate the unknown relative scale λ_i. We solve this by minimizing the following energy function:

$$\begin{aligned} &\min_{i, n_i, d_i, R_i, t_i} E = E_{\text{arap}} + \alpha_1 E_{\text{proj}} + \alpha_2 E_{\text{cont}} + \alpha_3 E_{\text{orient}} \\ &\text{subject to} \sum_{i=1}^{N} \lambda_i = 1, \lambda_i > 0. \\ &R_i \in \mathbb{SO}(3), \|n_i\|_2 = 1. \end{aligned} \tag{6.9}$$

The equality constraint on λ fixes the unknown freedom of a global scale. The constraint on R_i is imposed to restrict the rotation matrix to lie on $\mathbb{SO}(3)$ manifold. The constant $\alpha_1, \alpha_2, \alpha_3$ are included for numerical consistency.

6.4.4 Implementation

We partition the reference image into about 1,000-2,000 superpixels [3]. Parameters such as α_1, α_2, α_3, β, σ were tuned differently for different datasets. To perform optimization of



a) b) c) d)

Figure 6.6: a) Superpixelled reference image b) Individual superpixel depth with arbitrary scale (*unorganised superpixel soup*) c) recovered depth map using our approach (*organised superpixel soup*) d) ground-truth depth map.

the proposed energy function (Eq. 6.9), we require initial set of proposals for motion and geometry.

Initial Proposal Generation

We exploit piece-wise rigid and planar assumption to estimate initial proposal for geometry and motion. We start by estimating homography (H_i) for each superpixel using dense feature correspondences. Piece-wise rigid assumption allows us for evaluating approximate rotation and correct translation direction via triangulation and chierality check [85] [88]. To obtain the correct normal direction and initial depth estimate, we solve the following system of equations for each superpixel:

$$H_i = K(R_i - \frac{t_i n_i}{d_i})K^{-1} \tag{6.10}$$

The reason we choose this strategy to obtain normal is because a simple decomposition of homography matrix to rotation, translation and normal can lead to sign ambiguity [184] [131]. Nevertheless, if one has correct rotation and direction of translation –which we infer from chierality check, then inferring normal becomes easy†. Here, we assume the depth 'd_i' to be a positive constant and the initial arbitrary reconstruction is in the +Z direction. This strategy of gathering 9-dimensional variables (6-motion variable and 3-geometry variable) for each individual superpixel gives us a good enough estimate to get started with the minimization of our overall energy function ‡.

To initialize 3D vectors in our formulation use the following well known relation:

$$\mathbf{x}_{ai} = \left[\left(\frac{u_{ai} - c_x}{f_x}\right), \left(\frac{v_{ai} - c_y}{f_y}\right), 1/n_i{}^T K^{-1} \begin{pmatrix} u_{ai} \\ v_{ai} \\ 1 \end{pmatrix} \right]^T (\lambda_i d_i) \tag{6.11}$$

where, (u_{ai}, v_{ai}) are image coordinates and (c_x, c_y, f_x, f_y) are camera intrinsic parameters which can be inferred from K matrix.

†The solution for the obtained normal must be normalized.

‡If the size of the superpixel is very small kindly use the neighbors to estimate motion parameters.

Optimization

With the good enough initialization of variables, we start to optimize our energy function Eq.(6.9). A global optimal solution is hard to achieve due to the non-convex nature of the proposed cost function (Eq. (6.9)). However, it can be solved efficiently using interior-point methods [16] [15]. Although the solutions found by the interior point method are at best local minimizers, empirically they appear to give good 3D reconstructions. In our experiments, we initialized all λ's with an initial value of $\frac{1.0}{N}$.

Next, we employ a particle based refinement algorithm to rectify our initial motion and geometry beliefs. Specifically, we used the Max-Product Particle Belief propagation (MP-PBP) procedure with the TRW-S algorithm [97] to optimize over the surface normals, rotations, translations and depths for all 3D superpixels using Eq.(6.12). We generated 50 particles as proposals for the unknown parameters around the already known beliefs to initiate refinement moves. Repeating this strategy for 5-10 iterations, we obtain a smooth and refined 3D structure of the dynamic scene.

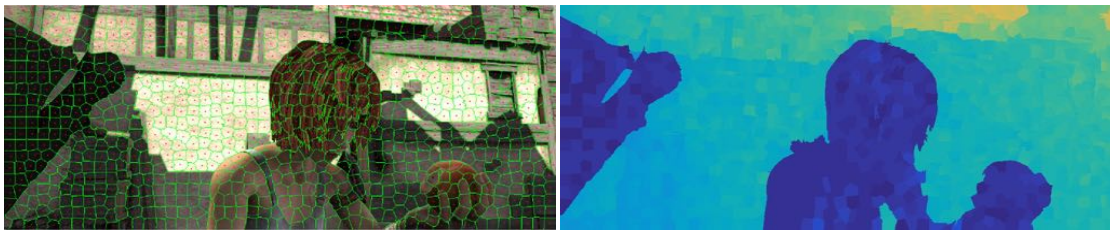
$$E_{\text{ref}} = E_{\text{arap}} + \alpha_1 E_{\text{proj}} + \alpha_2 E_{\text{cont}} + \alpha_3 E_{\text{orient}} \tag{6.12}$$

Why particle based filtering is required? Assigning superpixels to a set of planes can lead to non-smooth blocky effect at their boundaries. Under our formulation, scale assignment to each plane is governed by its anchor point. Now, even if the neighboring planes have similar scale, it may be misaligned in the world coordinate, violating the geometry of the scene. Moreover, as we set the proposal for planes to be limited due to practical consideration, we may not be able to assign accurate depth to each pixels. But, we can definitely do better by refining our solution using TRW-S [97] or similar algorithm's subjected to computational constraint.

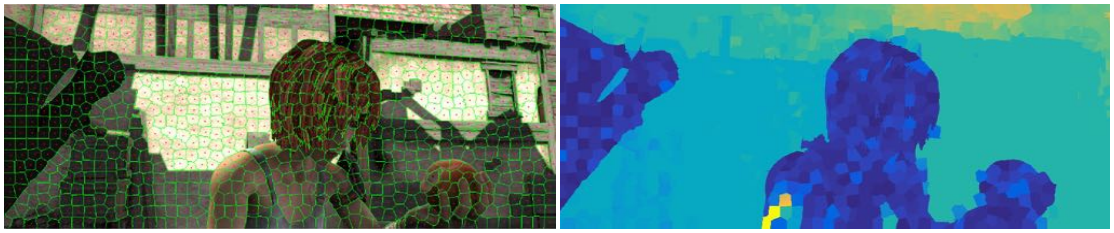
Use of continuous optimization method: We also used gradient based approach such as BFGS and L-BFGS to optimize our objective function using available MATLAB library. Although it showed some improvement in terms of reconstruction accuracy, however, it consumes a lot of time (30 minutes or more) to provide approximate solution. Consequently, we confine ourself to discrete approach for evaluations.

6.5 Experimental Evaluation

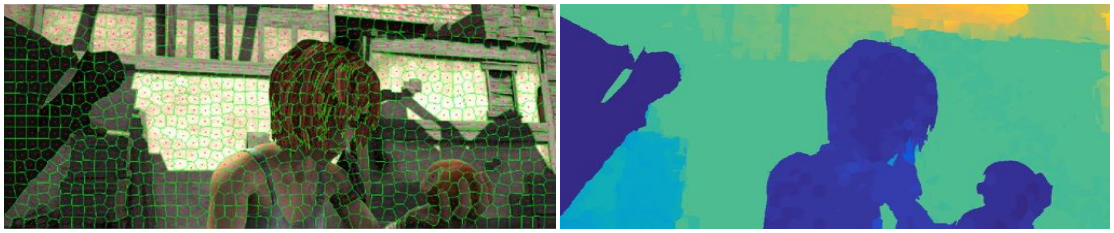
We evaluated our formulation both qualitatively and quantitatively on various standard benchmark datasets, namely MPI Sintel [24], KITTI [68], VKITTI [59] and You-Tube Object



Result with only rigid as possible term**(E_arap)**



Result with planar re-projection, 3D continuity term and orientation **(E_proj + E_cont + E_orient)**



Result with all terms **E_arap** + **E_proj** + **E_cont** + **E_orient**.

Figure 6.7: Effects of using "as rigid as possible", "Planar re-projection", "3D continuity" and "Orientation" term. Top row: By enforcing the "as rigid as possible" term only, the recovered relative scales are correct but the reconstructed planes are misaligned with respect to their neighbors. Middle row: With the planar re-projection, 3D continuity and orientation term enforced, the resultant 3D reconstruction achieves continuous neighboring boundaries, however, the relative scales for every plane in 3D is not correct. Bottom row: By enforcing the the "as rigid as possible" term along with all the other smoothness terms, we can handle both relative scales and 3D reconstruction for a complex dynamic scene.

dataset [147]. All these dataset contains images of dynamic scene where both camera and objects are in motion w.r.t each other. To test the reconstruction results on deformable objects we used Paper, T-shirt [184] [183] and Back Sequence [65]. Before we dive into our experimental analysis on the aforementioned datasets, we would like to show and briefly discuss the role of different terms in our formulation.

Ablation Analysis

Firstly, in the proposed optimization framework the *3D continuity* term is defined over boundaries between neighboring superpixels, which alone is not sufficient to constrain the motion beyond its immediate neighbors. Secondly, $_{\text{proj}}$ and $_{\text{orient}}$ has nothing to do with scale computation whatsoever. Hence, combining these three terms is not good enough to explain the correct scales for each of the object present in the scene. On the other hand, *as rigid as possible* term is defined for each superpixel's anchor point over the K-NN graph structure. However, it does not take into account the alignment of planes in 3D along the boundaries. As a result overall reconstruction suffers. Thus, this demonstrates that all the terms are essential for reliable dynamic 3D reconstruction. Fig.(6.7) illustrates the contribution of different terms toward the final reconstruction result.

Evaluation

We aim at an evaluation as comprehensive as possible and as a result we evaluated our method with different kind of scenes –*rigid, non-rigid, complex dynamic scene i.e composition of both rigid and non-rigid*, available with benchmark datasets. We selected the most commonly used error metric to evaluate the fidelity of the depth map.

Evaluation Metric

For quantitative evaluation, the errors are reported in mean relative error (MRE), defined as $\frac{1}{P}\sum_{i=1}^{P}|z^i_{\text{gt}} - z^i_{\text{est}}|/z^i_{\text{gt}}$. Here, z^i_{est}, z^i_{gt} denotes the estimated and ground-truth depth respectively with being the total number of points. The error is computed after re-scaling the recovered depth properly, as the reconstruction is obtained up to an unknown global scale. We used MRE for the sake of consistency with the previous work [149]. Quantitative evaluations for the YouTube-Objects dataset and the Back dataset are missing due to the absence of ground-truth results.

Experimental setup and processing time: We partition the reference image using SLIC superpixels [3]. We used current state-of-the art optical flow to compute dense optical flow [12]. To initialize the motion and geometry variables, we used the the procedure

Figure 6.8: Qualitative results using our algorithm in a complex dynamic scene. Example images are taken from MPI Sintel dataset [24]. Top row: Input reference image from *sleeping_1, sleeping_2, shaman_3, temple_2, alley_2* sequence (from left to right). Middle row: Ground-truth depth map for the respective frames. Bottom row: Recovered depth map using our method.

Figure 6.9: Qualitative results on KITTI Dataset [68]. The second row shows the obtained depth map for the respective frames. Note: Dense ground-truth depth data is not available with this dataset.

discussed in §6.4.4. Interior point algorithm [16] [15] and TRW-S [97] were employed to solve the proposed optimization. The implementation of the algorithm is done in MATLAB/C++. Our modified implementation (modified from our ICCV implementation[111]) takes on an average 15-20 minutes to converge for images of the size 1024×436 on a regular desktop with Intel core i7 processor (16 GB RAM) for 50 refinement particle per superpixel.

Results on MPI Sintel Dataset: We begin our analysis on the experimental results with MPI Sintel dataset [24]. This dataset is derived from animation movie featuring complex scenes. It contains highly dynamic sequences with large motions, significant illumination changes and non-rigidly moving objects. This dataset has emerged as a standard benchmark to evaluate dense optical flow algorithm's and recently, it has also been used in evaluation of dense 3D reconstruction methods for a general dynamic scenes [149].

The presence of non-rigid objects in the scene makes it a prominent choice for us to test our algorithm. It is a challenging dataset particularly for the piece-wise planar assumption

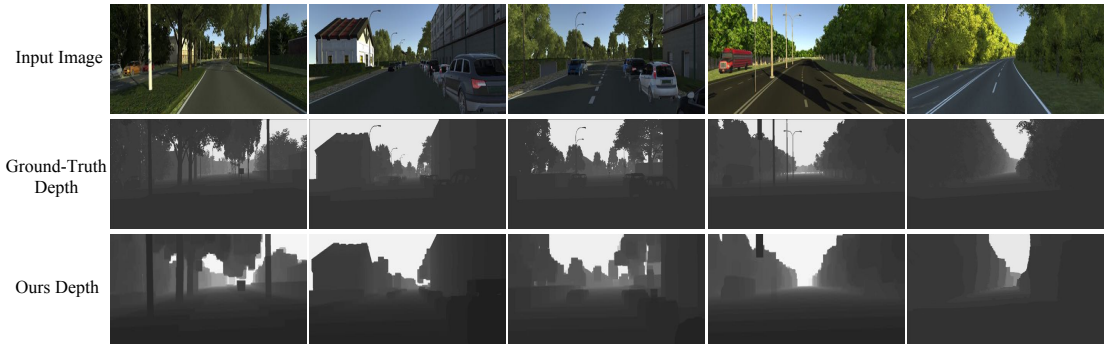


Figure 6.10: Qualitative results using our algorithm for the outdoor scenes. Examples are taken from VKITTI dataset [59]. Top row: Input reference image. Middle row: Ground-truth depth map for the respective frames. Bottom row: Recovered depth map using our method.

due to presence of many small and irregular shapes in the scene. Additionally, the presence of ground-truth depth map makes quantitative analysis much easier. We selected 120 pairs of images to test our method, which includes images from alley_1, ambush_4, mountain_1, sleeping_1 and temple_2. Fig.(6.8) shows some qualitative results on few images.

Results on KITTI Dataset: The KITTI dataset [68] features the real world outdoor scene targeting automobile application. Its images are acquired from camera mounted on the top of a car. It's a challenging dataset due to the fact that it contains images with large displacement of camera and realistic lighting condition. However, it only contains sparse ground-truth 3D information, which makes evaluation a bit strenuous. Nonetheless, it captures noisy real-life situation and therefore we believe it is well suited to test 3D reconstruction of a general dynamic scene. We selected 00-09 from odometry dataset to evaluate and compare our results. We calculated mean relative error only over the provided sparse 3D LiDAR points –after adjusting the global scale. Fig.(6.9) shows some qualitative results on few images.

Results on VKITTI Dataset: The Virtual KITTI dataset [59] contains computer rendered photo-realistic outdoor driving scenes which resemble the KITTI dataset. The advantage of using this dataset is that it provides perfect ground-truths for many measurements. Furthermore, it helps to simulate algorithm related to dense reconstruction with noise free and distortion-free images, facilitating quick experimentation. We selected 120 images from 0001_morning, 0002_morning, 0006_morning and 0018_morning. The qualitative results obtained are shown in Fig.(6.10).

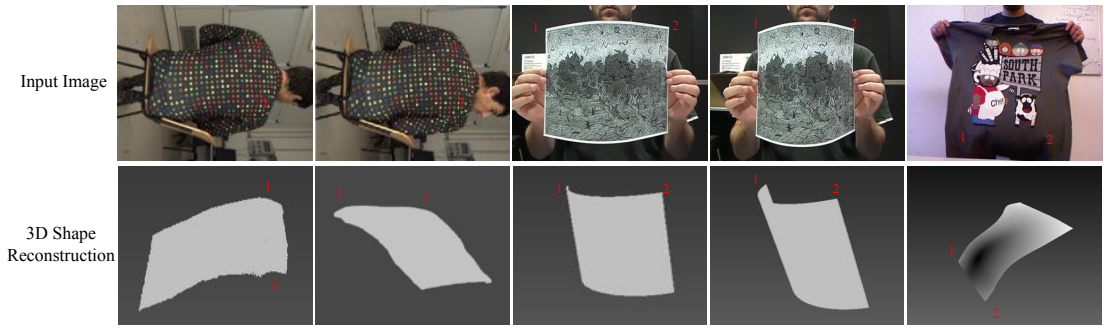


Figure 6.11: Dense 3D reconstruction of the objects that are undergoing non-rigid deformation over frames. Top row: Input reference frame from *Back sequence* [65], *Paper sequence* [184][183] and *t-shirt sequence*[184][183]. Bottom row: Qualitative 3D reconstruction results for the respective deforming object.

Method → (Method type)	DT [96] (SF)	GLRT [58] (MF)	BMM [44] (MF)	PTA [8] (MF)	DMDE [149] (TF)	Ours (TF)
MPI Sintel	0.4833	0.4101	0.3121	0.3177	0.297	0.1669
Virtual KITTI	0.2630	0.3237	0.2894	0.2742	-	0.1045
KITTI	0.2703	0.4112	0.3903	0.4090	0.148	0.1268
kinect_paper	0.2040	0.0920	0.0322	0.0520	-	0.0476
kinect_tshirt	0.2170	0.1030	0.0443	0.0420	-	0.0480

Table 6.1: Performance Comparison: this table lists the MRE errors. For DMDE [149] we used its previously reported result, as its implementation is not publicly available. Here, SF, MF and TF refers to single frame, multi-frame and two frame respectively.

Results on Non-Rigid Sequence: We also tested our method on some commonly used dense non-rigid sequence namely kinect_paper [184], kinect_tshirt [184] and back sequence [65][§]. Most of the benchmark approach to solve non-rigid structure from motion use multiple frames and orthographic camera model. Despite a two-frame method and perspective camera model, we are able to capture the deformation of non-rigid object and achieve its reliable reconstruction. Qualitative results for dense non-rigid object sequence are shown in Fig.(6.11). To compute the mean relative error, we align and scale our shape (fixing global ambiguity) w.r.t ground-truth shape.

[§]Note: The intrinsic matrix for back sequence is not available with the dataset, we estimated an approximate value of it using 2D-3D relation available from Garg *et al.* [65].

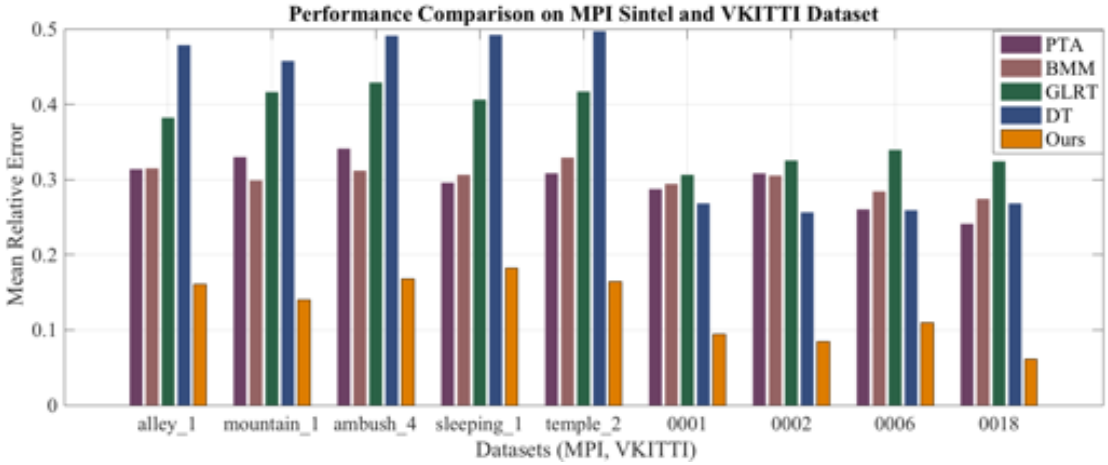


Figure 6.12: Quantitative comparison with our method with PTA [8], BMM [44], GLRT[58], DT [96] on benchmark datasets. The depth error is calculated by adjusting the numerical scale of the obtained depth map to the ground-truth value, to account for global scale ambiguity. Comparison on MPI Sintel [24] and Virtual KITTI [59]dataset. These numerical values show the fidelity of reconstruction that can be retrieved on these benchmark datasets using our formulation.

Comparison

The performance of our method is compared to several dynamic reconstruction methods, which include the Block Matrix Method (BMM) [44], Point Trajectory Approach (PTA) [8], Low-rank Reconstruction (GBLR) [58]), Depth Transfer (DT) [96], DMDE [149] and ULDEMV [204]. This comparison is made over the available benchmark datasets i.e MPI Sintel, KITTI, VKITTI, *T-shirt, Paper, Back*. Table 6.1 provides the statistical details of our results in comparison to the baseline approach. Clearly, our method outperforms others in outdoor sequence and provides a commendable performance for deformable sequence. While compiling the results per frame comparison is made over th entire sequence. Evaluation in the case of KITTI dataset is done only for the provided sparse 3D LiDAR points. Fig.(6.12), Fig.(6.13) and Fig. (6.16c) show per category statistical performance of our approach with other competing methods on benchmark dataset. Additionally, we also performed a qualitative comparison on MPI Sintel [24], KITTI[68] and You-Tube object dataset[147] (see Fig.(6.14) and Fig.(6.15)). It can be inferred that our method consistently delivers superior performance on all of these datasets.

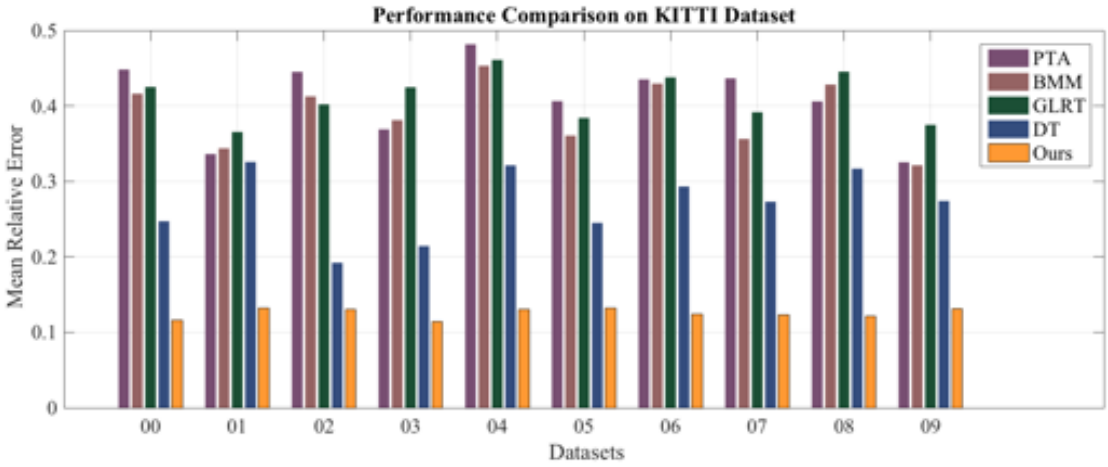


Figure 6.13: Quantitative comparison with our method with PTA [8], BMM [44], GLRT[58], DT [96] on benchmark datasets. The depth error is calculated by adjusting the numerical scale of the obtained depth map to the ground-truth value, to account for global scale ambiguity. Comparison on KITTI [68] dataset. These numerical values show the fidelity of reconstruction that can be retrieved on these benchmark datasets using our formulation.

Figure 6.14: Qualitative evaluation of our approach with the Video-PopUp [152]. Clearly, our method provides more dense and detailed reconstruction of the scene. In the second row t-shirt description is missing with Video-PopUp [152] approach. By contrast our method has no such holes. Note: The results presented here for Video-PopUp are taken from their webpage since the source code provided by the authors crashes frequently.

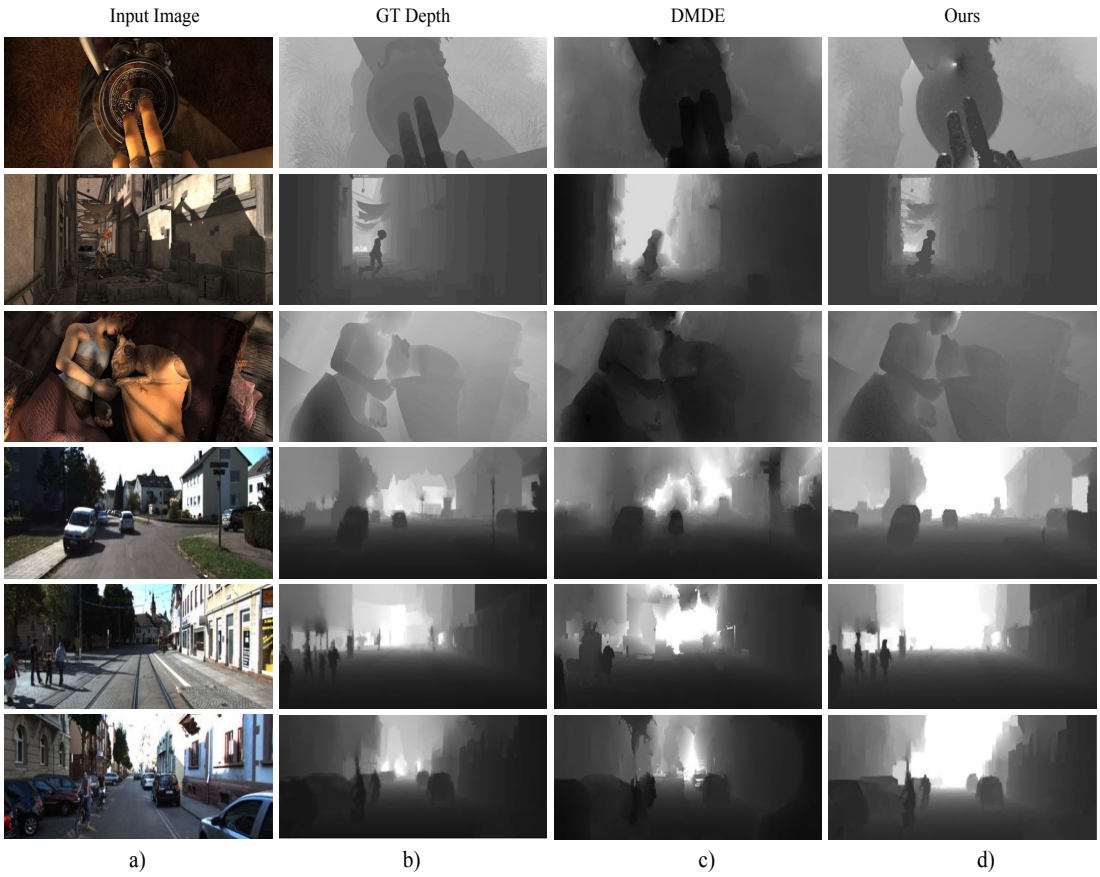
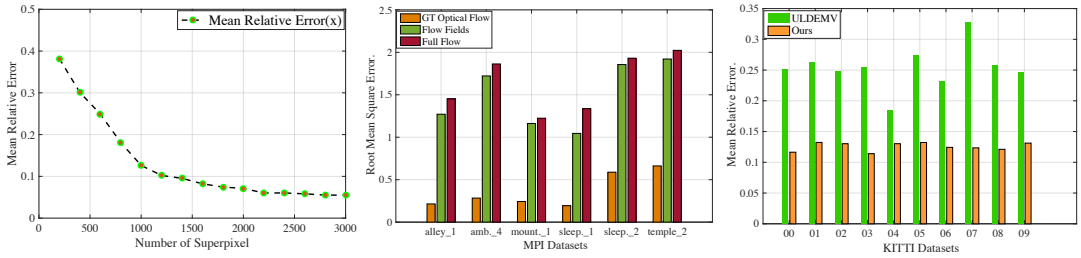


Figure 6.15: Qualitative comparison of our method with DMDE [149] on MPI Sintel [24] and KITTI Dataset [24]. Left to Right: For each input reference image, we show its ground-truth depth map (GT Depth), depth map reported by DMDE [149] and depth map obtained using our approach. Note: Dense GT depth map for KITTI Dataset is taken from DMDE [149] work.



(a) Dependence on superpixel model **(b)** Dependence on optical flow **(c)** Comparison with learning method

Figure 6.16: (a) Fluctuation in mean relative depth error with the change in number of superpixels. It can be observed that after 1000 superpixel the MRE more or less starts saturating with no significant effect on the overall accuracy. However, it was observed that the motion estimation becomes critical with the increase in number of superpixels. (b) Performance evaluation in RMSE (in meters) with the state-of-the-art optical flow methods in comparison to the ground-truth optical flow (MPI Sintel [24] dataset). (c) Mean Relative Depth Error comparison with a recently proposed unsupervised learning based approach (ULDEMV [204]) on KITTI dataset [68].

Performance Analysis

Besides statistical evaluation, we also conducted several other experiments to better analyze the performance of our algorithm. These experiment will better illustrate the different aspects of the proposed approach.

Performance with variation in number of superpixels: Our method uses SLIC based over segmentation of the reference frame to discretize the 3D space. Therefore, the number of superpixels that will be used to faithfully represent the real-world plays a crucial role in the accuracy of piece-wise continuous reconstruction. If the number of superpixel is very high the estimation of motion parameters becomes tricky and therefore neighboring superpixel are used to estimate rigid motion which leads to computation challenges. In contrast, few number of superpixels are unable to capture the intrinsic details of a complex dynamic scene. So, a trade-off between the two is often a better choice. Fig. (6.16a) shows the plot of depth error variations with respect to change in the number of superpixels.

Performance with different optical flow algorithm's: As our method needs dense optical flow correspondences between the frames, the performance of our method is directly dependent on the accuracy of the dense optical flow estimation. Therefore, to analyze the sensitivity of our method to different optical flow methods, we conducted experiments by testing our method with the ground-truth optical flow, and the state-of-the-art optical

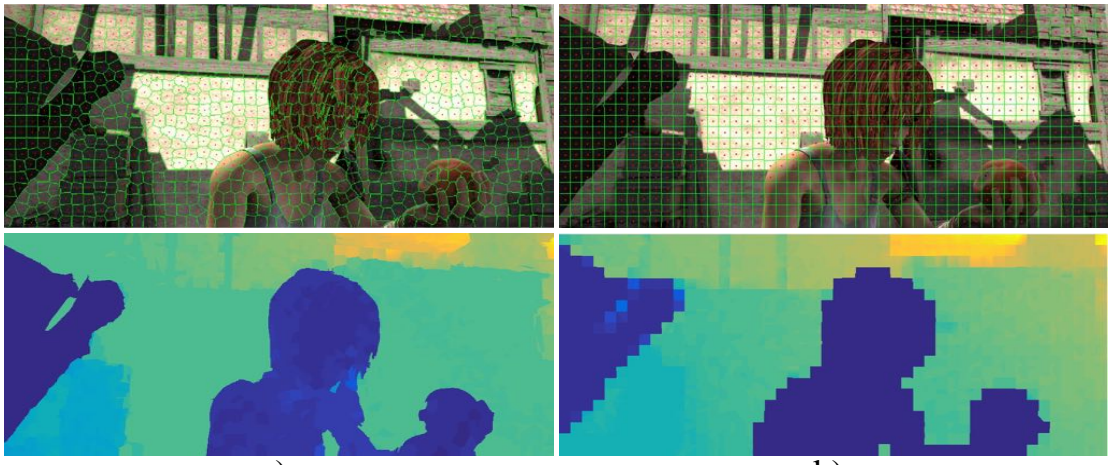
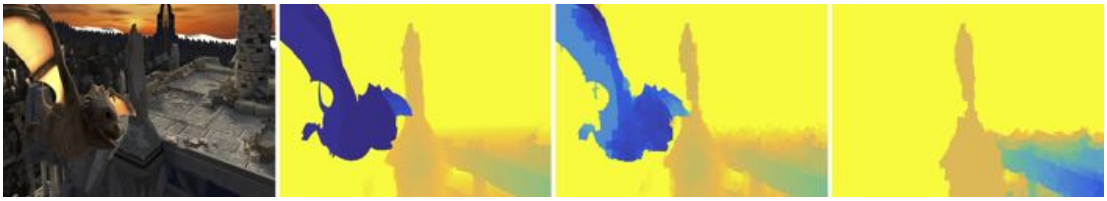


Figure 6.17: Effects of superpixel pattern on the reconstruction of a dynamic scene. a) with SLIC as superpixels (MRE for the shown frame is 0.0912) b) with uniform grid as superpixels (MRE achieved for the given frame is 0.1442).

flow methods [12] [31] to inspect the efficiency of our method. In Fig.(6.16b), we show the reconstruction performance evaluated in RMSE ¶ with different optical flow as inputs. This experiment reveals the importance of dense optical flow estimation in achieving accurate reconstruction of a dynamic scene. While ground truth optical flow naturally achieves the best performance, the difference between different state-of-the-art optical flows estimations is not dramatic. Therefore, we conclude that our method can achieve better performance with the available of dense optical flow algorithm's.

Performance with regular grid as image superpixel: Its not only the number of superpixel that affects the accuracy of reconstruction under piece-wise planar assumption but the choice of superpixel pattern. To analyze this dependency, we took the worst possible case i.e divide the reference image into approximately 1000 regular grid and compare its performance against 1000 SLIC superpixel. Our observation clearly shows the decline in performance in comparison to SLIC superpixels. However, the difference in accuracy is not very significant (see Fig.(6.17)).

¶Root mean square error RMSE $= \sqrt{\frac{1}{P}\sum_{i=1}^{P}(z_{\text{gt}}^{i} - z_{\text{e}}^{i})^2}$, where z_{e}^{i}, z_{gt}^{i} denotes the estimated and the ground-truth depth respectively and P is the total number of points



a) b) c) d)

Figure 6.18: Effect of parameter K in building the K-NN graph. Our algorithm results in good reconstruction if a suitable K is chosen, in accordance with the levels of complexity in a dynamic scene. (b) Ground-truth depth-map (scaled for illustration purpose). (c) when K=4, a reasonable reconstruction is obtained. (d) when K=20, regions tend to grow bigger. (Best viewed in color.)

Effects of K in K-NN Graph: Under our method, the ARAP energy term is evaluated within K nearest neighbors, different K's may have different effect on the resultant 3D reconstruction. We conducted an experiment to analyze the effect of varying K on the MPI Sintel dataset and the results are illustrated in Fig.(6.18). With the increase of K, the rigidity constraint is enforced in increased neighborhood, which makes the 3D reconstruction tends to be globally rigid. In most of our experiments, we used a K in the range of $15 - 20$, which achieved satisfactory reconstructions. Increasing the value of K directly affects the overall computational complexity of the algorithm.

6.6 Limitations

The success of our method depends on the effectiveness of the piece-wise planar and rigid assumption. Our method may fail if the piece-wise smooth model is no longer a valid approximation for the dynamic scene. For example, very fine or very small structures which are considerably far from the camera are difficult to recover under the piecewise planar assumption. Furthermore, our approach may also fail, when the motions of the dynamic objects in the scene between consecutive frame are significantly large such that most of its neighboring plane relations in the reference frame get violated in the next frame. Couple of examples for such situations are discussed in Fig.(6.19).

Moreover, our algorithm is computationally expensive to execute on a regular desktop machine. This is due to the higher order graph relation and particle based refinement using TRW-S. One can use different optimization algorithms such as BFGS, L-BFGS and their variants [25]. In any case, the higher order relation increases the range of interaction which makes the problem computationally expensive.

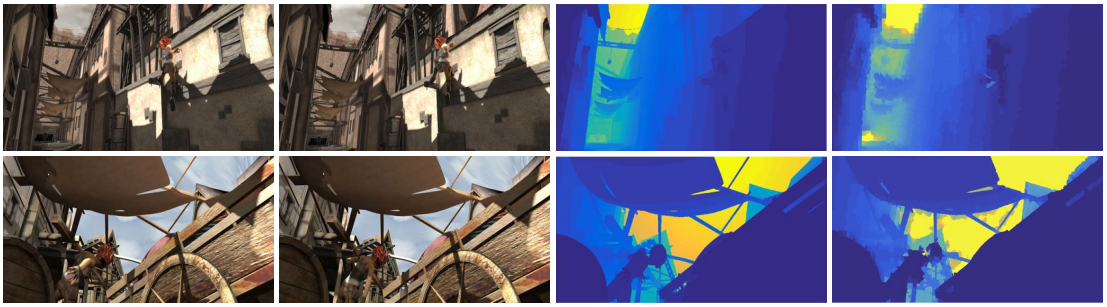


Figure 6.19: (a)-(b) are the reference frame and the next frame. It is a very challenging case for proper scale recovery with monocular images with dynamic motion. In both of these cases the motion of the girl between two consecutive frames is very large and therefore, the neighboring relations with the planes (say superpixels in image domain) in the consecutive frames gets violated. In such cases, our method may not be able to provide correct scales for each moving planes in 3D. In the first example, the complicated motion of the feet of the girl leads to wrong scale estimation. In the second example, the cart along with girl is moving w.r.t the camera. The hand of the girl has a substantial motion in the consecutive frames which leads to incorrect estimation of scale. (c)-(d) Ground-truth and obtained depth map respectively.

6.7 Closing Remarks

In this chapter we explored, investigated and supplied a distinct perspective to one of the classical problem in geometric computer vision i.e to reconstruct a dense 3D model of a *complex, dynamic, and generally non-rigid* scene from its two perspective images. This topic of research is often considered as a very challenging task in structure-from-motion. In spite of its reasonable challenges, we have demonstrated that dense-detailed 3D reconstruction of dynamic scenes is in fact possible, provided that certain prior assumptions about the scene geometry and about the deformation in the scene are satisfied. Both assumptions we stated are mild, realistic and commonly satisfied by the real-world scenario's.

Our comprehensive evaluation on benchmark datasets shows that, our new insight to solve dense monocular 3D reconstruction of a general dynamic scene provides better results than other competing methods. This said, we think a more profound research on top of our idea may help in development of sophisticated SfM algorithm's.

We believe our algorithm provides a plausible new direction to perceive a complex dynamic scene with a single monocular camera. Lastly, we want to stress on the point that rigidity is a powerful concept in SFM, and careful or acute extensions of the present approach may open up new path to advance dense 3D reconstruction from images.

7

Dense Depth Estimation of a Complex Dynamic Scene without Explicit 3D Motion Estimation

Contents

In the last chapter we describe how to estimate dense 3D reconstruction of a dynamic scene using two perspective frames. Our geometric method to address this problem using a piece-wise

rigid scene model requires a reliable estimation of motion parameters for each local model, which can be tricky to obtain and validate. In this chapter we will show that given per-pixel optical flow correspondences between two consecutive frames and the sparse depth prior for the reference frame, we can recover the dense depth map for the successive frames without solving for motion parameters. By assigning the locally rigid structure to the piece-wise planar approximation of a dynamic scene which transforms as rigid as possible over frames, we will demonstrate that we can bypass the motion estimation step. In essence, our formulation provides a new way to think and recover dense depth map of a complex dynamic scene which is recursive, incremental and motion free in nature and therefore, it can also be integrated with the modern machine learning frameworks for large-scale depth-estimation applications. Our proposed method does not make any prior assumption about the rigidity of a dynamic scene, as a result, it is applicable to a wide range of scenarios. Experimental results show that our method can effectively provide the depth for the successive/multiple frames of a dynamic scene without using any motion parameters.

7.1 Introduction

Dense depth estimation of complex dynamic scenes from two consecutive frames has recently gained enormous attention from several industries involved in augmented reality, autonomous driving, movies *etc*. Despite the recent research in solving this problem has provided some promising theory and results, its success still strongly depends on the *accurate* estimation of motion parameters.

To our knowledge, almost all the existing *geometric* solutions to this problem have tried to fit the well-established theory of rigid reconstruction to estimate per-pixel depth of *dynamic* scenes from monocular images [139, 111, 149]. Hence, these extensions are intricate to execute and highly depends on per-object or per-superpixel [3] *reliable* motion estimates [139, 111, 149]. The main issue with these frameworks is that, even if the depth for the first/reference frame is known, we must solve for per-superpixel or per-object motion to obtain the depth for the next frame. As a result, the composition of their objective function fails to utilize the depth knowledge and therefore, it does not integrate to the large-scale applications. In this work, we argue that in a dynamic scene, if the depth for the reference frame is known then it seems "unnecessary or at least undesirable" to estimate motion to recover the dense depth map for the next frame. Therefore, the rationale behind relative motion estimation as an essential paradigm for obtaining the depth of a complex dynamic scene seems *optional* under the prior knowledge about the depth of the reference frame and dense optical flow between frames. To endorse our argument, we propose a new motion free approach which is easy to implement and allow the users to get rid of the complexity associated with the optimization on $\mathbb{SE}(3)$ manifold.

Figure 7.1: Given consecutive monocular perspective frame (a), (b) of a complex dynamic scene and the dense optical flow correspondences between them (d). Also, assume an approximate sparse depth prior for the reference frame is provided as input (c), then, our algorithm under the piecewise planar approximation of a dynamic scene gives per-pixel depth estimate for the next frame (f) without solving for any motion parameters. (e) ground-truth depth.

We posit that the recent geometric methods to solve this task have been limited by their inherent dependence on the motion parameters. Consequently, we present an alternative method to realize the dynamic scene depth estimation task as a global as-rigid-as-possible (ARAP) optimization problem which is motion-free. Inspired by the prior work [111], we model the dynamic scene as a set of locally planar surface, now previous work constrains the movement of local planar structure based on the homography [131] and its relative motion between frames. In contrast, we propose that ARAP constraint over a dynamic scene may not need 3D motion parameters, and its definition just based on 3D Euclidean distance metric is a sufficient regularization to supply the depth for the next frame. To this point, one may ask "*Why ARAP assumption for a dynamic scene?*"

We want to recapitulate the intuition we developed in the last chapter. Consider a general real-world dynamic scene, the change we observe in the scene between consecutive time frame is not arbitrary, rather it is regular. Hence, if we observe a local transformation closely, it changes rigidly, but the overall transformation that the scene undergoes is non-rigid. Therefore, to assume that the dynamic scene deforms as rigid as possible seems quite convincing and

practically works well for most real-world dynamic scenes.

To use this ARAP model, we first decompose the dynamic scene as a collection of moving planes. We considered K-nearest neighbors per superpixel [3] —which is an approximation of a surfel in the projective space, to define our ARAP model. For each superpixel, we choose three points *i.e.*, an anchor point (center of the plane), and two other non-collinear points. Since the depth for the reference frame is assumed to be known (for at least 3 non-collinear points per superpixel), we can estimate per plane normal for the reference frame, but to estimate per plane normal for the next frame, we need depth for at least 3 non-collinear points per plane §7.3. If per-pixel depth for the reference frame is known, then ARAP model can be extended to pixel level without any loss of generality. The only reason for such discrete planar approximation is the computational complexity.

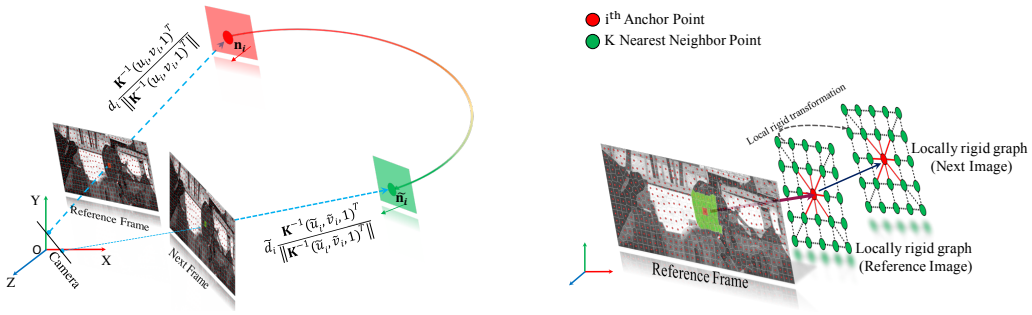
In this work, we make the following contributions:

- We provide a motion-free approach to estimate the dense depth map of a complex dynamic scene.
- Our algorithm under piece-wise planar and as rigid as possible assumption appropriately encapsulates the behavior of a dynamic scene to estimate per pixel depth.
- Although the formulation is shown to work ideally for classical case of two consecutive frames, its incremental in nature and therefore, it is easy to extend to handle multiple frames without estimating any 3D motion parameters. Experimental results over multiple frames show the validity of our claim §7.4.

7.2 Related Literature and Motivation

Recently, numerous work motivated by the success of deep learning has been published for the dense depth estimation of a dynamic scene from images [204, 69, 190, 62]. The noticeable part is, none of these work shows their results on complex dynamic scene say MPI dataset [24]. For brevity, in this chapter, we limit our discussion to the recent works that are motivated *geometrically* to solve this problem, leading to the easy discourse of our contributions. Also, we briefly discuss why our formulation can be more beneficial to the learning algorithms for this task than other geometric approaches [111, 149].

Motion-free approach to estimate the 3D geometry of a rigid scene introduced by Li [122] and its extension [94] to single non-rigidly deforming object are restricted to handle few *sparse* points over multiple frames (M view, N point). To the best of our knowledge at the time of writing this thesis, two significant class of work in the recent past have been proposed for estimating *dense* depth map of the entire dynamic scene from two consecutive monocular images [139, 111, 149], however, all of these methods are motion dependent. These work can broadly be classified as (a) object level motion segmentation approach (b) object level motion segmentation free approach.



(a) Piece-wise planar model

(b) Locally rigid k-nearest neighbor graph

Figure 7.2: (a) Piece-wise planar approximation of a dynamic scene. Each superpixel is assumed to be an approximation of a 3D plane in the projective space. The center of the plane is shown with a filled circle (anchor point). (b) Decomposition of the scene into a local graph structure. Locally rigid graph model with its k-nearest neighbor is shown for the reference frame and the next frame.

(A) OBJECT-LEVEL MOTION SEGMENTATION APPROACH: Ranftl *et al.* [149] proposed a two/three-staged approach to solve dense monocular depth estimation of a dynamic scene. Given the dense optical flow field, the method first performs an object level motion segmentation using epipolar geometry [85]. Per-object motion segmentation is then used to perform object level 3D reconstruction using triangulation [85]. To obtain a scene consistent depth map, ordering constraint and smoothness constraint were employed over Quick-shift superpixel [185] graph to deliver the final result.

(B) OBJECT-LEVEL MOTION SEGMENTATION FREE APPROACH: Kumar *et al.* [111] argued that "in a general dynamic scene setting, the task of densely segmenting rigidly moving object or parts is not trivial". They proposed an over-parametrized algorithm to solve this task without using object-specific motion segmentation. The method dubbed as "Superpixel Soup" showed that under two mild assumptions about the dynamic scene *i.e.*, (a) the deformation of the scene is locally rigid and globally as rigid as possible and (b) the scene can be approximated by piece-wise planar model, scale consistent 3D reconstruction of a dynamic scene can be obtained for both the frames with a higher accuracy. Inspired by locally rigid assumption, recently, Noraky *et al.* [139] proposed a method that uses optical flow and depth prior to estimate pose and 3D reconstruction of a deformable object.

CHALLENGES WITH SUCH GEOMETRIC APPROACHES: Although these methods provide a plausible direction to solve this challenging problem, its usage to real-world applications is very limited. The major challenge with these approaches is the correct estimation of motion parameters. The method proposed by Ranftl *et al.* [149] estimates per-object relative rigid

motion which is not a sensible choice if the object themselves are deforming. On the other hand method such as [139, 111] estimates per superpixel/region relative rigid motion which is sensitive to the size of the superpixels and distance of the surfel from the camera.

The point we are trying to make is, given the depth for the reference frame of a dynamic scene, *can we correctly estimate the depth for the next frame using the aforementioned approaches?*. Maybe yes, but then, we have to again estimate relative rigid motion for each object or superpixel and so on and so forth. Inspired by the "as-rigid-as-possible" (ARAP) intuition [111], in this work, we show that if we know the depth for the reference frame and dense optical flow correspondences between consecutive frames, then estimating relative motion is not essential, under the locally planar assumption. We can successfully estimate the depth for the next frame by exploiting as-rigid-as-possible global constraint. These depth estimate using ARAP can further be refined using boundary depth continuity constraint.

The next concern could be *why we are after solving this problem in a motion free way?*. Keeping in mind the success of deep learning approaches to estimate per-frame dense depth map, our cost function can directly provide the depth for the next frame of a dynamic scene without any motion estimate. And since the choice of a reference frame and the next frame is relative, it further provides a recursive way to improve depth estimate over iteration if supplied with appropriate priors. Moreover, our formulation provides the flexibility to solve for depth at a pixel level rather than at an object level or superpixel level which is hard to realize using motion based approaches [139, 111, 149]. Nevertheless, to reduce the overall computational cost, we stick to optimize our objective function at superpixel level.

7.3 Piecewise Planar Scene Model

Inspired by the recent work on dense depth estimation of a general dynamic scene [111], our model parameterizes the scene as a collection of piece-wise planar surface, where each local plane is assumed to be moving over frames. The global deformation of the entire scene is assumed to be as rigid as possible. Moreover, we assign the center of each plane (anchor point) to act as a representative for the entire points within that plane (see Fig.7.2). In addition to the anchor point of each plane, we take two more points from the same plane so that these three points are non-collinear (see Fig.7.3). This strategy is used to define our as rigid as possible constraint between the reference frame and next frame without using any motion parameters. As the depth for the reference frame and the optical flow between the two successive frames is assumed to be known a priori, each local planar region is described using only four parameters —normal and depth, instead of nine [111].

Our model first assigns each pixel of the reference frame to a superpixel using SLIC algorithm [3] and each of these superpixels then acts as a representative for its 3D plane geometry. Since the global geometry of the dynamic scene is assumed to be deforming ARAP, we solve

for the depth in the next frame subject to the transformation that each plane undergoes from the first frame to the next frame should be as minimum as possible. The solution to ARAP global constraint provides depth for three points per plane in the next frame, which is used to estimate the normal and depth of the plane. The estimated depth and normal of each plane is then used to calculate per pixel depth in the next frame.

Although our algorithm is described for the classical two-frame case, it is easy to extend to the multi-frame case. The energy function we define below is solved in two steps: First, we solve for the depth of each superpixel in the next frame using as rigid as possible constraint. Due to the piece-wise planar approximation of the scene, the overall solution to the depth introduces discontinuity along the boundaries. To remove the blocky artifacts —due to the discretization of the scene, we smooth the obtained depth along the boundaries of all the estimated 3D plane in the second step using TRWS [97]. If the ARAP cost function is extended to pixel-level then the boundary continuity constraint can be avoided [90]. Nevertheless, over-segmentation of the scene provides a good enough approximation of a dynamic scene and is computationally easy to handle.

7.3.1 Model overview

Notation: We refer two consecutive perspective image I, I′ as the reference frame and next frame respectively. Vectors are represented by bold lowercase letters, for e.g., 'x' and the matrices are represented by bold uppercase letters, for e.g., 'X'. The 1-norm, 2-norm of a vector is denoted as $|.|_1$ and $\|.\|_2$ respectively.

7.3.2 As-Rigid-As-Possible (ARAP)

The idea of ARAP constraint is well known in practice and has been widely used for shape modeling and shape manipulation [92]. Recently Kumar *et al.* [111] exploited this idea to estimate scale consistent dense 3D structure of a dynamic scene. The motivation to use ARAP constraint in our work is inspired by [111] idea *i.e.* restrict the deformation such that the overall transformation in the scene between frames is as small as possible.

Let (d_i, d_j) and $(\tilde{d}_i, \tilde{d}_j)$ be the depth of two neighboring 3D points i, j from the reference coordinate in the consecutive frames. Let $(u_i, v_i, 1)^T$, $(u_j, v_j, 1)^T$ be its image coordinate in the reference frame and $(\tilde{u}_i, \tilde{v}_i, 1)^T$, $(\tilde{u}_j, \tilde{v}_j, 1)^T$ be its image coordinate in the next frame. If 'K' denotes the intrinsic camera calibration matrix then, $e_i = \mathrm{K}^{-1}(u_i, v_i, 1)^T / \|\mathrm{K}^{-1}(u_i, v_i, 1)^T\|_2$, $e_j = \mathrm{K}^{-1}(u_j, v_j, 1)^T / \|\mathrm{K}^{-1}(u_j, v_j, 1)^T\|_2$ is the unit vector in the direction of the i^{th}, j^{th} 3D point respectively for the reference frame. Similarly, the corresponding unit vectors in the next frame is denoted with $\tilde{e}_i, \tilde{e}_j$ (see Fig. 7.2a). Using these notations, we define the ARAP con-

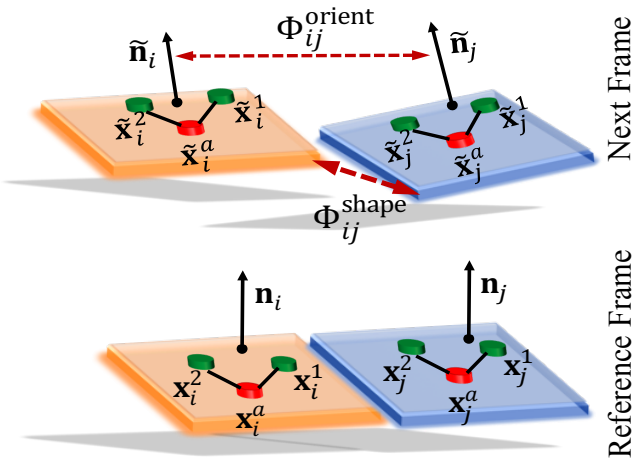


Figure 7.3: Intuition on orientation and shape regularization. Anchor point and two non-collinear points are shown in red and green respectively. Dark red line show the change in the next frame.

straint as:

$$\Phi^{\text{arap}} = \sum_{i=1}^{3N} \sum_{j \in \mathcal{N}_i^k} w_{ij}^{(1)} \Big| \underbrace{\|d_i e_i - d_j e_j\|_2}_{\text{reference frame}} - \underbrace{\|\tilde{d}_i \tilde{e}_i - \tilde{d}_j \tilde{e}_j\|_2}_{\text{next frame}} \Big|_1 \tag{7.1}$$

Here, N is the total number of planes used to approximate the scene and $\mathcal{N}_i^k$ is the 'k' neighboring planes local to i^{th} superpixel (see Fig. 7.2b). $w_{ij}^{(1)}$ is the exponential weight fall off based on the image distance of the points *i.e.*slowly break the rigidity constraint if the points are far apart in the image space. This constraint encapsulates our idea *i.e.*, the change in the distance of i^{th} point relative to its local neighbors in the next frame should be as minimum as possible. Note that the summation goes over $3N$ rather than N due the reason discussed in Sec. §7.1

7.3.3 Orientation and Shape Regularization

Solving the ARAP constraint provides us the depths for three non-collinear points per-plane for the next frame. We use these three depth estimate per plane to solve for their normals in the next frame. Let the 3D points corresponding to the three depths for i^{th} superpixel in the next frame be denoted as $\tilde{\mathrm{x}}_i^a$, $\tilde{\mathrm{x}}_i^1$ and $\tilde{\mathrm{x}}_i^2$ respectively. We estimate the normals in the next frame as:

$$\tilde{\mathrm{n}}_i = \frac{(\tilde{\mathrm{x}}_i^a - \tilde{\mathrm{x}}_i^1) \times (\tilde{\mathrm{x}}_i^a - \tilde{\mathrm{x}}_i^2)}{\|(\tilde{\mathrm{x}}_i^a - \tilde{\mathrm{x}}_i^1) \times (\tilde{\mathrm{x}}_i^a - \tilde{\mathrm{x}}_i^2)\|_2}, \tag{7.2}$$

where superscript 'a' is used intentionally to denote the anchor point, which is assumed to be at the center of each plane (see Fig. 7.3). Rewriting Eq. (7.2) in terms of depth

$$\tilde{\mathrm{n}}_i = \frac{(\tilde{d}_i^a \tilde{e}_i^a - \tilde{d}_i^1 \tilde{e}_i^1) \times (\tilde{d}_i^a \tilde{e}_i^a - \tilde{d}_i^2 \tilde{e}_i^2)}{\|(\tilde{d}_i^a \tilde{e}_i^a - \tilde{d}_i^1 \tilde{e}_i^1) \times (\tilde{d}_i^a \tilde{e}_i^a - \tilde{d}_i^2 \tilde{e}_i^2)\|_2}. \tag{7.3}$$

(A) ORIENTATION SMOOTHNESS CONSTRAINT: Once we compute the normal for each plane and 3D coordinates of the anchor point, which lies on the plane, we estimate the depth of the plane as follows

$$\tilde{\mathrm{n}}_i^T \tilde{\mathrm{x}}_i^a = \tilde{d}_i^{p_a}. \tag{7.4}$$

The computed depth of the plane is then used to solve for per-pixel depth in the next frame —assuming the intrinsic camera matrix is known [111, 85]. To encourage the smoothness in the change of angles between each adjacent planes (see Fig. 7.3), we define the orientation regularization as

$$\Phi_{ij}^{\text{orient}} = \lambda_1 \varrho_1 \Big(1 - \frac{|\tilde{\mathrm{n}}_i^T \tilde{\mathrm{n}}_j|}{\|\tilde{\mathrm{n}}_i\| \|\tilde{\mathrm{n}}_j\|} \Big), \tag{7.5}$$

where, λ_1 is an empirical constant and $\varrho_1(x) = \min(|x|, \sigma_1)$ is the truncated l_1 function with σ_1 as a scalar parameter.

(B) SHAPE SMOOTHNESS CONSTRAINT: In our representation, the dynamic scene model is approximated by the collection of piecewise planar regions. Hence, the solution to per-pixel depth obtained using Eq. (7.1) to Eq. (7.4) may provide discontinuity along the boundaries of the planes in 3D (see Fig. 7.3). To allow smoothness in the 3D coordinates for each adjacent planes along their region of separation, we define the shape smoothness constraint as

$$\Phi^{\text{shape}} = \sum_{(i,j) \in N_b} w_{ij}^{(2)} \varrho_2 (\underbrace{\|d_i e_i - d_j e_j\|_2^2}_{\text{reference frame}} + \underbrace{\|\tilde{d}_i \tilde{e}_i - \tilde{d}_j \tilde{e}_j\|_2^2}_{\text{next frame}}). \tag{7.6}$$

The symbol 'N_b' denotes the set of boundary pixels of i^{th} superpixel that are shared with the boundary pixel of other superpixels. The weight $w_{ij}^{(2)} = \exp(-\beta \|\mathrm{I}_i - \mathrm{I}_j\|_2)$ takes into account the color consistency of the plane along the boundary points —weak continuity constraint [18]. Since all the pixels within the same plane are assumed to share the same model, smoothness for the pixels within the plane is not essentially required. Similar to orientation regularization, $\varrho_2(x) = \min(|x|, \sigma_2)$ is the truncated l_1 penalty function with σ_2 as a scalar parameter. The overall optimization steps of our method is provided in Algorithm (5).

Algorithm 5 : A Motion Free Approach

Input: (I, I′), optical_flow(I, I′), K, depth for reference frame.
Output: Dense depth map for the next frame.
1: Over-segment the reference frame into N superpixels [3].
2: Assign anchor point for each superpixel and two other points in the same plane such that these three points are non-collinear (see Fig. 7.3).
3: Use K-NN algorithm over superpixels to get the K-nearest neighbor index set.
4: Solve for per-superpixel depth in the next frame §7.3.2

$$\begin{aligned} &\Phi^{\text{arap}} \rightarrow \underset{\tilde{d}_i}{\text{minimize}} \\ &\text{subject to: } \tilde{d}_i > 0, \quad |\tilde{d}_i - d_i| < di\sigma \\ &\text{where, } di\sigma \text{ is the variance in the depth.} \end{aligned} \tag{7.7}$$

Note: The second constraint provides a trust region for the fast and proper convergence of a non-convex problem (Fig.7.10). Can be thought of as max/min restriction to the scene deformation.

5: Estimate the normal of each plane in the next frame Eq. (7.3).
6: Estimate the depth of each plane Eq. (7.4).
7: Solve per pixel depth for the next frame using per plane depth ($\tilde{d}_i^{p_a}$), K, normal of the plane and its image coordinate.
8: Refine the depth of the next frame by minimizing Eq. (7.5), Eq. (7.6) with respect to depth and normal [97] §7.3.3.

$$\begin{aligned} &(\Phi^{\text{orient}} + \Phi^{\text{shape}}) \rightarrow \underset{\tilde{d}_i, \tilde{\text{n}}_i}{\text{minimize}} \\ &\text{subject to: } \tilde{d}_i > 0, \|\tilde{\text{n}}_i\| = 1. \end{aligned} \tag{7.8}$$

9: (Optional) For generalizing the idea to multi-frame, repeat the above steps by making the next frame as the reference frame and new frame as the next frame.

7.4 Experimental Evaluation

We performed the experimental evaluation of our approach on two benchmark datasets, namely MPI Sintel [24] and KITTI [68]. These two datasets conveniently provide a complex and realistic environment to test and compare our dense depth estimation algorithm. We compared the accuracy of our approach against two recent state-of-the-art methods [111, 149] that use geometric approach to solve dynamic scene dense depth estimation from monocular images. These comparisons are performed using three different dense optical flow estimation algorithms, namely PWC-Net [165], FlowFields [12] and Full Flow [31]. All the depth estima-

tion accuracies are reported using mean relative error (MRE) metric. Let $\tilde{d}$ be the estimated depth and $\tilde{d}^{gt}$ be the ground-truth depth, then MRE is defined as

$$\text{MRE} = \frac{1}{P}\sum_{i=1}^{P}\frac{|\tilde{d}_i - \tilde{d}_i^{gt}|}{\tilde{d}_i^{gt}}, \tag{7.9}$$

where 'P' denotes the total number of points. The statistical results for DMDE [149] and Superpixel Soup [111] are taken from their published work for comparison.

Implementation Details: We over-segment the reference frame into 1000-1200 superpixels using SLIC algorithm [3] to approximate the scene. Almost all of the experiments use fixed value of $di\sigma = 1$ and $\mathcal{N}_i^k = 20$-25. For computing the dense optical flow correspondences between images we used both traditional methods and deep-learning framework such as PWC-Net [165], FlowFields[12] and Full Flow [31]. The depth for the reference image is initialized using Mono-Depth [69] model on the KITTI dataset and using Superpixel Soup algorithm [111] on the MPI-Sintel dataset. The reason for such inconsistent choice is that available deep-neural network depth estimation model fails to provide reasonable depth estimate on the MPI dataset –see supplementary material. The proposed optimization is solved in two stages, firstly Eq. (7.7) is optimized using SQP [146] algorithm and Eq. (7.8) is optimized using TRW-S [97] algorithm. The choice of the optimizer is purely empirical, and the user may choose different optimization algorithm to solve the same cost function. The algorithm is implemented in C++/MATLAB which takes 10-12 minutes on a commodity desktop computer to provides the results.

The implementation is performed under two different experimental settings. In the first setting, given the sparse (*i.e.* for three non-collinear points per superpixel) depth estimate of a dynamic scene for the reference frame, we estimate the per-pixel depth for the next frame. In the second experimental setting, we generalize this idea of two frame depth estimation to multiple frames by computing the depth estimates over frames. For easy understanding, MATLAB codes are provided in the Appendix (E) showing our idea of ARAP on synthetic examples of a dynamic scene.

MPI Sintel: This dataset gives an ideal setting to evaluate depth estimation algorithms for complex dynamic scenes. It contains image sequences with considerable motion and severe illumination change. Moreover, the large number of non-planar scenes and non-rigid deformations makes it a suitable choice to test the piece-wise planar assumption. We selected seven set of scenes namely *alley*_1, *alley*_2, *ambush*_5, *bandage*_1, *bandage*_2, *market*_2 and *temple*_2 from the clean category of this dataset to test our method.

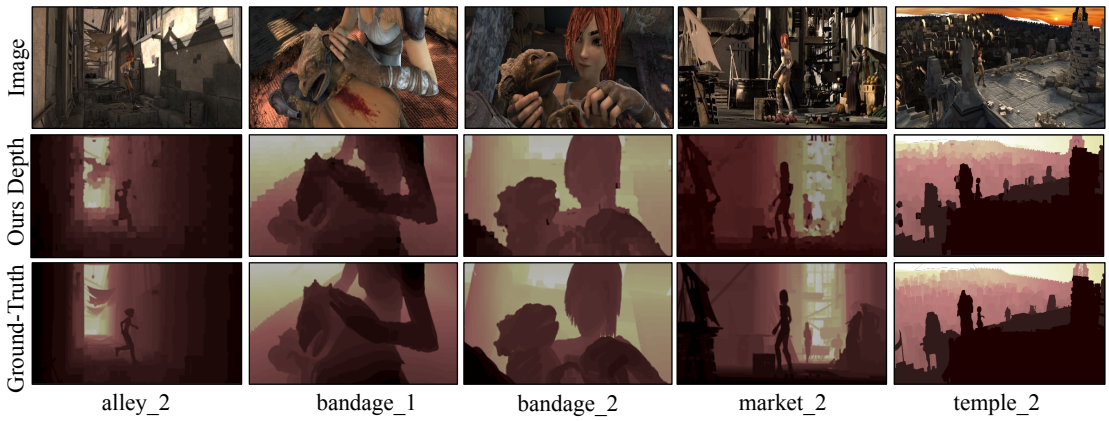


Figure 7.4: Depth results on the MPI Sintel dataset[24] for the next frame under two frame experimental setting. 2nd and 3rd row show ours and ground-truth depth map results respectively.

OF↓ / Methods →	DMDE [149]	S. Soup [111]	Ours
PWC Net [165]	-	-	0.1848
Flow Fields [12]	0.2970	0.1669	0.1943
Full Flow [31]	-	0.1933	0.2144

Table 7.1: Comparison of dense depth estimation methods under *two consecutive frame setting* against the state-of-the-art approaches on the MPI Sintel dataset [24]. For consistency, the evaluations are performed using mean relative error metric (MRE).

(a) Two-frame results: While testing our algorithm for the two-frame case, the reference frame depth is initialized using recently proposed superpixel-soup algorithm [111]. The optical flow between the frames is computed using methods such as PWC-Net [165], Flow Fields [12] and Full Flow [31]. Table (7.1) shows the statistical performance comparison of our method against other geometric approaches. The statistics clearly show that we can perform almost equally well without motion estimation. Qualitative results within this setting are shown in Fig.(7.4).

(b) Multi-frame results: In multi-frame setting, only the depth for the first frame is initialized. The result obtained for the next frame is then used for the upcoming frames to estimate its dense depth map. Since we are dealing with the dynamic scene, the environment changes slowly and therefore, the error starts to accumulate over frames. Fig.(7.9a) reflects this propagation of error over frames. Qualitative results over multiple frames are shown in Fig.(7.5).

(a) Image (MPI Dataset) (b) Dense Depth Estimate over frames

Figure 7.5: Results on MPI Sintel dataset [24] under multi-frame experimental setting. (a) Image frame for which the depth is initialized. (b) Depth estimation results using our method over frames.

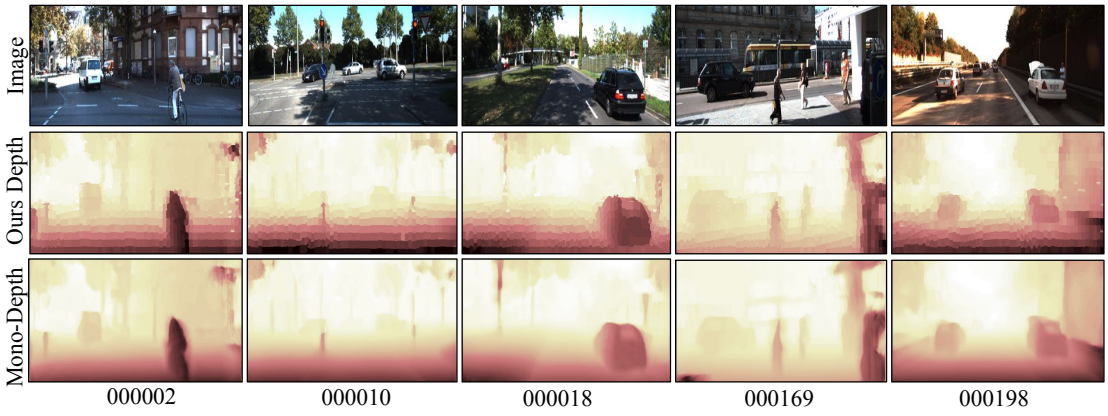
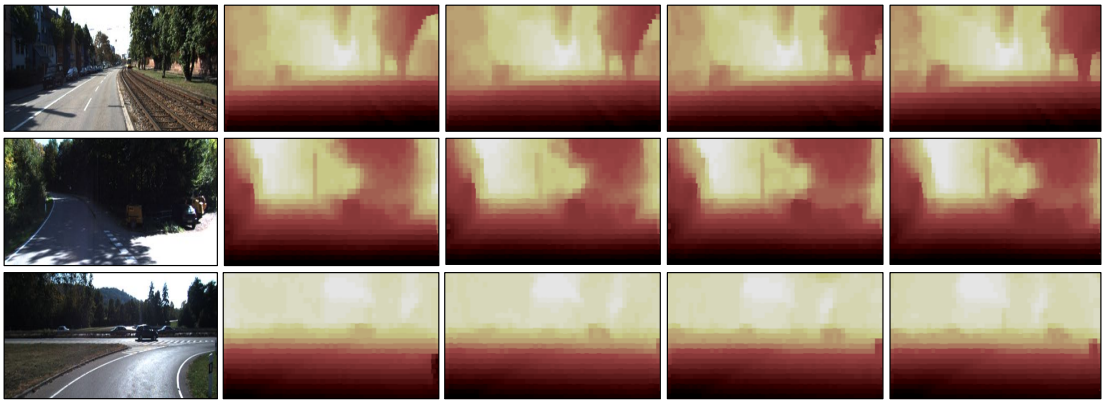


Figure 7.6: Results on KITTI 2015 benchmark dataset under two frame experimental setting. 3rd row: Monodepth [69] results on the same sequence for the next frame for qualitative comparison.



(a) Image (KITTI) (b) Dense Depth Estimate over frames

Figure 7.7: Results on KITTI raw dataset under multi-frame experimental setup. (a) Reference image for which the depth is initialized (b) Dense depth results over frames using our algorithm.

KITTI: The KITTI dataset has emerged as a standard benchmark dataset to evaluate the performance of dense depth estimation algorithms. It contains images of outdoor driving scenes with different lighting conditions and large camera motion. We tested our algorithm on both KITTI raw-data and KITTI 2015 benchmark. For KITTI dataset, we used Monodepth method [69] to initialize the reference frame depth. Dense optical flow correspondences are obtained using the same aforementioned methods. For consistency, the depth estimation error measurement on KITTI dataset follows the same order of 50 meters as presented in [69] work.

(a)Two-frame results: KITTI 2015 scene flow dataset provides two consecutive frame pair of a dynamic scene to test algorithms. Table (7.2) provides the depth estimation statistical result of our algorithm in comparison to other competing methods. Here, our results are a bit better using PWC-Net [165] optical flow and Monodepth [69] depth initialization. Fig.(7.6) shows the qualitative results using our approach in comparison to the Monodepth [69] for the next frame.

(b)Multi-frame results: To test the performance of our algorithm on multi-frame KITTI dataset, we used KITTI raw dataset specifically from the city, residential and road category. The depth for only the first frame is initialized using monodepth deep learned model and then we estimate the depth for the upcoming frames. Due to very large displacement in the scene per frame (>150) pixels, the rate of change of error accumulation curve for KITTI dataset (Fig. 7.9b) is a bit steeper than MPI Sintel. Fig.(7.7) and Fig.(7.9b) show the qualitative results and depth error accumulation over frames on KITTI raw dataset respectively.

OF↓ / Methods →	DMDE [149]	S. Soup [111]	Ours
PWC Net [165]	-	-	0.1182
Flow Fields [12]	0.1460	0.1268	0.1372
Full Flow [31]	-	0.1437	0.1665

Table 7.2: Comparison of dense depth estimation under *two consecutive frame setting* against the state-of-the-art approaches on KITTI dataset [24]. For consistency, the evaluations are performed using mean relative error metric (MRE). The results are better due to monodepth initialization for the reference frame.

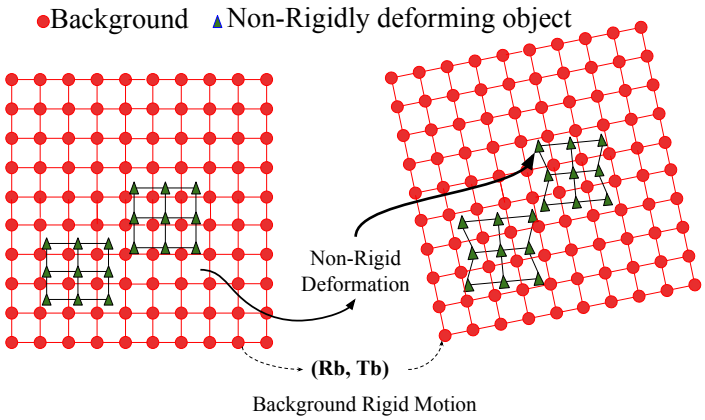


Figure 7.8: Synthetic example to conduct in-depth behavior analysis of the ARAP. Two objects are deforming independently over a rigid background motion. The objects are at a finite separation from the background. For numerical details on this example, kindly go through the Appendix(E).

Besides experimental evaluations under the aforementioned variable initialization, we also conducted other experiments to better understand the behavior of the proposed method. We conducted experiments on a synthetic example shown in Fig.(7.8) for easy understanding to the readers. MATLAB codes are provided in the Appendix (E) for reference.

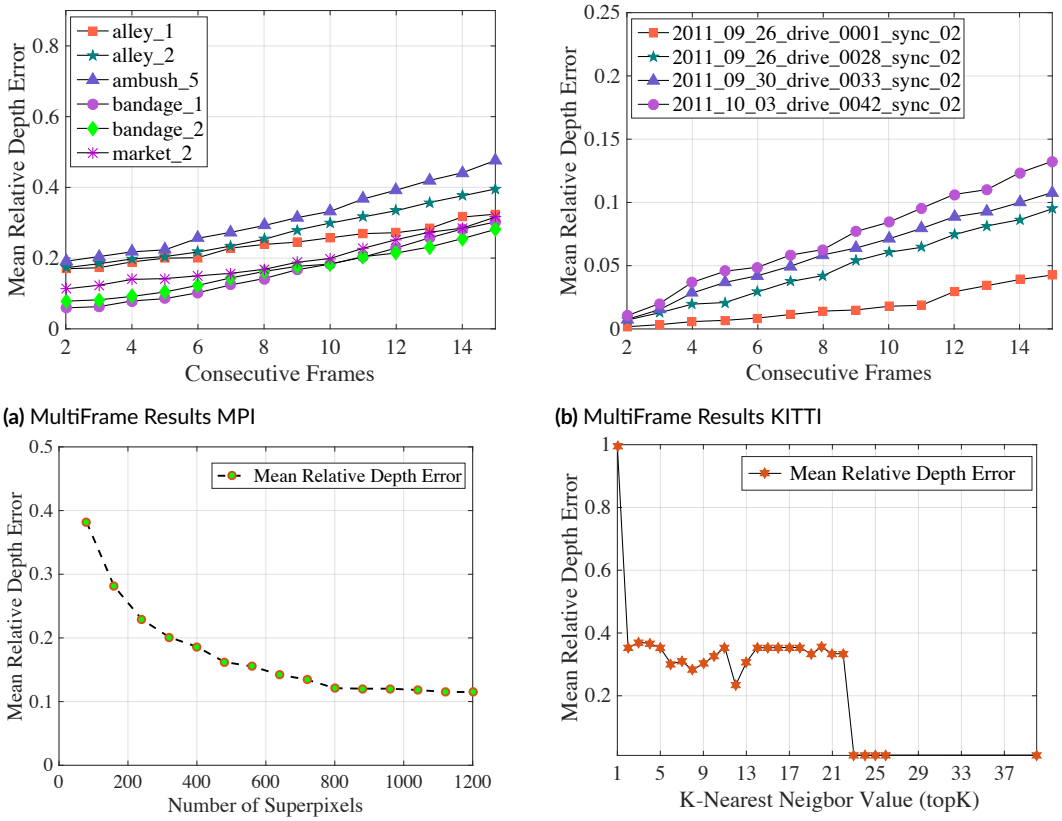
(a) Effect of the variable N: The number of superpixels to approximate the dynamic scene can affect the performance of our method. A small number of superpixel can provide poor depth result, whereas a very large number of superpixel can increase the computation time. Fig.(7.9c) shows the change in the accuracy of depth estimation with respect to change in the number of superpixels. The curve suggests that for KITTI and MPI Sintel 1000-1200 superpixel provides a reasonable approximation to the dynamic scenes.

(b) Effect of the variable $\mathcal{N}_i^k$: The number of K-nearest neighbors to define the local rigidity graph can have a direct effect on the performance of the algorithm. Although $\mathcal{N}_i^k = 20 - 25$ works well for the tested benchmarks, it is purely an empirical parameter and can be different for a distinct dynamic scene. Fig.(7.9d) demonstrates the performance of the algorithm with the change in the values of $\mathcal{N}_i^k$.

(c) Performance of the algorithm under noisy initialization: This experiment is conducted to study the sensitivity of the method to noisy depth initialization. Fig.(7.10a) shows the change in the 3D reconstruction accuracy with the variation in the level of noise from 1% to 9%. We introduced the Gaussian noise using randn() MATLAB function and the results are documented for the example shown in Fig.(7.8) after repeating the experiment for 10 times and taking its average values. We observe that our algorithm can provide arguable results when the noise level gets high.

(d) Performance of the algorithm under restricted isometry constraint with Φ^{arap} objective function: While minimizing the ARAP objective function under the $|\tilde{d}_i - d_i| < di\sigma$ constraint, we restrict the convergence trust region of the optimization. This constraint makes the algorithm works extremely well —both in terms of timing and accuracy, if an approximate knowledge about the deformation that the scene may undergo is known a priori. Fig. 7.10b shows the 3D reconstruction accuracy as a function of $di\sigma$ for the example shown in Fig.(7.8). Clearly, if we have an approximate knowledge about the scene transformation, we can get high accuracy in less time. See Fig.(7.10d) which illustrates the quick convergence by using this constraint under suitable range of $di\sigma$.

(e) Nature of convergence of the proposed ARAP optimization:
1) *Without restricted isometry constraint*: As rigid as possible minimization Φ^{arap} under the constraint $\tilde{d}_i > 0$ is alone a good enough constraint to provide acceptable results. However, it may take a considerable number of iterations to do so. Fig.(7.10c) shows the convergence curve.



(a) MultiFrame Results MPI

(b) MultiFrame Results KITTI

(c) Results with change in number of Superpixel

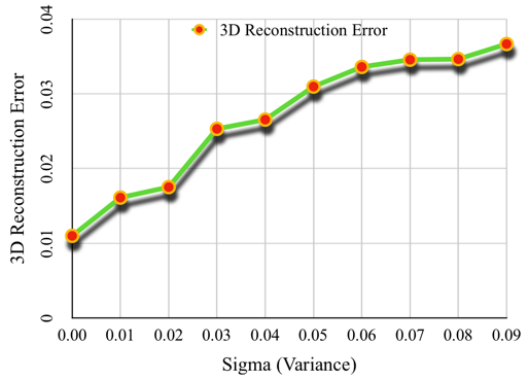
(d) Results with change in number of local neighbors

Figure 7.9: (a)-(b) Accumulation of error over frames for MPI and KITTI dataset respectively. (c) Change in the depth estimation accuracy w.r.t number of superpixel. (d) Variation in the depth accuracy as a function of k-nearest neighbor ($\mathcal{N}_i^k$)

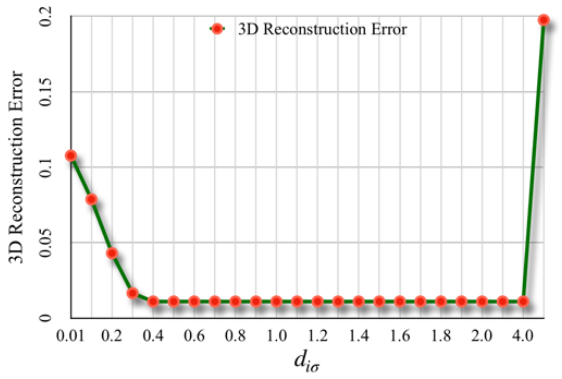
2) *With restricted isometry constraint*: Employing the approximate bound on the deformation that the scene may undergo in the next time instance can help fast convergence with similar accuracy. Fig.(7.10d) shows that the same accuracy can be achieved in 60-70 iterations.

7.6 Limitation and Discussion

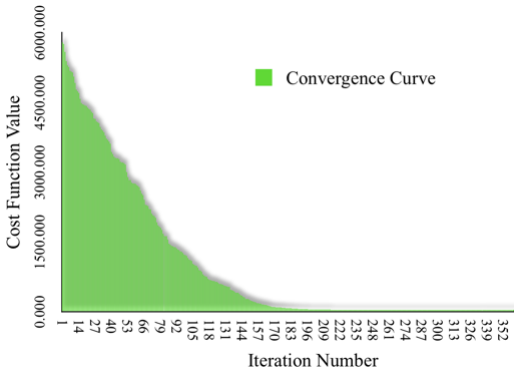
Even though our method works well for diverse dynamic scenes, there are still a few challenges associated with the formulation. Firstly, very noisy depth initialization for the reference frame can provide unsettling results. Secondly, our method is challenged by the instant arrival or removal of the dynamic subjects in the scene, and in such cases, it may need reinitialization of the reference depth. Lastly, well-known limitations such as occlusion and temporal



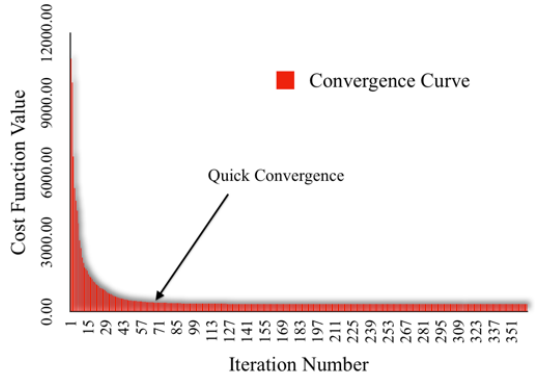
(a) 3D reconstruction error with change in noisy depth



(b) 3D reconstruction Error variation with $di\sigma$



(c) Convergence without prior knowledge on depth change



(d) Convergence with prior knowledge on depth change

Figure 7.10: (a) Depth results for the next frame with different levels of Gaussian noise in the reference frame coordinate initialization. (b) Variation in the performance with the change in the $di\sigma$ values for synthetic example. (c) Convergence curve of the ARAP objective function. (d) Quick convergence with similar accuracy on the same example can be achieved by using restricted isometric constraint.

consistency, especially around the regions close to the boundary of the images can also affect the accuracy of our algorithm.

Discussion: In defense, we would like to state that motion based methods to structure from motion is prone to noisy data as well. Algorithms like motion averaging [77], M-estimators and random sampling [171] are quite often used to rectify the solution.
(a) *Why do we choose geometric approach to initialize our algorithm on MPI dataset?* LKVO network [190] is one of the top performing networks for dense depth estimation on KITTI dataset. Our implementation of this network on the MPI dataset provided us with unsatisfactory results. Qualitative results obtained using this network on the clean class is provided in the supplementary material. The training parameters are also provided for reference.
(b) *What do we gain or lose by our motion free approach?*
Estimating all kinds of conceivable motion in a complex dynamic scene from images is a challenging task, in that respect, our method provides an alternative way to achieve per pixel depth without estimating any 3D motion. However, in achieving this we are allowing the gauge freedom between the frames (temporal relations in 3D over frames).

7.7 Closing Remark

The problem of estimating per-pixel depth of a dynamic scene, where the complex motions are prevalent is a challenging task to solve. Quite naturally, previous methods rely on standard motion estimation techniques to solve this problem, which in fact is a non-trivial task for a non-rigid scene. In contrast, this chapter introduces a new way to perceive this problem, which essentially trivializes the motion estimate as a compulsory step. By observing the behavior of most of the real-world dynamic scenes closely, it can be inferred that it locally transforms rigidly and globally as rigid as possible. Such observation allows us to propose a motion-free algorithm to dense depth estimation under the piece-wise planar approximation of the scene.

Although the proposed approach has some limitations, we believe our motion free approach provides a promising direction to explore for the future work in this field. We believe our idea can significantly benefit the deep-learning based methods in the areas of structure from motion and visual SLAM.

A

Mathematical derivation and discussion related to chapter 2

In this appendix, we first provide mathematical derivation to the sub-problems proposed in the paper. Also, we provide few qualitative comparison of our method in comparison to Dai *et al.* approach [44] for reference followed by some general discussions.

A.1 Mathematical Derivations

The augmented form of the optimization is as follows:

$$\begin{aligned}\mathcal{L}_{\varrho}(S^{\sharp}, S) = \mu\|S^{\sharp}\|_{\Theta,*} + \frac{1}{2}\|W - RS\|_F^2 + \frac{\varrho}{2}\|S^{\sharp} - g(S)\|_F^2 + \\ < Y, S^{\sharp} - g(S) >\end{aligned} \tag{A.1}$$

(a) Solution to S: Minimization the Eq:(A.1) w.r.t 'S' gives the following form

$$\begin{aligned}\underset{S}{\operatorname{argmin}}\ \mathcal{L}_{\varrho}(S) = \frac{1}{2}\|W - RS\|_F^2 + \frac{\varrho}{2}\|g^{-1}(S^{\sharp}) - S\|_F^2 + \\ < g^{-1}(Y), g^{-1}(S^{\sharp}) - S > \\ \equiv \underset{S}{\operatorname{argmin}} \frac{1}{2}\|W - RS\|_F^2 + \frac{\varrho}{2}\|S - \left(g^{-1}(S^{\sharp}) + \frac{g^{-1}(Y)}{\varrho}\right)\|_F^2\end{aligned} \tag{A.2}$$

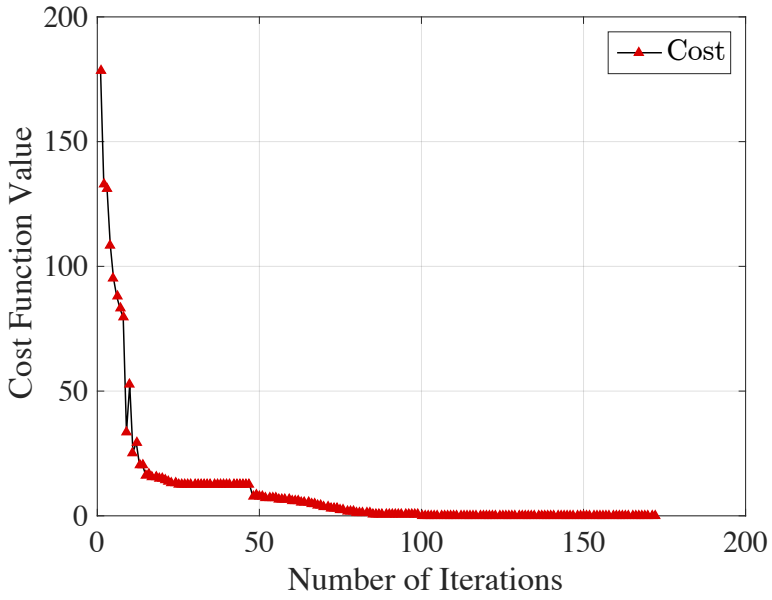


Figure A.1: Convergence Curve

Taking the derivative of Eq:(A.2) w.r.t S and equating it to zero gives

$$(\varrho I + R^T R)S = \varrho\left(g^{-1}(S^\sharp) + \frac{g^{-1}(Y)}{\varrho}\right) + R^T W \tag{A.3}$$

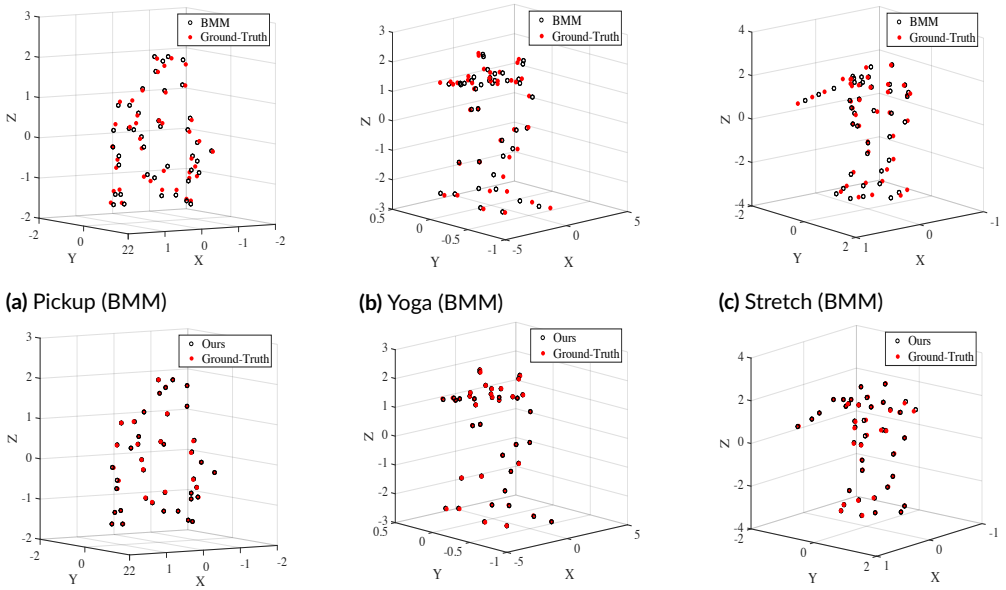
(b) Solution to $S^\sharp$: Minimization the Eq:(A.1) w.r.t '$S^\sharp$' gives the following form:

$$\begin{aligned} &\equiv \underset{S^\sharp}{\operatorname{argmin}}\, \mu\|S^\sharp\|_{\Theta,*} + \frac{\varrho}{2}\|S^\sharp - g(S)\|_F^2 + < Y, S^\sharp - g(S) > \\ &\equiv \underset{S^\sharp}{\operatorname{argmin}}\, \mu\|S^\sharp\|_{\Theta,*} + \frac{\varrho}{2}\|S^\sharp - \left(g(S) - \frac{Y}{\varrho}\right)\|_F^2 \end{aligned} \tag{A.4}$$

The Eq:(A.4) is solved by using the thresholding operator $\mathcal{S}[\tau](\sigma) = \text{sign}(\sigma).\max(|\sigma| - \tau, 0)$. Let $[U, \Sigma, V]$ be the singular value decomposition of $(g(S) - Y/\varrho)$ then the solution to $S^\sharp$ is given by $S^\sharp = U\mathcal{S}[\Theta\mu/\varrho](\Sigma)V$, with Θ as the weight assigned to singular values.

A.2 Convergence Curve

Figure A.1 show the convergence curve of our proposed optimization for solving non-rigid shape matrix.



(a) Pickup (BMM) **(b)** Yoga (BMM) **(c)** Stretch (BMM)

(d) Pickup (Ours) **(e)** Yoga (Ours) **(f)** Stretch (Ours)

Figure A.2: Qualitative comparison of our algorithm with the classical baseline BMM [44] under the same model complexity value (K). The first row and the second row shows the 3D reconstruction using Dai *et al.*and our approach respectively on the benchmark dataset.(Best viewed in color)

A.3 Qualitative Comparison

At last, we provide the visual comparison of our algorithm in comparison to the targeted baseline [44] in Figure A.2. The results clearly shows that by simple yet powerful rectification to simple prior free idea, we can achieve a significant boost in the reconstruction quality*.

Note: The term <<regularity>> in the section(2) paragraph "plausible rectification" to the solution of rotation, in the main paper, is used in a loose sense. Kindly, ignore this if it's not mathematically precise to use it to convey the intuition.

Q. Why the assumption of <<smooth>> deformation of an object over frames is reasonable in solving NRSfM?
In many real world scenario's the transition of a non-rigidly moving object from one state to another over frames is not arbitrary but is well ordered or regular in terms of rigidity. Such assumption successfully captures the general notion about the global behavior of a deforming

*Our claims are easy to verify and test using Dai *et al.* [44] publicly available code at http://users.cecs.anu.edu.au/ yuchao/publication.htm

surface, at the same time maintains the local attribute of the surface. Therefore, to assume smooth motion is a reasonable choice and works well for most non-rigidly moving object [148].

B

Mathematical derivation related to chapter 3

B.1 Solution to each unknown variables

In this appendix, we provide a detailed derivation for each of the sub-problems introduce in chapter (3)[108]. Recall that the Augmented Lagrangian formulation of our optimization problem which is defined as:

$$\begin{aligned}\mathcal{L}(S, S^\sharp, C_1, C_2, E_1, E_2, J, \{Y_i\}_{i=1}^8) = \frac{1}{2}\|W - RS\|_F^2 + \lambda_1\|E_1\|_1 + \gamma_1\|E_1\|_F^2 + \lambda_2\|J\|_* + \\ \lambda_3\|E_2\|_1 + \gamma_3\|E_2\|_F^2 + < Y_1, S^\sharp - g(S) > + \frac{\beta}{2}\|S^\sharp - g(S)\|_F^2 + < Y_2, S - SC_1 > + \\ \frac{\beta}{2}\|S - SC_1\|_F^2 + < Y_3, S^\sharp - S^\sharp C_2 > + \frac{\beta}{2}\|S^\sharp - S^\sharp C_2\|_F^2 + < Y_4, 1^T C_1 - 1^T > + \\ \frac{\beta}{2}\|1^T C_1 - 1^T\|_F^2 + < Y_5, 1^T C_2 - 1^T > + \frac{\beta}{2}\|1^T C_2 - 1^T\|_F^2 + < Y_6, C_1 - E_1 > + \\ \frac{\beta}{2}\|C_1 - E_1\|_F^2 + < Y_7, C_2 - E_2 > + \frac{\beta}{2}\|C_2 - E_2\|_F^2 + < Y_8, S^\sharp - J > + \frac{\beta}{2}\|S^\sharp - J\|_F^2.\end{aligned} \tag{B.1}$$

The ADMM works by alternatively updating each variable one at a time while assuming the remaining variables as constant.

B.1.1 Solution for S

$$S = \underset{S}{\operatorname{argmin}} \frac{1}{2}\|W - RS\|_F^2 + < Y_1, S^\sharp - g(S) > + \frac{\beta}{2}\|S^\sharp - g(S)\|_F^2 + < Y_2, S - SC_1 > + \frac{\beta}{2}\|S - SC_1\|_F^2.$$

We are minimizing this equation w.r.t S. Therefore, we convert the second and third term in the above equation to be of the dimension of S.

$S^\sharp = g(S) \Rightarrow S = g^{-1}(S^\sharp)$ (linear mapping).

Similarly, Lagrange multiplier Y_1 is mapped to the dimension of S.

$$S = \underset{S}{\operatorname{argmin}} \frac{1}{2}\|W - RS\|_F^2 + \frac{\beta}{2}\|g^{-1}(S^\sharp) - S\|_F^2 + < g^{-1}(Y_1), g^{-1}(S^\sharp) - S > +$$

$$< Y_2, S - SC_1 > + \frac{\beta}{2}\|S - SC_1\|_F^2.$$

$$= \underset{S}{\operatorname{argmin}} \frac{1}{2}\|W - RS\|_F^2 + \frac{\beta}{2}(\|g^{-1}(S^\sharp)\|_F^2 + \|S\|_F^2 - 2Tr((g^{-1}(S^\sharp))^T S) +$$

$$Tr((g^{-1}(Y_1))^T(g^{-1}(S^\sharp))) - Tr((g^{-1}(Y_1))^T S) + < Y_2, S - SC_1 > + \frac{\beta}{2}\|S - SC_1\|_F^2.$$

$$= \underset{S}{\operatorname{argmin}} \frac{1}{2}\|W - RS\|_F^2 + \frac{\beta}{2}\Big(\|S\|_F^2 - 2Tr((g^{-1}(S^\sharp))^T S) - \frac{2}{\beta}Tr((g^{-1}(Y_1))^T S)\Big) +$$

$$< Y_2, S - SC_1 > + \frac{\beta}{2}\|S - SC_1\|_F^2. \ \{S^\sharp, Y_1 \text{ are constants when minimizing over } S\}$$

Since, adding constants to the above form will not affect the solution of S.

Therefore, we are adding $\|g^{-1}(S^\sharp) + \frac{g^{-1}(Y_1)}{\beta})\|_F^2$ inside the second term,

which will give us the form

$$S = \underset{S}{\operatorname{argmin}} \frac{1}{2}\|W - RS\|_F^2 + \frac{\beta}{2}\|S - (g^{-1}(S^\sharp) + \frac{g^{-1}(Y_1)}{\beta})\|_F^2 + < Y_2, S - SC_1 > + \frac{\beta}{2}\|S - SC_1\|_F^2. \tag{B.2}$$

The closed form solution for S can be derived by taking derivative of (B.2) w.r.t to S and

equating to zero.

$$\frac{1}{\beta}(R^TR+\beta I)S+S(I-C_1)(I-C_1^T)=\frac{1}{\beta}R^TW+\left(g^{-1}(S^\sharp)+\frac{g^{-1}(Y_1)}{\beta}-\frac{Y_2}{\beta}(I-C_1^T)\right). \tag{B.3}$$

B.1.2 Solution for $S^\sharp$

$$S^\sharp=\underset{S^\sharp}{\operatorname{argmin}} < Y_1, S^\sharp-g(S)> +\frac{\beta}{2}\|S^\sharp-g(S)\|_F^2+ < Y_3, S^\sharp-S^\sharp C_2> +$$

$$\frac{\beta}{2}\|S^\sharp-S^\sharp C_2\|_F^2+ < Y_8, S^\sharp-J> +\frac{\beta}{2}\|S^\sharp-J\|_F^2.$$

Here, also the first two term and last two terms is condensed to a simpler form for mathematical convenience without affecting the final solution.

$$S^\sharp=\underset{S^\sharp}{\operatorname{argmin}}\ Tr\big(Y_1^TS^\sharp\big)-Tr\big(Y_1^Tg(S)\big)+\frac{\beta}{2}\big(\|S^\sharp\|_F^2+\|g(S)\|_F^2-2Tr((S^\sharp)^Tg(S))\big)$$

$$+ < Y_3, S^\sharp-S^\sharp C_2> +\frac{\beta}{2}\|S^\sharp-S^\sharp C_2\|_F^2+Tr\big(Y_8^TS^\sharp\big)-Tr\big(Y_8^TJ\big)+\frac{\beta}{2}\big(\|S^\sharp|_F^2+\|J\|_F^2+$$

$$-2Tr\big((S^\sharp)^TJ\big).$$

Since, we are minimizing over $S^\sharp$. The terms which are not dependent on $S^\sharp$ is considered as constants, which gives us:

$$S^\sharp=\underset{S^\sharp}{\operatorname{argmin}}\frac{\beta}{2}\big(\|S^\sharp\|_F^2-2Tr(S^\sharp)^T(g(S)-\frac{Y_1}{\beta})\big)+ < Y_3, S^\sharp-S^\sharp C_2> +\frac{\beta}{2}\|S^\sharp-S^\sharp C_2\|_F^2$$

$$+\frac{\beta}{2}\big(\|S^\sharp\|_F^2-2Tr(S^\sharp)^T(J-\frac{Y_8}{\beta})\big).$$

Adding $\|g(S)-\frac{Y_1}{\beta}\|_F^2$ and $\|J-\frac{Y_8}{\beta}\|_F^2$ inside the first term and last term respectively to get the quadratic form. As these terms are constants when minimizing over $S^\sharp$ it will not affect the final solution.

$$S^\sharp=\underset{S^\sharp}{\operatorname{argmin}}\frac{\beta}{2}\|S^\sharp-(g(S)-\frac{Y_1}{\beta})\|_F^2+ < Y_3, S^\sharp-S^\sharp C_2> +\frac{\beta}{2}\|S^\sharp-S^\sharp C_2\|_F^2+$$

$$\frac{\beta}{2}\|S^\sharp-(J-\frac{Y_8}{\beta})\|_F^2. \tag{B.4}$$

The closed form solution for $S^\sharp$ can be derived by taking derivative of (B.4) w.r.t $S^\sharp$ and equat-

ing to zero.

$$S^{\sharp}(2I + (I - C_2)(I - C_2^T)) = \left(g(S) - \frac{Y_1}{\beta}\right) + (J - \frac{Y_8}{\beta}) - \frac{Y_3}{\beta}(I - C_2^T). \tag{B.5}$$

B.1.3 Solution for C_1

$$\begin{aligned} C_1 &= \underset{C_1}{\operatorname{argmin}} < Y_2, S - SC_1 > + \frac{\beta}{2}\|S - SC_1\|_F^2 + < Y_4, 1^T C_1 - 1^T > + \\ &\frac{\beta}{2}\|1^T C_1 - 1^T\|_F^2 + < Y_6, C_1 - E_1 > + \frac{\beta}{2}\|C_1 - E_1\|_F^2. \\ &= \underset{C_1}{\operatorname{argmin}} \frac{\beta}{2}\|SC_1 - (S + \frac{Y_2}{\beta})\|_F^2 + \frac{\beta}{2}\|1^T C_1 - (1^T - \frac{Y_4}{\beta})\|_F^2 + \frac{\beta}{2}\|C_1 - (E_1 - \frac{Y_6}{\beta})\|_F^2. \end{aligned} \tag{B.6}$$

The closed form solution for C_1 is solved as:

$$(S^T S + 11^T + I)C_1 = S^T(S + \frac{Y_2}{\beta}) + 1(1^T - \frac{Y_4}{\beta}) + (E_1 - \frac{Y_6}{\beta}). \tag{B.7}$$

$$C_1 = C_1 - \operatorname{diag}(C_1), \tag{B.8}$$

B.1.4 Solution for C_2

$$\begin{aligned} C_2 &= \underset{C_2}{\operatorname{argmin}} < Y_3, S^{\sharp} - S^{\sharp}C_2 > + \frac{\beta}{2}\|S^{\sharp} - S^{\sharp}C_2\|_F^2 + < Y_5, 1^T C_2 - 1^T > + \\ &+ \frac{\beta}{2}\|1^T C_2 - 1^T\|_F^2 + < Y_7, C_2 - E_2 > + \frac{\beta}{2}\|C_2 - E_2\|_F^2. \\ &= \underset{C_2}{\operatorname{argmin}} \frac{\beta}{2}\|S^{\sharp}C_2 - (S^{\sharp} + \frac{Y_3}{\beta})\|_F^2 + \frac{\beta}{2}\|1^T C_2 - (1^T - \frac{Y_5}{\beta})\|_F^2 + \frac{\beta}{2}\|C_2 - (E_2 - \frac{Y_7}{\beta})\|_F^2. \end{aligned} \tag{B.9}$$

The closed form solution for C_2 is derived as:

$$\left((S^{\sharp})^T S^{\sharp} + 11^T + I\right) C_2 = (S^{\sharp})^T(S^{\sharp} + \frac{Y_3}{\beta}) + 1(1^T - \frac{Y_5}{\beta}) + (E_2 - \frac{Y_7}{\beta}). \tag{B.10}$$

$$C_2 = C_2 - \operatorname{diag}(C_2), \tag{B.11}$$

B.1.5 Solution for E_1

$$
\begin{aligned}
E_1 &= \underset{E_1}{\operatorname{argmin}} \lambda_1 \|E_1\|_1 + \gamma_1 \|E_1\|_F^2 + < Y_6, C_1 - E_1 > + \frac{\beta}{2} \|C_1 - E_1\|_F^2. \\
&= \underset{E_1}{\operatorname{argmin}} \lambda_1 \|E_1\|_1 + \gamma_1 \|E\|_F^2 + \frac{\beta}{2} \|E_1 - (C_1 + \frac{Y_6}{\beta})\|_F^2. \\
&= \underset{E_1}{\operatorname{argmin}} \lambda_1 \|E_1\|_1 + \gamma_1 \|E_1\|_F^2 + \frac{\beta}{2} \|E_1\|_F^2 - \beta < E_1, (C_1 + \frac{Y_6}{\beta}) > \\
&= \underset{E_1}{\operatorname{argmin}} \lambda_1 \|E_1\|_1 + (\gamma_1 + \frac{\beta}{2})(\|E_1\|_F^2 + \frac{2\beta}{2\gamma_1 + \beta} < E_1, C_1 + \frac{Y_6}{\beta} >). \\
&= \underset{E_1}{\operatorname{argmin}} \lambda_1 \|E_1\|_1 + (\gamma_1 + \frac{\beta}{2}) \|E_1 - \frac{\beta}{2\gamma_1 + \beta}(C_1 + \frac{Y_6}{\beta})\|_F^2.
\end{aligned}
\tag{B.12}
$$

The closed form solution for E_1 is reached as:

$$
E_1 = \mathcal{S}\big[\frac{\lambda_1}{\gamma_1 + \beta/2}\big]\Big(\frac{\beta}{2\gamma_1 + \beta}(C_1 + \frac{Y_6}{\beta})\Big) \tag{B.13}
$$

B.1.6 Solution for E_2

The derivation for the solution of E_2 is similar to the solution of E_1.

$$
\begin{aligned}
E_2 &= \underset{E_2}{\operatorname{argmin}} \lambda_3 \|E_2\|_1 + \gamma_3 \|E_2\|_F^2 + < Y_7, C_2 - E_2 > + \frac{\beta}{2} \|C_2 - E_2\|_F^2 \\
&= \underset{E_2}{\operatorname{argmin}} \lambda_3 \|E_2\|_1 + (\gamma_3 + \frac{\beta}{2}) \|E_2 - \frac{\beta}{2\gamma_3 + \beta}(C_2 + \frac{Y_7}{\beta})\|_F^2.
\end{aligned}
\tag{B.14}
$$

The closed form solution for E_2 is reached as:

$$
E_2 = \mathcal{S}\big[\lambda_3/(\gamma_3 + \beta/2)\big]\Big(\frac{\beta}{2\gamma_3 + \beta}(C_2 + \frac{Y_7}{\beta})\Big). \tag{B.15}
$$

B.2 Tables for each comparison

Table B.1: Table corresponding to Figure 3.7

Datasets	BMM	PND	Zhu et al.	Kumar et al.	Ours
Dance+Yoga	0.045	0.078	0.052	0.046	0.043
Drink+Walking	0.074	0.060	0.083	0.073	0.071
Shark+Stretch	0.024	0.015	0.067	0.025	0.019
Walking+Yoga	0.070	0.072	0.087	0.070	0.066
Face+Pickup	0.032	0.012	0.018	0.025	0.022
Face+Yoga	0.017	0.010	0.028	0.019	0.017
Shark+Yoga	0.035	0.018	0.094	0.037	0.033
Stretch+Yoga	0.039	0.109	0.045	0.039	0.036

Table B.2: Table corresponding to Figure 3.8

Datasets	BMM	PND	Zhu et al.	Kumar et al.	Ours
p2_free_2	0.1973	0.1544	0.1142	0.1992	0.1171
p2_grab_2	0.2018	0.1570	0.0960	0.2080	0.0822
p3_ball_1	0.1356	0.1477	0.0832	0.1348	0.0810
p4_meet_12	0.0802	0.0862	0.0972	0.0821	0.0815
p4_table_12	0.2313	0.1588	0.1322	0.2313	0.0994

Table B.3: Table corresponding to Figure 3.11

Datasets	BMM	PND	Zhu et al.	Kumar et al.	Ours
Face Sequence 1	0.078	0.077	0.082	0.075	0.073
Face Sequence 2	0.059	0.062	0.063	0.050	0.052
Face Sequence 3	0.042	0.051	0.057	0.038	0.039
Face Sequence 4	0.049	0.041	0.056	0.044	0.040

C

Mathematical derivation and discussion related to chapter 4

In this material, we provide a detailed mathematical derivation to the proposed optimization in the chapter (4)[104]. Additionally, we provide additional qualitative results and insights.

C.1 Mathematical Derivations

$$\underset{S_s, S_t^\sharp, C_s, C_t, J_s J_t}{\text{minimize}} \; \mathrm{E} = \frac{1}{2}\|W_s - RS_s\|_F^2 + \frac{\beta}{2}\|S_t^\sharp - \mathscr{T}_1(S_s)\|_F^2 + < Y_1, S_t^\sharp - \mathscr{T}_1(S_s) > + \gamma\|S_t^\sharp\|_* + \lambda_1\|\mathcal{T}_s - \mathcal{T}_s C_s\|_F^2 +$$
$$\lambda_3\|J_s\|_* + \frac{\beta}{2}\|C_s - J_s\|_F^2 + < Y_2, C_s - J_s > + \lambda_2\|\mathcal{T}_t - \mathcal{T}_t C_t\|_F^2 + \lambda_4\|J_t\|_* + \frac{\beta}{2}\|C_t - J_t\|_F^2 + < Y_3, C_t - J_t >$$
subject to:
$$\Psi_s = \xi(C_s, S_s, q); \Psi_t = \xi(C_t, S_t^\sharp, q);$$
$$S_s = \zeta(\Psi_s, \Sigma_s, V_s, N_s); S_t^\sharp = \zeta(\Psi_t, \Sigma_t^\sharp, V_t, N_t);$$
$$W_s = \mathscr{T}_2(W_s, S_s). \tag{C.1}$$

C.1.1 Background

To make the optimization simpler, let's consider an error term that involves the tensor structure

$$\|E_s\|_F^2 = \|\mathcal{T}_s - \mathcal{T}_s C_s\|_F^2. \tag{C.2}$$

Considering the i^{th} term, and using $\|E_{si}\|_F^2 = trace(E_{si}^T E_{si})$
From our notation definition $\mathcal{T}_s = \{(\psi_{s1})(\psi_{s1})^T, (\psi_{s2})(\psi_{s2})^T ..., (\psi_{sK_s})(\psi_{sK_s})^T\}$ and $C_s \in \mathbb{R}^{K_s \times K_s}$

$$\|E_{si}\|_F^2 = trace\Big[\Big((\psi_{si}\psi_{si}^T) - \sum_{j=1}^{K_s} c_{ij}(\psi_{sj}\psi_{sj}^T)\Big)^T \Big((\psi_{si}\psi_{si}^T) - \sum_{j=1}^{K_s} c_{ij}(\psi_{sj}\psi_{sj}^T)\Big)\Big]$$

$$\|E_{si}\|_F^2 = trace\big((\psi_{si}\psi_{si}^T)^T(\psi_{si}\psi_{si}^T)\big) - 2\sum_{j=1}^{K_s} c_{ij} trace\big((\psi_{si}\psi_{si}^T)^T(\psi_{sj}\psi_{sj}^T)\big) + \tag{C.3}$$

$$\sum_{l=1}^{K_s}\sum_{m=1}^{K_s} c_{il}c_{im} trace\big((\psi_{sl}\psi_{sl}^T)^T(\psi_{sm}\psi_{sm}^T)\big).$$

Now using the trace cyclic property and the orthonormality property of matrices.

$$\|E_{si}\|_F^2 = trace(I_d) - 2\sum_{j=1}^{K_s} c_{ij} trace\big((\psi_{sj}^T\psi_{si})(\psi_{si}^T\psi_{sj})\big) + \sum_{l=1}^{K_s}\sum_{m=1}^{K_s} c_{il}c_{im} trace\big((\psi_{sl}^T\psi_{sm})(\psi_{sm}^T\psi_{sl})\big).$$

$$\|E_{si}\|_F^2 = d - 2\sum_{j=1}^{K_s} c_{ij}\Omega_{ij}^s + \sum_{l=1}^{K_s}\sum_{m=1}^{K_s} c_{il}c_{im}\Omega_{lm}^s, \text{ where, } \Omega_{ij}^s = trace\big((\psi_{sj}^T\psi_{si})(\psi_{si}^T\psi_{sj})\big). \tag{C.4}$$

Here, d stands for the dimension. Notice Ω_{ij}^s has a dimension of $d \times d$ which is easy to handle than the total number of points in a dense datasets. Also, it's simple to verify that Ω_{ij}^s is symmetric.

Using Equation (C.4) and $\Omega_s = (\Omega_{ij}^s)_{i,j=1}^{K_s} \in \mathbb{R}^{K_s \times K_s}$, we can rewrite Equation (C.2) as follows

$$\begin{aligned}
&\|E_s\|_F^2 = const - 2trace(C_s\Omega_s) + trace(C_s\Omega_s C_s^T) \\
&\Rightarrow \|E_s\|_F^2 = const - 2trace(C_sL_sL_s^T) + trace((C_sL_s)(C_sL_s)^T), \text{ where } L_sL_s^T = \text{Cholesky}(\Omega_s) \\
&\Rightarrow \|E_s\|_F^2 = const + \|L_s - C_sL_s\|_F^2 \{\because \text{constant w.r.t } C_s \text{ will not affect the minimization}\}
\end{aligned} \tag{C.5}$$

Similarly, other tensor structure can be equivalently represented in the temporal domain.

Overall Optimization

Substituting the above derivation in Equation (C.1) gives us a simpler representation

$$\underset{S_s,S_t^\sharp,C_s,C_t,J_s,J_t}{\text{minimize}}\ \mathrm{E} = \frac{1}{2}\|W_s - RS_s\|_F^2 + \frac{\beta}{2}\|S_t^\sharp - \mathscr{T}_1(S_s)\|_F^2 + < Y_1, S_t^\sharp - \mathscr{T}_1(S_s) > + \gamma\|S_t^\sharp\|_* + \lambda_1\|L_s - C_sL_s\|_F^2 +$$
$$\lambda_3\|J_s\|_* + \frac{\beta}{2}\|C_s - J_s\|_F^2 + < Y_2, C_s - J_s > + \lambda_2\|L_t - C_tL_t\|_F^2 + \lambda_4\|J_t\|_* + \frac{\beta}{2}\|C_t - J_t\|_F^2 + < Y_3, C_t - J_t >$$
subject to:
$$\Psi_s = \xi(C_s, S_s, q);\ \Psi_t = \xi(C_t, S_t^\sharp, q);$$
$$S_s = \zeta(\Psi_s, \Sigma_s, V_s, N_s);\ S_t^\sharp = \zeta(\Psi_t, \Sigma_t^\sharp, V_t, N_t);$$
$$W_s = \mathscr{T}_2(W_s, S_s); \tag{C.6}$$

Solution to S_s

$$\begin{aligned}
&\equiv \underset{S_s}{\text{argmin}}\frac{1}{2}\|W_s - RS_s\|_F^2 + \frac{\beta}{2}\|S_t^\sharp - \mathscr{T}_1(S_s)\|_F^2 + < Y_1, S_t^\sharp - \mathscr{T}_1(S_s) > \\
&\equiv \underset{S_s}{\text{argmin}}\frac{1}{2}\|W_s - RS_s\|_F^2 + \frac{\beta}{2}\|\mathscr{T}_1^{-1}(S_t^\sharp) - S_s\|_F^2 + < \mathscr{T}_1^{-1}(Y_1), \mathscr{T}_1^{-1}(S_t^\sharp) - S_s > \\
&\equiv \underset{S_s}{\text{argmin}}\frac{1}{2}\|W_s - RS_s\|_F^2 + \frac{\beta}{2}\|S_s - \big(\mathscr{T}_1^{-1}(S_t^\sharp) + \frac{\mathscr{T}_1^{-1}(Y_1)}{\beta}\big)\|_F^2.
\end{aligned} \tag{C.7}$$

The solution to S_s can be derived by differentiating the above term w.r.t S_s and equating it to zero.

$$S_s \equiv (R^TR + \beta I)^{-1}\Big(\beta\big(\mathscr{T}_1^{-1}(S_t^\sharp) + \frac{\mathscr{T}_1^{-1}(Y_1)}{\beta}\big) + R^TW_s\Big)$$

Solution to $S_t^\sharp$

$$\begin{aligned}
&\equiv \underset{S_t^\sharp}{\text{argmin}}\gamma\|S_t^\sharp\|_* + \frac{\beta}{2}\|S_t^\sharp - \mathscr{T}_1(S_s)\|_F^2 + < Y_1, S_t^\sharp - \mathscr{T}_1(S_s) > \\
&\equiv \underset{S_t^\sharp}{\text{argmin}}\gamma\|S_t^\sharp\|_* + \frac{\beta}{2}\|S_t^\sharp - \big(\mathscr{T}_1(S_s) - \frac{Y_1}{\beta}\big)\|_F^2
\end{aligned} \tag{C.8}$$

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

$$\begin{aligned}
&S_t^\sharp \equiv U_t\mathcal{S}[\gamma/\beta](\Sigma_t)V_t, \\
&\text{where, } [U_t, \Sigma_t, V_t] = \text{svd}(\mathscr{T}_1(S_s) - \frac{Y_1}{\beta})
\end{aligned} \tag{C.9}$$

Solution to C_s

$$\begin{aligned} &\equiv \underset{C_s}{\operatorname{argmin}} \lambda_1 \|L_s - C_s L_s\|_F^2 + \frac{\beta}{2}\|C_s - J_s\|_F^2 + < Y_2, C_s - J_s > \\ &\equiv \underset{C_s}{\operatorname{argmin}} \lambda_1 \|L_s - C_s L_s\|_F^2 + \frac{\beta}{2}\|C_s - \big(J_s - \frac{Y_2}{\beta}\big)\|_F^2 \end{aligned} \tag{C.10}$$

The solution to C_s can be derived by differentiating the above term w.r.t C_s and equating it to zero.

$$C_s \equiv \Big(2\lambda_1 L_s L_s^T + \beta(J_s - \frac{Y_2}{\beta})\Big)\big(2\lambda_1 L_s L_s^T + \beta I_s\big)^{-1}$$

Solution to C_t

Similar to the C_s solution derivation, it's solution can be derived as follows:

$$\begin{aligned} &\equiv \underset{C_t}{\operatorname{argmin}} \lambda_2 \|L_t - C_t L_t\|_F^2 + \frac{\beta}{2}\|C_t - J_t\|_F^2 + < Y_3, C_t - J_t > \\ &\equiv \underset{C_t}{\operatorname{argmin}} \lambda_2 \|L_t - C_t L_t\|_F^2 + \frac{\beta}{2}\|C_t - \big(J_t - \frac{Y_3}{\beta}\big)\|_F^2 \end{aligned} \tag{C.11}$$

$$C_t \equiv \Big(2\lambda_2 L_t L_t^T + \beta(J_t - \frac{Y_3}{\beta})\Big)\big(2\lambda_2 L_t L_t^T + \beta I_t\big)^{-1}$$

Solution to J_s

$$\begin{aligned} &\equiv \underset{J_s}{\operatorname{argmin}} \lambda_3 \|J_s\|_* + \frac{\beta}{2}\|C_s - J_s\|_F^2 + < Y_2, C_s - J_s > \\ &\equiv \underset{J_s}{\operatorname{argmin}} \lambda_3 \|J_s\|_* + \frac{\beta}{2}\|J_s - \big(C_s + \frac{Y_2}{\beta}\big)\|_F^2 \end{aligned} \tag{C.12}$$

Similar to Equation C.9 derivation, using the soft-thresholding operation, its optimal solution can be obtained as

$$J_s \equiv U_{J_s}\mathcal{S}[\lambda_3/\beta](\Sigma_{J_s})V_{J_s}, \text{ where } [U_{J_s}, \Sigma_{J_s}, V_{J_s}] = \text{svd}(C_s + \frac{Y_2}{\beta}) \tag{C.13}$$

Solution to J_t

$$\begin{aligned} &\equiv \underset{J_t}{\operatorname{argmin}} \lambda_4 \|J_t\|_* + \frac{\beta}{2}\|C_t - J_t\|_F^2 + < Y_3, C_t - J_t > \\ &\equiv \underset{J_t}{\operatorname{argmin}} \lambda_4 \|J_t\|_* + \frac{\beta}{2}\|J_t - \big(C_t + \frac{Y_3}{\beta}\big)\|_F^2 \end{aligned} \tag{C.14}$$

$$J_t \equiv U_{J_t}\mathcal{S}[\lambda_4/\beta](\Sigma_{J_t})V_{J_t}, \text{ where } [U_{J_t}, \Sigma_{J_t}, V_{J_t}] = \text{svd}(C_t + \frac{Y_3}{\beta}) \tag{C.15}$$

C.1.2 Proof

We have stated in the Algorithm table that $\Omega_s \succeq 0$ The following lemma provides the proof for the same.

Lemma C.1.1. *Given a set of orthonormal matrices* $\Psi_s = \left\{ \{\psi_{si}\}_{i=1}^{K_s} : \forall \psi_{si} \in \mathbb{R}^{d \times n}, {\psi_{si}}^T \psi_{si} = I \right\}$, *if* $\exists\ \Omega_{ij}^s = trace[\left(\psi_{sj}^T \psi_{si}\right)\left(\psi_{si}^T \psi_{sj}\right)]$ *such that* $\Omega_s = (\Omega_{ij}^s)_{i,j=1}^{K_s} \in \mathbb{R}^{K_s \times K_s}$, *then* $\Omega_s \succeq 0$.

Proof. $Z_i = \psi_{si}\psi_{si}^T$ is a $d \times d$ symmetric matrix.

$$
\begin{aligned}
\text{As per the statement, } \Omega_{ij}^s &= trace[\left(\psi_{sj}^T \psi_{si}\right)\left(\psi_{si}^T \psi_{sj}\right)] = trace[\left(\psi_{sj}\psi_{sj}^T\right)\left(\psi_{si}\psi_{sj}^T\right)] \\
&= trace(Z_j Z_i) = trace(Z_j Z_i^T) = trace(Z_i^T Z_j) \\
\Omega_s &= (\Omega_{ij}^s)_{i,j=1}^{K_s}, \text{ then, } \Omega_s = Z^T Z \quad \{\text{Skipping some elementary steps}\} \\
&\Rightarrow \Omega_s \succeq 0.
\end{aligned}
\tag{C.16}
$$

□

Similarly, the positive semi-definite proof for Ω_t can be derived. Note: In case $\Omega_s = 0 || \Omega_t = 0$ while implementing this algorithm, then add ∂ (a very small positive number) to the diagonal elements of Ω_s or Ω_t accordingly, to get to an approximate Cholesky factorization. Mathematically, approximate $\Omega_s = 0 || \Omega_t = 0$ as $\Omega \approx \Omega + \partial I$ to make it numerically positive definite.

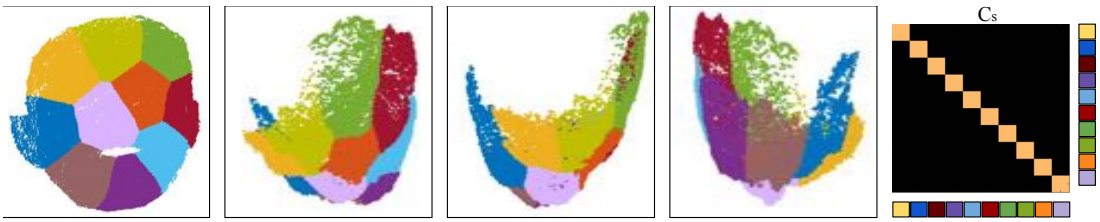
C.2 Qualitative Results

C.2.1 Analysis of C_s and C_t

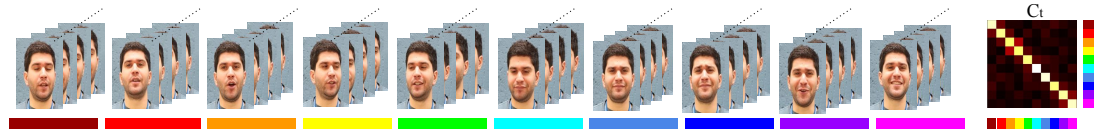
In the experiment section we mentioned about the observation of C_s and C_t matrix. Since, no ground-truth data's are available to quantify these matrices, we provide a visual observation for the same. We used the spectral clustering [137] to group the trajectories and shapes after convergence to infer the output of C_s and C_t matrix. Fig.(C.1) shows the output of this experiment. Visually it can be observed that local low-rank linear subspace are properly procured —both spatially and temporally.

C.3 Rotation Estimate

We used the method proposed by Dai *et al.* [44] to estimate rotation which only depends on the K value (model complexity) and therefore, it can efficiently handle dense feature correspondence over multiple frame to estimate rotation. Assuming that a single non-rigid deforming object constitutes a global relative camera pose over frames is a reasonable choice



(a)



(b)

Figure C.1: (a) Grouping of the trajectories based on C_s matrix. We provide four different views to check the fidelity of our result and assumption. (b) Grouping of the shapes based on C_t matrix. Color corresponding to the group block is shown with the color bars (extreme right). This simulation is done on the real face sequence [65] with K_s and K_t = 10.

and works efficiently. Most of the past approaches also used this assumption to solve rotation [44, 41, 7, 107, 109]. Quantitative results on several datasets also shows that high-quality reconstruction can be obtained under such assumption. *Additionally, it has also been observed that different camera path can lead to different reconstruction results. For now, its investigation on the algorithm performance is left for future discussions.*

Note: For technical details on the compactness of grassmannians, kindly refer to [132] for comprehensive theory. Nevertheless, there are many other books and notes on differential manifolds which provides information on the compactness of grassmannians.

D

Mathematical Derivations and Extra Experimental Analysis of Chapter 5

This appendix provides mathematical derivation to the objective function proposed in chapter (5). We provide some more qualitative results and statistical evaluations of our algorithm. Lastly, we made a brief comment on the challenges associated with handling temporal grassmannians for NRSfM problem.

D.1 Mathematical derivation to the optimization of the objective function

In this section, we provide mathematical derivation of the following optimization proposed in the paper.

$$\begin{aligned}
&\underset{Z,\tilde{C},S,S^{\sharp}}{\text{minimize}} \frac{1}{2}\|W - RS\|_F^2 + \beta_1\|\chi - \chi\tilde{C}\|_F^2 + \beta_2\|S^{\sharp}\|_* + \\
&\qquad \frac{\varrho}{2}\|S^{\sharp} - f(S)\|_F^2 + < \mathrm{L}_1, S^{\sharp} - f(S) > + \beta_3\|Z\|_* + \\
&\qquad \frac{\varrho}{2}\|\tilde{C} - Z\|_F^2 + < \mathrm{L}_2, \tilde{C} - Z > \\
&\text{subject to: } \xi = f_g(\mathrm{P}, S), \tilde{\xi} = f_h(\Delta, \xi), \\
&S = f_s(\xi, \Sigma, \xi_v), \mathrm{P} = f_p(\tilde{\xi}, \tilde{C}, \mathrm{P}_o)
\end{aligned} \tag{D.1}$$

The constraints in the Eq:(D.1) are invoked over iteration. The solution to each subproblem is obtained by taking the derivative of the above ALM form w.r.t the concerned variable and equating it to zero.

D.1.1 Solution to 'S'

$$\begin{aligned}
&\equiv \underset{S}{\text{argmin}} \frac{1}{2}\|W - RS\|_F^2 + \frac{\varrho}{2}\|S^{\sharp} - f(S)\|_F^2 + \\
&\qquad < \mathrm{L}_1, S^{\sharp} - f(S) > \\
&\equiv \underset{S}{\text{argmin}} \frac{1}{2}\|W - RS\|_F^2 + \frac{\varrho}{2}\|S - \left(f^{-1}(S^{\sharp}) + \frac{f^{-1}(\mathrm{L}_1)}{\varrho}\right)\|_F^2
\end{aligned} \tag{D.2}$$

Taking the derivative of the above equation w.r.t 'S' and equating it to zero gives

$$(R^T R + \varrho I)S = R^T W + \varrho\left(f^{-1}(S^{\sharp}) + \frac{f^{-1}(\mathrm{L}_1)}{\varrho}\right) \tag{D.3}$$

We used MATLAB mldivide() function to solve it during our implementation. You may use any linear algebra package to solve the above well known form.

D.1.2 Solution to '$S^\sharp$'

Similar to previous derivation, we can write the ALM form for the variable $S^\sharp$

$$\begin{aligned} &\equiv \underset{S^\sharp}{\operatorname{argmin}} \beta_2 \|S^\sharp\|_* + \frac{\varrho}{2}\|S^\sharp - f(S)\|_F^2 + < \mathrm{L}_1, S^\sharp - f(S) > \\ &\equiv \underset{S^\sharp}{\operatorname{argmin}} \beta_2 \|S^\sharp\|_* + \frac{\varrho}{2}\|S^\sharp - \left(f(S) - \frac{\mathrm{L}_1}{\varrho}\right)\|_F^2 \end{aligned} \tag{D.4}$$

The above sub-problem is well-known form for nuclear norm minimization. By defining the soft-thresholding operator $\mathcal{S}[\tau](v) = \operatorname{sign}(v)\max(|v| - \tau)$, the solution of $S^\sharp$ can be obtained by

$$S^\sharp = U_s \mathcal{S}[\beta_2/\varrho](\Sigma_s) V_s \tag{D.5}$$

where, $[U_s, \Sigma_s, V_s] = \operatorname{svd}\left(f(S) - \mathrm{L}_1/\varrho\right)$

D.1.3 Solution to 'Z'

$$\begin{aligned} &\equiv \underset{Z}{\operatorname{argmin}} \beta_3 \|Z\|_* + \frac{\varrho}{2}\|\tilde{C} - Z\|_F^2 + < \mathrm{L}_2, \tilde{C} - Z > \\ &\equiv \underset{Z}{\operatorname{argmin}} \beta_3 \|Z\|_* + \frac{\varrho}{2}\|Z - \left(\tilde{C} + \frac{\mathrm{L}_2}{\varrho}\right)\|_F^2 \end{aligned} \tag{D.6}$$

Using the soft-thresholding function as mentioned before, the solution to Z is given by

$$Z \equiv U_z \mathcal{S}[\beta_3/\varrho](\Sigma_z) V_z \tag{D.7}$$

where $[U_z, \Sigma_z, V_z] = \operatorname{svd}\left(\tilde{C} + \mathrm{L}_2/\varrho\right)$

D.1.4 Solution to '$\tilde{C}$'

Deriving the solution for '$\tilde{C}$' from the sub-problem involving the variable '$\tilde{C}$' is not straight forward rather, it's a bit involved and therefore, we first derive an equivalent form for the error term that involves tensor $\mathcal{X}$. The equivalent form is easy to handle and program on computers. Lets consider the following error term:

$$\|\mathcal{X} - \mathcal{X}\tilde{C}\|_F^2 \tag{D.8}$$

Using the notation from our paper, for any i^{th} Grassmann point this error term in Eq:(D.8) can be written as

$$\operatorname{Tr}\left(\left((\Theta_i\Theta_i^T) - \sum_{j=1}^{K} c_{ij}(\Theta_j\Theta_j^T)\right)^T \left((\Theta_i\Theta_i^T) - \sum_{j=1}^{K} c_{ij}(\Theta_j\Theta_j^T)\right)\right) \tag{D.9}$$

Expanding the above form gives

$$\begin{aligned}
&\equiv \mathrm{Tr}\big((\Theta_i\Theta_i^T)^T(\Theta_i\Theta_i^T)\big) - 2\sum_{j=1}^{K} c_{ij}\mathrm{Tr}\big((\Theta_i\Theta_i^T)^T(\Theta_j\Theta_j^T)\big) + \\
&\sum_{l=1}^{K}\sum_{m=1}^{K} c_{il}c_{im}\mathrm{Tr}\big((\Theta_l\Theta_l^T)^T(\Theta_m\Theta_m^T)\big)
\end{aligned} \tag{D.10}$$

From our definition $\Theta \in \mathbb{R}^{\tilde{d}\times p}$ as an orthonormal matrix. Using it simplifies the above equation to:

$$\begin{aligned}
&\equiv p - 2\sum_{j=1}^{K} c_{ij}\Gamma_{ij} + \sum_{l=1}^{K}\sum_{m=1}^{K} c_{il}c_{im}\Gamma_{lm} \\
&\text{where, } \Gamma_{ij} = \mathrm{Tr}\big((\Theta_j^T\Theta_i)(\Theta_i^T\Theta_j)\big) \ \{\text{using trace cyclic property}\}
\end{aligned} \tag{D.11}$$

Let $\Gamma = (\Gamma_{ij})_{ij=1}^{K} \in \mathbb{R}^{K\times K}$. Its easy to verify that Γ is symmetric positive semi-definite. Therefore, using cholesky factorization of $\mathrm{chol}(\Gamma) = LL^T$, we can re-write the above equation as

$$\begin{aligned}
&\equiv p - 2\mathrm{Tr}(\tilde{C}LL^T) + \mathrm{Tr}(\tilde{C}LL^T\tilde{C}^T) \\
&\equiv \text{const} + \|L - \tilde{C}L\|_F^2 \\
&\text{where, const. means constant w.r.t } \tilde{C}
\end{aligned} \tag{D.12}$$

By substituting the result from Eq:(D.12) to the sub-problem w.r.t $\tilde{C}$, we get the following form:

$$\begin{aligned}
&\equiv \underset{\tilde{C}}{\mathrm{argmin}}\ \beta_1\|L - \tilde{C}L\|_F^2 + \frac{\varrho}{2}\|\tilde{C} - Z\|_F^2 + < \mathrm{L}_2, \tilde{C} - Z > \\
&\equiv \underset{\tilde{C}}{\mathrm{argmin}}\ \beta_1\|L - \tilde{C}L\|_F^2 + \frac{\varrho}{2}\|\tilde{C} - \Big(Z - \frac{\mathrm{L}_2}{\varrho}\Big)\|_F^2
\end{aligned} \tag{D.13}$$

Taking the derivative of the Eq:(D.13) w.r.t $\tilde{C}$ and equating it to zero.

$$\tilde{C}(2\beta_1 LL^T + \varrho I) = 2\beta_1 LL^T + \varrho\Big(Z - \frac{\mathrm{L}_2}{\varrho}\Big) \tag{D.14}$$

D.2 Solution to $E(\Delta)$

$$\begin{aligned}
&E(\Delta) \equiv \underset{\Delta}{\text{minimize}} \sum_{(i,j)}^{K} w_{ij}\frac{1}{2}\|\Delta^T(\Lambda_{ij})\Delta\|_F^2 \\
&\text{subject to:} \\
&\mathrm{Tr}\Big(\Delta^T\big(\sum_{i=1}^{K}\lambda_{ii}\Omega_i\Omega_i^T\big)\Delta\Big) = 1
\end{aligned} \tag{D.15}$$

The optimization equation proposed for $E(\Delta)$ 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 $X = \left(\sum_{i=1}^{K} \lambda_{ii}\Omega_i\Omega_i^T\right)$ and expanding the Frobenius norm term, we can re-write the equation as:

$$
\begin{aligned}
E(\Delta) &\equiv \underset{\Delta}{\text{minimize}} \sum_{(i,j)}^{K} \frac{w_{ij}}{2} \text{Tr}\left(\Delta^T \Lambda_{ij} \Delta \Delta^T \Lambda_{ij} \Delta\right) \\
E(\Delta) &\equiv \underset{\Delta}{\text{minimize}}\ \text{Tr}\left(\Delta^T \sum_{(i,j)}^{K} \frac{w_{ij}}{2} \Lambda_{ij} \Delta^{t-1} \Delta^{(t-1)T} \Lambda_{ij} \Delta\right) \qquad \text{(D.16)} \\
&\text{subject to:} \\
&\text{Tr}\left(\Delta^T X \Delta\right) = 1
\end{aligned}
$$

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

$$
\begin{aligned}
&E(\Delta) \equiv \underset{\Delta}{\text{minimize}}\ \text{Tr}(\Delta^T Y \Delta) \\
&\text{subject to:} \qquad \text{(D.17)} \\
&\text{Tr}\left(\Delta^T X \Delta\right) = 1
\end{aligned}
$$

The equivalent Lagrangian function form is given by

$$
\text{Tr}(\Delta^T Y \Delta) + \lambda\left(1 - \text{Tr}\left(\Delta^T X \Delta\right)\right) \qquad \text{(D.18)}
$$

The Eq:(D.18) is of the standard form to generalized eigen value problem. You may use any standard linear algebra package to solve it.

D.3 Discussion

D.3.1 Why we opt not to distrub the temporal continuity for this problem?

Although clustering of frames into smaller groups (Grassmannians) allows simpler model, however, its quite possible that there will be repeat of certain activities or expression in the video sequence (say facial expression). In such cases the Grassmannians at frame 'f' and frame 'f+n' will be assigned to same group. Here, 'n' is the time instant at which activities repeat or is similar. As a result, such representation procedure may disturb the overall time continuity of the sequence. Also, these group of frames may form high-dimensional grassmannians, in order to project it into low-dimension using

neighboring Grassmannians will get extremely difficult, for example, how to decide neighboring grassmannians using temporal grassmann samples?. On the other hand, grouping of trajectories (spatial) does not disturb the temporal continuity of the trajectory and we can easily define the neighbors using spatial information *i.e.*, spatial neighbors tend to be neighbors throughout the sequence, for a single deforming object (unless breaks or disassociate, which is very rare). But in shape space, we don't have any prior knowledge to define neighboring relation.

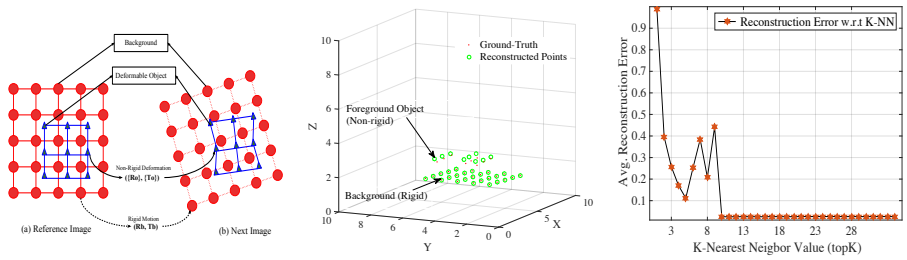
E

Code and Extra Experimental Analysis of Chapter 7

In this supplementary material, we first provide the MATLAB simulation code on two synthetic examples. These examples explains and show the utility of as rigid as possible constraint to recover the 3D points in a dynamic scene setting without estimating motion. Secondly, we provide few more statistical experiment results about the behavior of our algorithm under noisy initialization and different $di\sigma$ values (if the second constraint is used with Φ^{arap}). Although some of the evaluations are also provided in the main paper, we provide it again with numerical examples for completeness and easy understanding. Lastly, we provide some general discussion on our approach.

E.1 Synthetic Experiment Code and Explanation

We provide the code showing the utility of as rigid as possible constraint on two synthetic experimental setting of a dynamic scene. In these experiments, the background and the objects are shown in red and blue color respectively. The background undergoes a rigid motion and the object undergoes a non-rigid deformation in the scene. Given the depth of the reference frame and the image correspondences of the feature points, we can estimate the 3D reconstruction for both the foreground and the background in the next frame just by using the ARAP constraint without using any 3D motion parameters.



(a) (b) (c)

Figure E.1: (a) Experimental setup for the first experiment (b) 3D reconstruction for the next frame after optimization (c) The 3D reconstruction error variations against the number of nearest neighbor in the experiment (topK variable in the code).

E.1.1 Experiment (1)

1. Scene Setup: A background and an object in the reference frame. The background undergoes a rigid motion and the single object deforms non-rigidly in the next frame (see Figure E.1).
2. Input: 2D image feature correspondences, intrinsic camera parameters(K), depth of the points in the reference frame.
3. Output: 3D coordinates of the entire scene for the next frame.

(1) firstExample.m Main file.

```
%% Evaluation of concept on sythetic dataset.
% 1. Given the 3D points for the background and the deforming object (foreground) for the reference frame.
% 2. Also, you are provided with camera intrinsic calibration matrix(K), 2D image correspondance between reference frame and next frame
% 3. Situation: The background is undergoing a rigid motion and the object is deforming non-rigidly.
%% Problem: % Get the 3D reconstruction of this dynamic scene for the next time frame without solving for motion.
%% 1. Generate a synthetic dataset for the reference frame
%Create a synthetic situation of the problem.
%generate 3D for the reference frame
%Background coordinate
ref_Xb = [1, 2, 3, 4, 5; 1, 2, 3, 4, 5; 1, 2, 3, 4, 5; 1, 2, 3, 4, 5; 1, 2, 3, 4, 5];
ref_Yb = [1, 1, 1, 1, 1; 2, 2, 2, 2, 2; 3, 3, 3, 3, 3; 4, 4, 4, 4, 4; 5, 5, 5, 5, 5];
ref_Zb = 2 * ones(5, 5);
%Object coordinate
ref_Xo = [2.5, 3.5, 4.5; 2.5, 3.5, 4.5; 2.5, 3.5, 4.5];
```

```
ref_Yo = [2.5, 2.5, 2.5; 3.5, 3.5, 3.5; 5.0, 5.0, 5.0];
ref_Zo = 3 * ones(3, 3);
%Arrange in the matrix form
ref_Xb = ref_Xb'; ref_Yb = ref_Yb'; ref_Zb = ref_Zb';
ref_Xo = ref_Xo'; ref_Yo = ref_Yo'; ref_Zo = ref_Zo';
ref_X = [ref_Xb(:)', ref_Xo(:)'];
ref_Y = [ref_Yb(:)', ref_Yo(:)'];
ref_Z = [ref_Zb(:)', ref_Zo(:)'];
%% 2. Generate the synthetic dataset for the next frame
%give some rigid motion to the background
angle = deg2rad(3);
R = [cos(angle), 0, sin(angle); 0, 1, 0; -sin(angle), 0, cos(angle)];
t = [0.2, 0.2, 0.2]';
next_b = R*[ref_Xb(:)'; ref_Yb(:)'; ref_Zb(:)'] + repmat(t, [1, 25]);
next_Xb = next_b(1, :); next_Yb = next_b(2, :); next_Zb = next_b(3, :);
%give some inconsistent changes to the object
next_Xo = [2.6, 3.7, 4.7; 2.8, 3.6, 4.5; 2.5, 3.5, 4.6];
next_Yo = [2.6, 2.7, 2.75; 3.4, 3.45, 3.5; 5.05, 5.10, 5.15];
next_Zo = [2.9, 2.9, 2.9; 2.9, 2.9, 2.9; 2.9, 2.9, 2.9];
%arrange in the matrix form
next_Xo = next_Xo'; next_Yo = next_Yo'; next_Zo = next_Zo';
next_X = [next_Xb, next_Xo(:)'];
next_Y = [next_Yb, next_Yo(:)'];
next_Z = [next_Zb, next_Zo(:)'];
%% 3. Generate synthetic image for the reference frame and the next frame.
%some K matrix
fx = 100; fy = 100; cx = 240; cy = 320;
K = [fx, 0, cx; 0, fy, cy; 0, 0, 1];
%image point for the reference image
ref_img = K*[ref_X;ref_Y; ref_Z];
ref_img = ref_img./repmat(ref_img(3, :), [3, 1]);
%image point for the next image
next_img = K*[next_X; next_Y; next_Z];
next_img = next_img./repmat(next_img(3, :), [3, 1]);
%plot the image points
figure, plot(ref_img(1, :), ref_img(2, :), 'k.'); hold on;
plot(ref_img(1, 26:34), ref_img(2, 26:34), 'ro'); title('Reference Image');
figure, plot(next_img(1, :), next_img(2, :), 'k.'); hold on;
plot(next_img(1, 26:34), next_img(2, 26:34), 'ro'); title('Next Image');
%% 4. Define the neighbors based on the reference image distance
%total number of anchor node.
```

```
N = 34; %K-NN to consider
topK = 15; %vary form 1 to N
%get the index of the neighbors
[persuperpixelKNNid, persuperpixelwik] = givemeKNN(ref_img, N, topK);  %function call 1
%% 5. Use as rigid as possible optimization routine
%(Optional: You may provide explicit lower and upper bound for better convergence of a non-convex
problem)
%(For large scale problems such bounds can be handy)
%dvariance = ones(N, 1);
%lb = ref_Z' - dvariance; %lower bound on the variables
%ub = ref_Z' + dvariance; %upper bound on the variables
%general upper and lower bound
lb = zeros(N, 1); ub = []; Aeq = []; Beq = []; A = []; B = []; do = ones(N, 1)/N;
%optimization options
%for MATLAB 2017 version uncomment
%options = optimoptions('fmincon', 'Algorithm', 'sqp', 'Display', 'iter-detailed', 'MaxIter', 1000,
'MaxFunctionEvaluations', 300000, 'PlotFcns', @optimplotfval);
%for MATLAB 2015 version
options = optimoptions('fmincon', 'Algorithm', 'sqp', 'Display', 'iter-detailed', 'MaxIter', 1000, 'Max-
FunEvals', 300000, 'PlotFcns', @optimplotfval);
ref3D = [ref_X; ref_Y; ref_Z];
next3D = inv(K)*next_img;
disp('Optimizing....');
[depthVal, cost] = fmincon(@(d)objectiveFunctionARAP(d, ref3D, next3D, persuperpixelKNNid,
persuperpixelwik), do, A, B, Aeq, Beq, lb, ub,[], options); %function call 2
%% 6. Get the output depth and estimate the 3D.
output3D = zeros(3, N);
for i = 1:N
    output3D(:, i) = depthVal(i)*next3D(:, i);
end
%% 7. Plot the result
figure,
plot3(next_X(:), next_Y(:), next_Z(:), 'r.'); hold on;
plot3(output3D(1, :), output3D(2, :), output3D(3, :), 'go');
axis([0, 10, 0, 10, 0, 10]); grid on;
title('3D reconstruction for the next frame');
legend('Ground-Truth', 'Reconstructed Points')
%% 8. Perform error estimation (Relative Error)
gt_3D = [next_X(:)'; next_Y(:)'; next_Z(:)'];
es_3D = [output3D(1, :); output3D(2, :); output3D(3, :)];
error = norm(es_3D - gt_3D, 'fro')/norm(gt_3D, 'fro');
```

```
fprintf('Relative Error = %f \n', error);
```

(2) GIVEMEKNN.M First function file (K-nearest neighboring index)

```
function [persuperpixelKNNid, persuperpixelwik] = givemeKNN(ref_img, N, topK)
persuperpixelKNNid = cell(1, N); persuperpixelwik = cell(1, N); distanceMat = zeros(N, N);
for i = 1:N
    x_ai = ref_img(1:2, i);
    for j = 1:N
        x_ak = ref_img(1:2, j);
        distanceMat(i, j) = sqrt((x_ai(1, 1) - x_ak(1, 1))^2 + (x_ai(2, 1) - x_ak(2, 1))^2);
    end
end
[sortDistance, index] = sort(distanceMat, 2);
betad = 1;
for i = 1:N
    persuperpixelKNNidi.knnid = index(i, 2:topK); %1 id is always the same anchor (distance to itself
= 0);
    persuperpixelwiki.wik = exp(-betad*sortDistance(i, 2:topK));
end
end
```

(3) OBJECTIVEFUNCTIONARAP.M Second function file (As rigid as possible cost function definition).

```
function cost = objectiveFunctionARAP(d, ref3D, next3D, persuperpixelKNNid, persuperpixelwik)
N = length(persuperpixelKNNid);
cost = 0;
for i = 1:N
    knnid = persuperpixelKNNidi.knnid;
    di = d(i);
    Xi = ref3D(:, i);
    Xip = next3D(:, i);
    for j = 1:length(knnid)
        dj = d(knnid(1, j));
        Xj = ref3D(:, knnid(1, j));
        Xjp = next3D(:, knnid(1, j));
        cost = cost + abs(norm(Xi-Xj)-norm(di*Xip - dj*Xjp));
    end
end
end
```

E.1.2 Experiment (2)

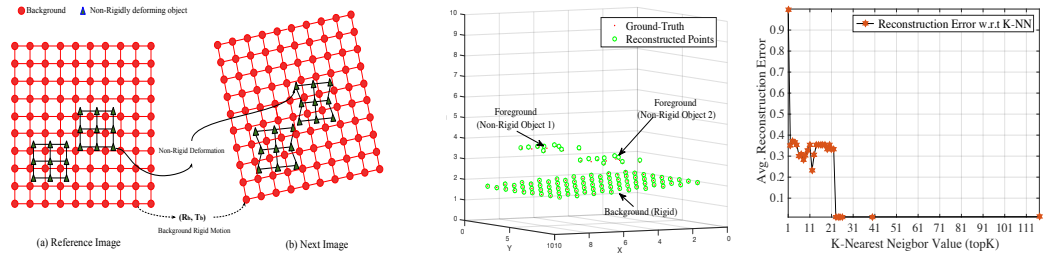
1. Scene Setup: A background with two objects in the reference frame scene. The background undergoes a rigid motion and both the objects deforms non-rigidly in the next frame (see Figure E.2).

2. Input: 2D image feature correspondences, intrinsic camera parameters(K), depth of the points in the reference frame.

3. Output: 3D coordinates of the entire scene for the next frame.

secondExample.m Main file.

```
% 1. Given the 3D points for the background and the two foreground object for the reference frame.
% 2. Also, you are provided with 2D image correspondance between reference frame and next frame.
% The 3D background is undergoing rigid motion and the two foreground are undergoing non-rigid
deformation.
% 3. use ARAP constraint to estimate the 3D output for the next frame.

%% 1. Generate a synthetic dataset for the reference frame

%3D in the reference frame.
ref_Xb = repmat(1 : 10, [10, 1]);
ref_Yb = ones(10, 10). * repmat((1 : 10)', [1, 10]);
ref_Zb = 2 * ones(10, 10);

ref_Xo1 = [2.5, 3.5, 4.5; 2.5, 3.5, 4.5; 2.5, 3.5, 4.5];
ref_Yo1 = [2.5, 2.5, 2.5; 3.5, 3.5, 3.5; 5.0, 5.0, 5.0];
ref_Zo1 = 3 * ones(3, 3);

ref_Xo2 = [7.5, 8.5, 9.5; 7.5, 8.5, 9.5; 7.5, 8.5, 9.5];
ref_Yo2 = [5.5, 5.5, 5.5; 6.5, 6.5, 6.5; 8.0, 8.0, 8.0];
ref_Zo2 = 4 * ones(3, 3);

% figure, plot3(ref_Xb(:), ref_Yb(:), ref_Zb(:), 'r*'); hold on;
% plot3(ref_Xo1(:), ref_Yo1(:), ref_Zo1(:), 'g.'); hold on;
% plot3(ref_Xo2(:), ref_Yo2(:), ref_Zo2(:), 'g.'); hold on;

ref_Xb = ref_Xb'; ref_Yb = ref_Yb'; ref_Zb = ref_Zb';
ref_Xo1 = ref_Xo1'; ref_Yo1 = ref_Yo1'; ref_Zo1 = ref_Zo1';
ref_Xo2 = ref_Xo2'; ref_Yo2 = ref_Yo2'; ref_Zo2 = ref_Zo2';

ref_X = [ref_Xb(:)', ref_Xo1(:)', ref_Xo2(:)'];
```



(a) (b) (c)

Figure E.2: (a) Experimental setup for the second experiment (b) 3D reconstruction of the points in the next frame after optimization (c) The 3D reconstruction error variations against the number of nearest neighbor in the experiment (topK variable in the code)

```
ref_Y = [ref_Yb(:)', ref_Yo1(:)', ref_Yo2(:)'];
ref_Z = [ref_Zb(:)', ref_Zo1(:)', ref_Zo2(:)'];
plot3(ref_X(:), ref_Y(:), ref_Z(:), 'ro'); hold on;

%% 2. Generate the synthetic dataset for next frame
angle = deg2rad(3);
R = [cos(angle), 0, sin(angle); 0, 1, 0; -sin(angle), 0, cos(angle)];
t = [0.2, 0.2, 0.2]';
next_b = R*[ref_Xb(:)'; ref_Yb(:)'; ref_Zb(:)'] + repmat(t, [1, 100]);

next_Xb = next_b(1, :);
next_Yb = next_b(2, :);
next_Zb = next_b(3, :);

next_Xo1 = [2.6, 3.7, 4.7; 2.8, 3.6, 4.5; 2.5, 3.5, 4.6];
next_Yo1 = [2.6, 2.7, 2.75; 3.4, 3.45, 3.5; 5.05, 5.10, 5.15];
next_Zo1 = [2.9, 2.9, 2.9; 2.9, 2.9, 2.9; 2.9, 2.9, 2.9];

next_Xo2 = [7.6, 8.7, 9.7; 7.8, 8.6, 9.5; 7.5, 8.5, 9.6];
next_Yo2 = [5.6, 5.7, 5.75; 6.4, 6.45, 6.5; 8.05, 8.10, 8.15];
next_Zo2 = [3.9, 3.9, 3.9; 3.9, 3.9, 3.9; 3.9, 3.9, 3.9];

% figure, hold on;
% plot3(next_Xb(:), next_Yb(:), next_Zb(:), 'ro'); hold on;
% plot3(next_Xo1(:), next_Yo1(:), next_Zo1(:), 'go'); hold on;
% plot3(next_Xo2(:), next_Yo2(:), next_Zo2(:), 'go'); hold on;
```

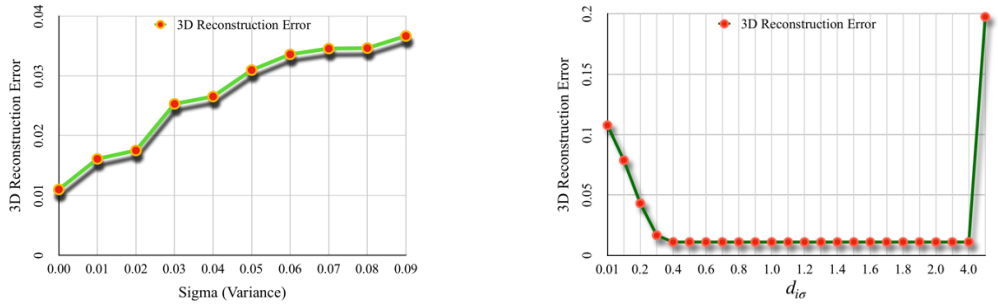
```
next_X01 = next_X01'; next_Y01 = next_Y01'; next_Z01 = next_Z01';
next_X02 = next_X02'; next_Y02 = next_Y02'; next_Z02 = next_Z02';

next_X = [next_Xb, next_X01(:)', next_X02(:)'];
next_Y = [next_Yb, next_Y01(:)', next_Y02(:)'];
next_Z = [next_Zb, next_Z01(:)', next_Z02(:)'];
%% 3. generate a synthetic image for the reference frame and next frame.
%some K matrix
fx = 100; fy = 100; cx = 240; cy = 320;
K = [fx, 0, cx; 0, fy, cy; 0, 0, 1];

% image point for the reference image
ref_img = K*[ref_X;ref_Y; ref_Z];
ref_img = ref_img./repmat(ref_img(3, :), [3, 1]);

% image point for the next image
next_img = K*[next_X; next_Y; next_Z];
next_img = next_img./repmat(next_img(3, :), [3, 1]);

%plot the image points
figure, plot(ref_img(1, :), ref_img(2, :), 'k.'); hold on;
plot(ref_img(1, 101:118), ref_img(2, 101:118), 'ro');

figure, plot(next_img(1, :), next_img(2, :), 'k.'); hold on;
plot(next_img(1, 101:118), next_img(2, 101:118), 'ro');

%% 4. Now define the neighbors based on the reference image distance
N = 118; %total number of anchor node.
topK = 22; %vary form 1 to N
[persuperpixelKNNid, persuperpixelwik] = givemeKNNforConcept(ref_img, N, topK);

%% 5. Perform ARAP optimization
%dvariance = ones(N, 1);
%lb = ref_Z' - dvariance; % lower bound on the variables, this works
%ub = ref_Z' + dvariance; % upper bound on the variables
lb = zeros(N, 1); %this also works
ub = []; %this also works
Aeq = []; % equality constraint
Beq = [];
A = []; % inequality constraint
B = [];
```

```
do = ones(N, 1)/N; %variable initialization

%optimization options
options = optimoptions('fmincon', 'Algorithm', 'sqp', 'Display', 'iter-detailed', 'MaxIter', 400, 'Max-
FunEvals', 300000, 'PlotFcns', @optimplotfval);
ref3D = [ref_X; ref_Y; ref_Z];
next3D = inv(K)*next_img;

disp('Optim');
[depthVal, cost] = fmincon(@(d)objectiveFunctionConceptARAP(d, ref3D, next3D, persuperpixelKN-
Nid, persuperpixelwik), do, A, B, Aeq, Beq, lb, ub, [], options);

output3D = zeros(3, N);
for i = 1:N
    output3D(:, i) = depthVal(i)*next3D(:, i);
end

figure,
plot3(next_X(:), next_Y(:), next_Z(:), 'r.'); hold on;
plot3(output3D(1, :), output3D(2, :), output3D(3, :), 'go');

 %% error estimation
gt_3D = [next_X(:)'; next_Y(:)'; next_Z(:)'];
es_3D = [output3D(1, :); output3D(2, :); output3D(3, :)];
error = norm(es_3D - gt_3D, 'fro')/norm(gt_3D, 'fro');
fprintf('Relative Error = %f \n', error)
```

E.2 Statistical Evaluation

We performed few more experiments to better understand the behavior of the algorithm under different input condition and variable initialization.

(a) Performance of the algorithm under noisy 3D initialization for the reference frame: This experiment is conducted to study the sensitivity of the method to noisy initialization. Fig. (E.3a) show the change in the 3D reconstruction accuracy with the variation in the level of noise from 1% to 9%. The Gaussian noise is introduced using randn() function of MATLAB and the result is documented for example(*E*.1.2) after repeating the experiment 10 times and taking its average value. We observe that algorithm can provide unsettling results when the noise becomes very large



(a) **(b)**

Figure E.3: (a) 3D reconstruction results for the next frame with different levels of Gaussian noise in the reference frame coordinate initialization. The curve is generated using the second synthetic experiment with K-NN as 117 (topK = 117) *i.e.* fully connected graph. (b) Variation in the performance with the change in the $di\sigma$ values for synthetic example 2.

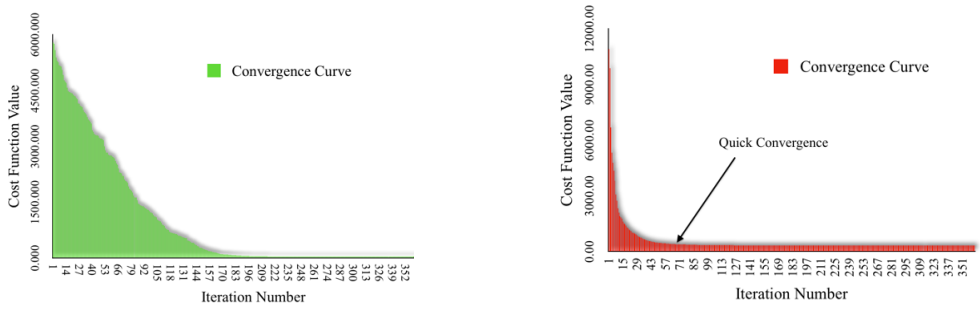
(b) Performance of the algorithm under restricted isometry constraint ($di\sigma$) with Φ^{arap} objective function: While minimizing the as rigid as possible objective function under the $|\tilde{d}_i - d_i| < di\sigma$ constraint, we restrict the convergence trust region of the optimization. This constraint makes the algorithm works extremely well —both in terms of timing and accuracy, if the prior knowledge about the deformation that the scene may undergo is known a priori. Fig.(E.3b) show the reconstruction accuracy as a function of $di\sigma$. Clearly, if we have the the approximate knowledge about the scene scene transformation, we can get high accuracy in less computation time. See Fig.(E.4b) which illustrates the quick convergence by using this constraint under proper the values of $di\sigma$.

(c) Nature of convergence of the proposed as rigid as possible optimization

- *Without restricted isometry constraint*: As rigid as possible minimization Φ^{arap} under the constraint $\tilde{d}_i > 0$ is a good enough constraint to provide acceptable results. However, it may take considerable number of iteration to do so. Fig.(E.4a) show the convergence curve
- *With restricted isometry constraint*: Employing the approximate bound on the deformation that the scene may undergo in the next time instance can help fast convergence with similar accuracy. Fig.(E.4b) show that the same accuracy can be achieved in 60 iteration.

E.3 Discussion

(a) Why do we choose geometric approach to initialize our algorithm on MPI dataset [24]? We tested the LKVO network [190] on MPI Sintel dataset which is one of the recent state-of-the-art network for dense depth estimation on KITTI dataset. Unfortunately, the network provides some unsettling results on MPI Sintel dataset. Fig.(E.5) show some results obtained by using this network after training



(a) (b)

Figure E.4: (a) Convergence curve of the cost function using SQP implementation of MATLAB toolbox for the second example. (b) Quick convergence with similar accuracy on the same example can be achieved by using isometry constraint.

Figure E.5: Depth results using the recent state-of-the-art LKVO network [190] after training on MPI Sintel dataset.

on the clean class of MPI Sintel dataset. The training parameters used for the this network on MPI Sintel dataset is provided below.

(b) What do we gain or lose by our motion free approach?
Estimating all kinds of conceivable motion in a complex dynamic scene from images is a challenging task, in that respect our method provides an alternative way to achieve per pixel depth without estimating any 3D motion. However, in achieving this we are allowing the gauge freedom between the frames (temporal relations in 3D over frames).

E.4 LKVO network flags and parameters used to train on MPI Sintel

We used the clean category to train the network. Here are the list of all the parameters and their default values

- dataroot, required=True, help='path to images (should have subfolders trainA, trainB, valA, valB, etc)'
- batchSize, type=int, default=1, help='input batch size'
- imH, type=int, default=128, help='imH'
- imW, type=int, default=416, help='imW'
- max_lk_iter_num, type=int, default=10, help='maximum iteration for LK update'
- gpu_ids, type=str, default='0', help='gpu ids: e.g. 0 0,1,2, 0,2. use -1 for CPU'
- name, type=str, default='experiment_name', help='name of the experiment. It decides where to store samples and models'
- nThreads, default=2, type=int, help='# threads for loading data'
- checkpoints_dir, type=str, default='./checkpoints', help='models are saved here'
- display_winsize, type=int, default=256, help='display window size'
- display_id, type=int, default=1, help='window id of the web display'
- display_port, type=int, default=8097, help='visdom port of the web display'
- display_single_pane_ncols, type=int, default=0, help='if positive, display all images in a single visdom web panel with certain number of images per row.'
- lk_level, type=int, default=1

- use_ssim, default=False, action='store_true', help='use ssim loss'
- smooth_term, type=str, default='lap', help='smoothness term type, choose between lap, 1st, 2nd'
- lambda_S, type=float, default=.01, help='smoothness cost weight'
- lambda_E, type=float, default=.01, help='explainable mask regulariation cost weight'
- epoch_num, type=int, default=20, help='number of epochs for training'
- display_freq, type=int, default=100, help='frequency of showing training results on screen'
- print_freq, type=int, default=10, help='frequency of showing training results on console'
- save_latest_freq, type=int, default=5000, help='frequency of saving the latest results'
- phase, type=str, default='train', help='train, val, test, etc'
- which_epoch, type=int, default=-1, help='which epoch to load? set to epoch number, set -1 to train from scratch'
- niter, type=int, default=100, help='# of iter at starting learning rate'
- niter_decay, type=int, default=100, help='# of iter to linearly decay learning rate to zero'
- beta1, type=float, default=0.5, help='momentum term of adam'
- lr, type=float, default=0.0002, help='initial learning rate for adam'
- no_html, action='store_true', help='do not save intermediate training results to [opt.checkpoints_dir]/[opt.name]/web/'

The altered parameters for training are listed below:

E.4.1 Training posenet, modified parameters:

1. –dataroot= formatted/data
2. –checkpoints_dir= checkpoints
3. –which_epoch= -1
4. –save_latest_freq= 1000
5. –batchSize = 1
6. –display_freq =50

7. –name= posenet
8. –lambda_S= 0.01
9. –smooth_term= 2nd
10. –use_ssim
11. –display_port =8009

E.4.2 Fine tuning the LKVO network, modified parameters

1. –dataroot= formatted/data
2. –checkpoints_dir= checkpoints
3. –which_epoch= -1
4. –save_latest_freq= 1000
5. –batchSize = 1
6. –display_freq =50
7. –name= finetune
8. –lk_level = 1
9. –lambda_S= 0.01
10. –smooth_term= 2nd
11. –use_ssim
12. –display_port =8009
13. –epoch_num = 10
14. –lr = 0.00001

References

[1] P.-A. Absil, R. Mahony, and R. Sepulchre. Riemannian geometry of grassmann manifolds with a view on algorithmic computation. *Acta Applicandae Mathematicae*, 80(2):199–220, 2004.

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

[3] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.

[4] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, 2011.

[5] I. Akhter, Y. Sheikh, and S. Khan. In defense of orthonormality constraints for nonrigid structure from motion. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 1534–1541, 2009.

[6] I. Akhter, Y. Sheikh, S. Khan, and T. Kanade. Nonrigid structure from motion in trajectory space. In *Advances in Neural Information Processing Systems*, pages 41–48, 2008.

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

[8] 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*, 33(7):1442–1456, 2011.

[9] D. Alessio, A. Henrik, J. Sebastian N, and S. Yaser. Non-rigid structure from motion challenge. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Disney Research, 2017.

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

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

[12] C. Bailer, B. Taetz, and D. Stricker. Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4015–4023, 2015.

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

[14] 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)*, volume 30, page 75. ACM, 2011.

[15] H. Y. Benson and D. F. Shanno. Interior-point methods for nonconvex nonlinear programming: cubic regularization. *Computational optimization and applications*, 58(2):323–346, 2014.

[16] H. Y. Benson, R. J. Vanderbei, and D. F. Shanno. Interior-point methods for nonconvex nonlinear programming: Filter methods and merit functions. *Computational Optimization and Applications*, 23(2):257–272, 2002.

[17] A. Blake. The least-disturbance principle and weak constraints. *Pattern Recognition Letters*, 1(5-6):393–399, 1983.

[18] A. Blake and A. Zisserman. *Visual reconstruction*. MIT press, 1987.

[19] 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*, 3(1):1–122, 2011.

[20] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[21] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3D shape from image streams. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 690–696, 2000.

[22] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shape from image streams. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 690–696. IEEE, 2000.

[23] D. Burago, Y. Burago, and S. Ivanov. *A course in metric geometry*, volume 33. American Mathematical Society Providence, 2001.

[24] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *European Conference on Computer Vision*, pages 611–625. Springer, 2012.

[25] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995.

[26] R. Cabral, F. De la Torre, J. P. Costeira, and A. Bernardino. Unifying nuclear norm and bilinear factorization approaches for low-rank matrix decomposition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2488–2495, 2013.

[27] J.-F. Cai, E. J. Candes, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

[28] 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*, pages 1896–1902, 2009.

[29] A. Chatterjee and V. Madhav Govindu. Efficient and robust large-scale rotation averaging. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 521–528, 2013.

[30] K. Chen, H. Dong, and K.-S. Chan. Reduced rank regression via adaptive nuclear norm penalization. *Biometrika*, 100(4):901–920, 2013.

[31] Q. Chen and V. Koltun. Full flow: Optical flow estimation by global optimization over regular grids. *arXiv preprint arXiv:1604.03513*, 2016.

[32] A. Chhatkuli, D. Pizarro, T. Collins, and A. Bartoli. Inextensible non-rigid shape-from-motion by second-order cone programming. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1719–1727, 2016.

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

[34] J. Cho, M. Lee, and S. Oh. Complex non-rigid 3d shape recovery using a procrustean normal distribution mixture model. *International Journal of Computer Vision*, pages 1–21, 2015.

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

[36] D. J. Crandall, A. Owens, N. Snavely, and D. P. Huttenlocher. Sfm with mrfs: Discrete-continuous optimization for large-scale structure from motion. *IEEE transactions on pattern analysis and machine intelligence*, 35(12):2841–2853, 2013.

[37] K. Crane. *Conformal geometry processing*. California Institute of Technology, 2013.

[38] K. Crane. Discrete differential geometry: An applied introduction. 2015.

[39] K. Crane, F. De Goes, M. Desbrun, and P. Schröder. Digital geometry processing with discrete exterior calculus. In *ACM SIGGRAPH 2013 Courses*, page 7. ACM, 2013.

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

[41] 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.

[42] Y. Dai, H. Li, and M. He. A simple prior-free method for non-rigid structure-from-motion factorization. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 2018–2025, 2012.

[43] Y. Dai, H. Li, and M. He. Projective multiview structure and motion from element-wise factorization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(9):2238–2251, 2013.

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

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

[46] A. Del Bue, F. Smeraldi, and L. Agapito. Non-rigid structure from motion using ranklet-based tracking and non-linear optimization. *Image and Vision Computing*, 25(3):297–310, 2007.

[47] A. Del Bue, J. Xavier, L. Agapito, and M. Paladini. Bilinear factorization via augmented lagrange multipliers. In *European Conference on Computer Vision*, pages 283–296. Springer, 2010.

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

[49] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

[50] A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM journal on Matrix Analysis and Applications*, 20(2):303–353, 1998.

[51] Y. C. Eldar, D. Needell, and Y. Plan. Uniqueness conditions for low-rank matrix recovery. *Applied and Computational Harmonic Analysis*, 33(2):309–314, 2012.

[52] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2790–2797. IEEE, 2009.

[53] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2765–2781, 2013.

[54] A. Eriksson, C. Olsson, F. Kahl, and T.-J. Chin. Rotation averaging and strong duality. *arXiv preprint arXiv:1705.01362*, 2017.

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

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

[57] A. W. Fitzgibbon and A. Zisserman. Multibody structure and motion: 3-d reconstruction of independently moving objects. In *Computer Vision-ECCV 2000*, pages 891–906. Springer, 2000.

[58] K. Fragkiadaki, M. Salas, P. Arbeláez, and J. Malik. Grouping-based low-rank trajectory completion and 3d reconstruction. In *Advances in Neural Information Processing Systems*, pages 55–63, 2014.

[59] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig. Virtual worlds as proxy for multi-object tracking analysis. *arXiv preprint arXiv:1605.06457*, 2016.

[60] S. Gaïffas and G. Lecué. Weighted algorithms for compressed sensing and matrix completion. *arXiv preprint arXiv:1107.1638*, 2011.

[61] 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*, pages 3884–3892, 2017.

[62] R. Garg, B. V. Kumar, G. Carneiro, and I. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision*, pages 740–756. Springer, 2016.

[63] 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*, pages 300–314. Springer, 2011.

[64] R. Garg, A. Roussos, and L. Agapito. Dense variational reconstruction of non-rigid surfaces from monocular video. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 1272–1279, 2013.

[65] R. Garg, A. Roussos, and L. Agapito. Dense variational reconstruction of non-rigid surfaces from monocular video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1272–1279, 2013.

[66] R. Garg, A. Roussos, and L. Agapito. A variational approach to video registration with subspace constraints. *International Journal of Computer Vision*, 104(3):286–314, 2013.

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

[68] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The KITTI dataset. *Int. J. Rob. Res.*, 32(11):1231–1237, Sept. 2013.

[69] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, page 7, 2017.

[70] P. Gotardo and A. Martinez. Non-rigid structure from motion with complementary rank-3 spaces. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 3065–3072, 2011.

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

[72] P. F. Gotardo and A. M. Martinez. Kernel non-rigid structure from motion. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 802–809. IEEE, 2011.

[73] P. F. Gotardo and A. M. Martinez. Non-rigid structure from motion with complementary rank-3 spaces. 2011.

[74] 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.

[75] V. M. Govindu. Lie-algebraic averaging for globally consistent motion estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 684–691. IEEE, 2004.

[76] V. M. Govindu. Robustness in motion averaging. In *Asian Conference on Computer Vision*, pages 457–466. Springer, 2006.

[77] V. M. Govindu. Motion averaging in 3d reconstruction problems. In *Riemannian Computing in Computer Vision. To Appear*. Springer, 2015.

[78] W. E. L. Grimson. *From images to surfaces: A computational study of the human early visual system*. MIT press, 1981.

[79] S. Gu, L. Zhang, W. Zuo, and X. Feng. Weighted nuclear norm minimization with application to image denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2862–2869, 2014.

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

[81] O. C. Hamsici, P. F. Gotardo, and A. M. Martinez. Learning spatially-smooth mappings in non-rigid structure from motion. In *European Conference on Computer Vision*, pages 260–273. Springer, 2012.

[82] 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*, pages 3120–3127. IEEE, 2013.

[83] 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*, pages 17–32. Springer, 2014.

[84] R. Hartley, J. Trumpf, Y. Dai, and H. Li. Rotation averaging. *International journal of computer vision*, 103(3):267–305, 2013.

[85] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[86] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[87] R. I. Hartley. In defense of the eight-point algorithm. *IEEE Transactions on pattern analysis and machine intelligence*, 19(6):580–593, 1997.

[88] R. I. Hartley and P. Sturm. Triangulation. *Computer vision and image understanding*, 68(2):146–157, 1997.

[89] G. E. Hinton. Relaxation and its role in vision. 1977.

[90] M. Hornacek, F. Besse, J. Kautz, A. Fitzgibbon, and C. Rother. Highly overparameterized optical flow using patchmatch belief propagation. In *European Conference on Computer Vision*, pages 220–234. Springer, 2014.

[91] 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*, pages 140–149, 2015.

[92] T. Igarashi, T. Moscovich, and J. F. Hughes. As-rigid-as-possible shape manipulation. In *ACM transactions on Graphics*, volume 24, pages 1134–1141. ACM, 2005.

[93] 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.

[94] P. Ji, H. Li, Y. Dai, and I. D. Reid. " maximizing rigidity" revisited: A convex programming approach for generic 3d shape reconstruction from multiple perspective views. In *IEEE International Conference on Computer Vision*, pages 929–937, 2017.

[95] F. Kahl and D. Henrion. Globally optimal estimates for geometric reconstruction problems. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 978–985. IEEE, 2005.

[96] K. Karsch, C. Liu, and S. B. Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *IEEE transactions on pattern analysis and machine intelligence*, 36(11):2144–2158, 2014.

[97] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE transactions on pattern analysis and machine intelligence*, 28(10):1568–1583, 2006.

[98] C. Kong and S. Lucey. Prior-less compressible structure from motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4123–4131, 2016.

[99] S. Kumar. Jumping manifolds: Geometry aware dense non-rigid structure from motion. *arXiv preprint arXiv:1902.01077*, 2019.

[100] S. Kumar. Jumping manifolds: Geometry aware dense non-rigid structure from motion supplementary material. 2019.

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

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

[103] S. Kumar. Non-rigid structure from motion: Prior-free factorization method revisited supplementary material. 2020.

[104] S. Kumar, A. Cherian, Y. Dai, and H. Li. Scalable dense non-rigid structure-from-motion: A grassmannian perspective supplementary material.

[105] S. Kumar, A. Cherian, Y. Dai, and H. Li. Scalable dense non-rigid structure-from-motion: A grassmannian perspective. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 254–263, 2018.

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

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

[108] S. Kumar, Y. Dai, and H. Li. Spatial-temporal union of subspaces for multi-body nrsfm: Supplementary material.

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

[110] S. Kumar, Y. Dai, and H. Li. Multi-body non-rigid structure-from-motion supplementary material. 2016.

[111] S. Kumar, Y. Dai, and H. Li. Monocular dense 3d reconstruction of a complex dynamic scene from two perspective frames. In *IEEE International Conference on Computer Vision*, pages 4649–4657, Oct 2017.

[112] S. Kumar, Y. Dai, and H. Li. Supplementary material: Monocular dense 3d reconstruction of a complex dynamic scene from two perspective frames. 2017.

[113] S. Kumar, Y. Dai, and H. Li. Superpixel soup: Monocular dense 3d reconstruction of a complex dynamic scene. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[114] S. Kumar, A. Dewan, and K. M. Krishna. A bayes filter based adaptive floor segmentation with homography and appearance cues. In *Proceedings of the Eighth Indian Conference on Computer Vision, Graphics and Image Processing*, page 54. ACM, 2012.

[115] S. Kumar, R. S. Ghorakavi, Y. Dai, and H. Li. Dense depth estimation of a complex dynamic scene without explicit 3d motion estimation. *arXiv preprint arXiv:1902.03791*, 2019.

[116] S. Kumar, R. S. Ghorakavi, Y. Dai, and H. Li. A motion free approach to dense depth estimation in complex dynamic scene. *CoRR*, abs/1902.03791, 2019.

[117] S. Kumar, M. S. Karthik, and K. M. Krishna. Markov random field based small obstacle discovery over images. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 494–500. IEEE, 2014.

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

[119] M. Lee, J. Cho, C.-H. Choi, and S. Oh. Procrustean normal distribution for non-rigid structure from motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1280–1287, 2013.

[120] M. Lee, J. Cho, C.-H. Choi, and S. Oh. Procrustean normal distribution for non-rigid structure from motion. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 1280–1287, 2013.

[121] M. Lee, J. Cho, and S. Oh. Consensus of non-rigid reconstructions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4670–4678, 2016.

[122] H. Li. Multi-view structure computation without explicitly estimating motion. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2777–2784. IEEE, 2010.

[123] H. Li, B. Adams, L. J. Guibas, and M. Pauly. Robust single-view geometry and motion reconstruction. In *ACM Transactions on Graphics*, volume 28, page 175, 2009.

[124] 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*, 35(1):171–184, 2013.

[125] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293(5828):133–135, 1981.

[126] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms, MA Fischler and O. Firschein, eds*, pages 61–62, 1987.

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

[128] C. Lu, J. Tang, S. Yan, and Z. Lin. Nonconvex nonsmooth low rank minimization via iteratively reweighted nuclear norm. *IEEE Transactions on Image Processing*, 25(2):829–839, 2016.

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

[130] L. Magerand and A. Del Bue. Practical projective structure from motion (p2sfm). In *Proceedings of the IEEE International Conference on Computer Vision*, pages 39–47, 2017.

[131] E. Malis and M. Vargas. *Deeper understanding of the homography decomposition for vision-based control*. PhD thesis, INRIA, 2007.

[132] J. Milnor and J. D. Stasheff. *Characteristic Classes.(AM-76)*, volume 76. Princeton university press, 2016.

[133] R. A. Newcombe and A. J. Davison. Live dense reconstruction with a single moving camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1498–1505. IEEE, 2010.

[134] R. A. Newcombe, D. Fox, and S. M. Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 343–352, 2015.

[135] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. Dtam: Dense tracking and mapping in real-time. In *2011 international conference on computer vision*, pages 2320–2327. IEEE, 2011.

[136] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 849–856. MIT Press, 2002.

[137] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856, 2002.

[138] D. Nistér. An efficient solution to the five-point relative pose problem. *IEEE transactions on pattern analysis and machine intelligence*, 26(6):756–770, 2004.

[139] J. Noraky and V. Sze. Depth estimation of non-rigid objects for time-of-flight imaging. In *IEEE International Conference on Image Processing*, pages 2925–2929. IEEE, 2018.

[140] T.-H. Oh, Y.-W. Tai, J.-C. Bazin, H. Kim, and I. S. Kweon. Partial sum minimization of singular values in robust pca: Algorithm and applications. *IEEE transactions on pattern analysis and machine intelligence*, 38(4):744–758, 2016.

[141] K. E. Ozden, K. Schindler, and L. Van Gool. Multibody structure-from-motion in practice. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(6):1134–1141, 2010.

[142] 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 *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 2898–2905, 2009.

[143] 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*, 96(2):252–276, 2012.

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

[145] K. B. Petersen, M. S. Pedersen, et al. The matrix cookbook. *Technical University of Denmark*, 7(15):510, 2008.

[146] M. J. Powell. A fast algorithm for nonlinearly constrained optimization calculations. In *Numerical analysis*, pages 144–157. Springer, 1978.

[147] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3282–3289. IEEE, 2012.

[148] V. Rabaud and S. Belongie. Re-thinking non-rigid structure from motion. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[149] R. Ranftl, V. Vineet, Q. Chen, and V. Koltun. Dense monocular depth estimation in complex dynamic scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4058–4066, 2016.

[150] C. Russell, J. Fayad, and L. Agapito. Dense non-rigid structure from motion. In *Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, pages 509–516. IEEE, 2012.

[151] C. Russell, R. Yu, and L. Agapito. Video pop-up: Monocular 3d reconstruction of dynamic scenes. In *European Conference on Computer Vision*, pages 583–598. 2014.

[152] C. Russell, R. Yu, and L. Agapito. Video pop-up: Monocular 3d reconstruction of dynamic scenes. In *European Conference on Computer Vision*, pages 583–598. Springer, 2014.

[153] J. L. Schonberger and J.-M. Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016.

[154] Y. Sheng, P. Willis, G. G. Castro, and H. Ugail. Facial geometry parameterisation based on partial differential equations. *Mathematical and Computer Modelling*, 54(5):1536–1548, 2011.

[155] T. Simon, J. Valmadre, I. Matthews, and Y. Sheikh. Separable spatiotemporal priors for convex reconstruction of time-varying 3d point clouds. In *European Conference on Computer Vision*, pages 204–219. 2014.

[156] K. N. Snavely. Scene reconstruction and visualization from internet photo collections. 2009.

[157] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM transactions on graphics (TOG)*, volume 25, pages 835–846. ACM, 2006.

[158] N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *International journal of computer vision*, 80(2):189–210, 2008.

[159] N. Snavely, S. M. Seitz, and R. Szeliski. Skeletal graphs for efficient structure from motion. In *CVPR*, volume 1, page 2, 2008.

[160] N. Snavely, I. Simon, M. Goesele, R. Szeliski, and S. M. Seitz. Scene reconstruction and visualization from community photo collections. *Proceedings of the IEEE*, 98(8):1370–1390, 2010.

[161] O. Sorkine and M. Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, volume 4, 2007.

[162] N. Srebro and T. Jaakkola. Weighted low-rank approximations. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 720–727, 2003.

[163] G. Strang, G. Strang, G. Strang, and G. Strang. *Introduction to linear algebra*, volume 3. Wellesley-Cambridge Press Wellesley, MA, 1993.

[164] 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*, pages 8934–8943, 2018.

[165] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[166] J. Taylor, A. D. Jepson, and K. N. Kutulakos. Non-rigid structure from locally-rigid motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2761–2768. IEEE, 2010.

[167] S. Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, 2002.

[168] C. Tomasi. Pictures and trails: a new framework for the computation of shape and motion from perspective image sequences. Technical report, Cornell University, 1993.

[169] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *Int'l J. Computer Vision*, 9(2):137–154, 1992.

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

[171] P. H. Torr and D. W. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *International Journal of Computer Vision*, 24(3):271–300, 1997.

[172] L. Torresani and A. Hertzmann. Automatic non-rigid 3D modeling from video. In *Proc. European Conf. Computer Vision*, pages 299–312, 2004.

[173] L. Torresani, A. Hertzmann, and C. Bregler. Learning non-rigid 3d shape from 2d motion. In *Advances in Neural Information Processing Systems*, pages 1555–1562, 2004.

[174] L. Torresani, A. Hertzmann, and C. Bregler. Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(5):878–892, 2008.

[175] L. Torresani, A. Hertzmann, and C. Bregler. Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. *IEEE transactions on pattern analysis and machine intelligence*, 30(5):878–892, 2008.

[176] L. Torresani, D. B. Yang, E. J. Alexander, and C. Bregler. Tracking and modeling non-rigid objects with rank constraints. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 493–500, 2001.

[177] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment—a modern synthesis. In *International workshop on vision algorithms*, pages 298–372. Springer, 1999.

[178] S. Ullman. The interpretation of structure from motion. *Proceedings of the Royal Society of London B: Biological Sciences*, 203(1153):405–426, 1979.

[179] S. Upadhyay, S. Kumar, and K. M. Krishna. Crf based frontier detection using monocular camera. In *Proceedings of the 2014 Indian Conference on Computer Vision Graphics and Image Processing*, page 69. ACM, 2014.

[180] J. Valmadre, S. Sridharan, S. Denman, C. Fookes, and S. Lucey. Closed-form solutions for low-rank non-rigid reconstruction. In *Digital Image Computing: Techniques and Applications (DICTA), 2015 International Conference on*, pages 1–6. IEEE, 2015.

[181] N. van der Aa, X. Luo, G. Giezeman, R. Tan, and R. Veltkamp. Umpm benchmark: A multi-person dataset with synchronized video and motion capture data for evaluation of articulated human motion and interaction. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1264–1269, Nov 2011.

[182] N. Van der Aa, X. Luo, G.-J. Giezeman, R. T. Tan, and R. C. Veltkamp. Umpm benchmark: A multi-person dataset with synchronized video and motion capture data for evaluation of articulated human motion and interaction. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1264–1269. IEEE, 2011.

[183] A. Varol, M. Salzmann, P. Fua, and R. Urtasun. A constrained latent variable model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2248–2255. Ieee, 2012.

[184] A. Varol, M. Salzmann, E. Tola, and P. Fua. Template-free monocular reconstruction of deformable surfaces. In *International Conference on Computer Vision*, pages 1811–1818. IEEE, 2009.

[185] A. Vedaldi and S. Soatto. Quick shift and kernel methods for mode seeking. In *European Conference on Computer Vision*, pages 705–718. Springer, 2008.

[186] S. Vicente and L. Agapito. Soft inextensibility constraints for template-free non-rigid reconstruction. In *European conference on computer vision*, pages 426–440. Springer, 2012.

[187] R. Vidal. Subspace clustering. *IEEE Signal Processing Magazine*, 28(2):52–68, 2011.

[188] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (gpca). *IEEE transactions on pattern analysis and machine intelligence*, 27(12):1945–1959, 2005.

[189] 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.

[190] C. Wang, J. Miguel Buenaposada, R. Zhu, and S. Lucey. Learning depth from monocular videos using direct methods. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2018.

[191] R. Wang, S. Shan, X. Chen, and W. Gao. Manifold-manifold distance with application to face recognition based on image set. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[192] X. Wang, M. Salzmann, F. Wang, and J. Zhao. Template-free 3d reconstruction of poorly-textured nonrigid surfaces. In *European Conference on Computer Vision*, pages 648–663. Springer, 2016.

[193] W. Whiteley. Rigidity and scene analysis. 2004.

[194] R. Williamson and L. Janos. Constructing metrics with the heine-borel property. *Proceedings of the American Mathematical Society*, 100(3):567–573, 1987.

[195] J. Xiao, J. Chai, and T. Kanade. A closed-form solution to non-rigid shape and motion recovery. *Int'l J. Computer Vision*, 67(2):233–246, 2006.

[196] J. Xiao, J.-x. Chai, and T. Kanade. A closed-form solution to non-rigid shape and motion recovery. In *European conference on computer vision*, pages 573–587. Springer, 2004.

[197] J. Xiao and T. Kanade. Non-rigid shape and motion recovery: Degenerate deformations. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 668–675, 2004.

[198] C. Xu, T. Wang, J. Gao, S. Cao, W. Tao, and F. Liu. An ordered-patch-based image classification approach on the image grassmannian manifold. *IEEE transactions on neural networks and learning systems*, 25(4):728–737, 2014.

[199] C. You, C.-G. Li, D. P. Robinson, and R. Vidal. Oracle based active set algorithm for scalable elastic net subspace clustering. *arXiv preprint arXiv:1605.02633*, 2016.

[200] 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*, pages 918–926. IEEE, 2015.

[201] L. Zappella, A. Del Bue, X. Lladó, and J. Salvi. Joint estimation of segmentation and structure from motion. *Computer Vision and Image Understanding*, 117(2):113–129, 2013.

[202] Z. Zha, X. Zhang, Y. Wu, Q. Wang, Y. Bai, and L. Tang. Analyzing the weighted nuclear norm minimization and nuclear norm minimization based on group sparse representation. *arXiv preprint arXiv:1702.04463*, 2017.

[203] D. Zhang, Y. Hu, J. Ye, X. Li, and X. He. Matrix completion by truncated nuclear norm regularization. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2192–2199. IEEE, 2012.

[204] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, page 7, 2017.

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

[206] Y. Zhu, D. Huang, F. De La Torre, and S. Lucey. Complex non-rigid motion 3d reconstruction by union of subspaces. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 1542–1549, June 2014.

[207] Y. Zhu and S. Lucey. Convolutional sparse coding for trajectory reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):529–540, March 2015.

[208] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.