TOWARD 3D RECONSTRUCTION OF STATIC AND DYNAMIC OBJECTS

Enliang Zheng

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Computer Science.

Chapel Hill
2016

Approved by:

Jan-Michael Frahm

Enrique Dunn

Tamara L. Berg

Vladimir Jojic

Yaser Sheikh

ABSTRACT

Enliang Zheng: Toward 3D Reconstruction of Static and Dynamic Objects
(Under the direction of Jan-Michael Frahm and Enrique Dunn)

The goal of image-based 3D reconstruction is to construct a spatial understanding of the world from a collection of images. For applications that seek to model generic real-world scenes, it is important that the reconstruction methods used are able to characterize both static scene elements (*e.g.* trees and buildings) as well as dynamic objects (*e.g.* cars and pedestrians). However, due to many inherent ambiguities in the reconstruction problem, recovering this 3D information with accuracy, robustness, and efficiency is a considerable challenge. To advance the research frontier for image-based 3D modeling, this dissertation focuses on three challenging problems in static scene and dynamic object reconstruction.

We first target the problem of static scene depthmap estimation from crowd-sourced datasets (*i.e.* photos collected from the Internet). While achieving high-quality depthmaps using images taken under a controlled environment is already a difficult task, heterogeneous crowd-sourced data presents a unique set of challenges for multi-view depth estimation, including varying illumination and occasional occlusions. We propose a depthmap estimation method that demonstrates high accuracy, robustness, and scalability on a large number of photos collected from the Internet.

Compared to static scene reconstruction, the problem of dynamic object reconstruction from monocular images is fundamentally ambiguous when not imposing any additional assumptions. This is because having only a single observation of an object is insufficient for valid 3D triangulation, which typically requires concurrent observations of the object from multiple viewpoints. Assuming that dynamic objects of the same class (*e.g.* all the pedestrians walking on a sidewalk) move in a common path in the real world, we develop a method that estimates the 3D positions of the dynamic objects from unstructured monocular images. Experiments on both synthetic and real datasets illustrate the solvability of the problem and the effectiveness of our approach.

Finally, we address the problem of dynamic object reconstruction from a set of unsynchronized videos capturing the same dynamic event. This problem is of great interest because, due to the increased availability of portable capture devices, captures using multiple unsynchronized videos are common in the real world. To resolve the challenges that arises from non-concurrent captures and unknown temporal overlap among video streams, we propose a self-expressive dictionary learning framework, where the dictionary entries are defined as the collection of temporally varying structures. Experiments demonstrate the effectiveness of this approach to the previously unsolved problem.

ACKNOWLEDGEMENTS

My deepest gratitude is to my advisors Jan-Michael Frahm and Enrique Dunn. I have been amazingly fortunate to have advisors who gave me the guidance and encouragement when my steps faltered, and the freedom to explore on my own.

I would also like to thank my committee members, Tamara L. Berg, Vladimir Jojic, and Yaser Sheikh, for their feedback and advice.

Additionally, I would like to thank my labmates, as their company and discussion made my time more fruitful and enjoyable: Philip Ammirato, Akash Bapat, Sangwoo Cho, Marc Eder, Pierre Fite-Georgel, Yunchao Gong, Rohit Gupta, Shubham Gupta, Xufeng Han, Jared Heinly, Junpyo Hong, Yi-Hung Jen, Dinghuang Ji, Alex Keng, Hadi Kiapour, Hyo Jin Kim, Wei Liu, Jie Lu, Licheng Yu, Vicente Ordóñez-Román, David Perra, True Price, Rahul Raguram, Patrick Reynolds, Johannes Schönberger, Meng Tan, Joseph Tighe, Sirion Vittayakorn, Ke Wang, Yilin Wang, Yi Xu, and Hongsheng Yang.

I would like to thank my parents, who encouraged me to explore, learn and pursue for a PhD.

Finally, I would like to thank my wife, Lingling Zheng, for her understanding and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

GMST	Generalized minimum spanning tree
JOST	Joint object class sequencing and trajectory triangulation
MRF	Markov Random Field
MVDE	Multiview depthmap estimation
NCC	Normalized cross correlation
NRSFM	Non-rigid structure from motion
RANSAC	Random sample consensus
SfM	Structure from motion

CHAPTER 1: INTRODUCTION

Imagery records what the world looks like by projecting the 3D scene onto an image plane. However, the 3D information, which depicts the geometry of real objects, is lost during this capture process. Conversely, 3D information is key to many applications, such as augmented/virtual reality (Ventura and Höllerer, 2008), robots and autonomous car navigation (Endres et al., 2012), image-based rendering (Chen and Williams, 1993), and image enhancement (Zhang et al., 2014). Moreover, as additional information to RGB (red, green, and blue) colors, 3D information is leveraged to improve performance of many computer vision tasks such as object classfication/recognition (Gupta et al., 2013) and human pose estimation (Shotton et al., 2011). Therefore, there is a strong desire to recover reliable 3D information from 2D imagery.

3D information, when stored in computers, can be represented using 3D point clouds, 3D polygon meshes, or depthmaps. A 3D point cloud is a set of data points in three-dimensional space representing the external surface of an object, and it can be classified as either dense or sparse based on the number of points it contains per unit surface area. A 3D polygon mesh provides additional information in the form of the geometric topology among the 3D points. Finally, a depthmap is a dense field of depth values indicating the distance of the observed surface relative to a camera, rather than in a global coordinate system. In practice, different representations are adopted according to the requirements of the specific application.

3D reconstruction from imagery, defined as a process that recovers 3D information from 2D image colors, is a traditional problem in 3D computer vision. Unlike the task of computer graphics that renders 2D imagery from 3D geometry, the inverse process of 3D reconstruction from imagery is more challenging since attemping to recover lost information inevitably introduces more ambiguities. Though methods for 3D reconstruction have been widely studied and have undoubtedly improved over the last few decades, the field still remains a viable and open area of active research.

This dissertation primarily focuses on the problems of dense static scene reconstruction and sparse dynamic object reconstruction from 2D imagery.

Dense static scene reconstruction. To obtain the 3D information of a static scene, most existing works leverage 2D correspondences and available camera parameters for 3D triangulation. Though camera parameters can typically be estimated via structure from motion or offline calibration methods, obtaining 2D correspondences robustly from image colors still requires further exploration. The 2D correspondences are defined as pixels in different images that observe the same part of a 3D scene. Under the assumption of a Lambertian surface, these 2D correspondences share the same or similar appearances/colors, and hence they have high color consistency.

For each point in one image, finding its correspondence in another image involves searching for candidate pixels with the best color consistency along a line defined by the 3D geometry (called an epipolar line), and the positions of candidate pixels are determined by depth hypotheses generated in a valid range. Once the correspondence is found, the depth of the corresponding pixel is uniquely determined. However, estimating dense correspondences robustly is difficult since ambiguities arise in the case of repetitive textures, homogeneous color regions, or occlusions along the epipolar line.

Recently, there has been a growing interest in using the ever-growing domain of crowd-sourced data (*i.e.* Internet collected photos) for reconstruction, and the large amount of free data has inspired many applications, such as virtual photo tours (Snavely et al., 2006) and image enhancement (Zhang et al., 2014). With the non-controlled imagery as input, finding 2D correspondences based on colors is more challenging due to a diversity of factors, including heterogeneous resolution and scene illuminations, unstructured viewing geometry, scene content variability, and image registration errors. To address these issues, it is normally assumed in the massive number of images, there are a subset of images sharing similar image characteristics. Therefore, determining a suitable subset of images or pixels for correspondence search becomes essential (Goesele et al., 2007).

Dense reconstruction typically has very high computational complexity, since the traditional process involves exhaustive evaluations of a large number of depth hypotheses (Yang and Pollefeys, 2003). The increasing availability of crowd-sourced datasets has explicitly brought efficiency and scalability to the forefront of application requirements. Moreover, the high complexity of a method would impede its usage in less-powerful electronic devices such as smart phones. To this end, there is a compelling demand to develop efficient and scalable methods for dense reconstruction.

Sparse dynamic object reconstruction. While static scene reconstruction only focuses on static parts of a scene, it is of great interest to reconstruct the dynamic part of the scene as well. The problem of dynamic object reconstruction specifically aims at 3D reconstruction under the circumstance of non-concurrent image captures. To be more precise, the dynamic object is only observed by one image at each time instance. This poses an additional challenge compared to the problem of static scene reconstruction, since 3D triangulation becomes invalid and impossible with the single observation, even assuming 2D correspondences among non-concurrent images are correctly found. Given a unitary observation, it is only known that the 3D point lies somewhere along the viewing ray determined by the 2D meansurement and the camera pose, but the depth along the viewing ray cannot be easily computed. Primarily due to this intrinsic difficulty, the state of the art for dynamic object reconstruction falls far behind that of static scene reconstruction.

The problem of dynamic object reconstruction is fundamentally under-constrained and requires further assumptions. Many existing works make various assumptions on scene geometry, object motion, capture pattern, *etc*. For instance, most non-rigid structure from motion (NRSFM) methods assume the 3D shapes of deforming objects lie in a low-dimensional subspace, and hence any shape can be represented as a linear combination of K shape bases (Bregler et al., 2000; Torresani et al., 2008; Dai et al., 2014). Trajectory-based methods assume smooth motion of the dynamic objects across time (Akhter et al., 2009b). When developing methods for dynamic object reconstruction, in addition to making valid assumptions, having fewer but more general assumptions is vital to enable the methods to work more universally and robustly in real scenarios.

One particular formulation of dynamic object reconstruction is trajectory triangulation, which computes the trajectory of a dynamic 3D point given a set of unitary observations across time. Under the assumption of smooth object motion and available sequencing information (*i.e.* the temporal order of images being taken), existing methods can achieve accurate reconstruction results

(Park et al., 2010; Valmadre and Lucey, 2012). Although the assumption of smooth object motion is typically true for real dynamic objects, in practice easily obtaining the sequencing information and achieving high reconstruction accuracy cannot be satisfied simultaneously (Zhu et al., 2011; Valmadre and Lucey, 2012). The sequencing information essentially captures the physical constraint that a moving 3D point observed in two temporally close images will have a relatively small amount of spatial movement. In effect, it is this spatial proximity that is leveraged by the existing methods (Park et al., 2010; Valmadre and Lucey, 2012) for reconstruction. In contrast, our research focuses on 3D reconstruction of dynamic objects given no or only partial information of the spatial/temporal proximity.

1.1 Thesis Statement

The geometry of a scene can be recovered from uncontrolled image/video collections, through incorporating pixel-level image association into a scalable multiview stereo framework for dense reconstruction of static scene elements, and explicit modeling of spatio-temporal relations of unordered observations for sparse reconstruction of dynamic scene elements.

1.2 Outline of Contributions

This dissertation contributes significantly to advance the state-of-the-art techniques for the problems of static scene reconstruction and dynamic object reconstruction, and it builds on our published works (Zheng et al., 2014a,b, 2015).

PatchMatch Based Joint View Selection and Depth Estimation: Chapter 3 focuses on the problem of depthmap estimation using Internet collected photos. The non-controlled input imagery presents practical challenges such as heterogeneous scene illuminations and unstructured viewing geometry. Therefore, it is vital to determine a subset of images or pixels in the dataset for robust depth estimation. Moreover, the ever-increasing number of crowd-sourced datasets have explicitly brought efficiency and scalability to the forefront of application requirements.

To solve this problem, we propose a probabilistic framework for joint view selection and depth estimation at the pixel level. Our new method obtains more complete depthmaps compared to the state-of-the-art method for Internet collected photos (Goesele et al., 2007). To increase the efficiency and scalability, our framework seamlessly incorporates the PatchMatch scheme (Bleyer et al., 2011) to reduce the size of the depth hypothesis set. Also, the memory requirement of our framework scales linearly with respect to the number of source images, as opposed to exponentially (Strecha et al., 2006). Moreover, our method is designed to process each row or column of the reference image independently, enabling easy parallelization and GPU implementation.

Joint Object Class Sequencing and Trajectory Triangulation: Chapter 4 targets the problem of reconstructing the 3D positions of dynamic objects from a set of unstructured images. Each dynamic object is observed only once in the image collection, rendering traditional approaches for 3D triangulation for static scenes impossible. To tackle the fundamentally under-constrained problem, we assume that all of the objects of the same class (*e.g.* pedestrians or cars) move in a common path in 3D space. Then, our method estimates the 3D positions of the dynamic objects by triangulating the trajectory formed by all the objects moving in the common path.

To the best of our knowledge, no current methods have solved this challenging problem. Our method uses the object detection outputs as a general feature for each dynamic object, as opposed to typical image features such as points or edges. In solving the problem, recovering the sequencing information, which is defined as the topology of the trajectory in this specific problem (*i.e.* the information of spatial proximity), is vital for trajectory triangulation. We propose to jointly estimate the sequencing information and the 3D points, which is posed as minimizing a nonconvex function. To this end, we propose a novel discrete-continuous optimization approach based on the generalized minimum spanning tree (GMST).

Dynamic Object Reconstruction from Unsynchronized Videos: Chapter 5 also aims at the problem of dynamic object reconstruction, but using unsynchronized video streams as input. To handle this underconstrained problem, we observe that any shape at one time instance is a linear combination of the shapes at other time instances (self-expression), under the assumption of smooth object motion. The problem is then solved by learning a self-expressive dictionary, which is defined as a collection of temporally varied structures.

The main contribution of this chapter is solving the new problem of dynamic object reconstruction without temporal order information across video streams (also called sequencing information). This is contradictory to the existing works that strictly rely on available sequencing information (Park et al., 2010; Valmadre and Lucey, 2012). Moreover, to the best of our knowledge, we are the first to use the self-expression prior for dynamic object reconstruction. This prior has the potential to be used in the traditional non-rigid structure from motion (NRSFM) problem, where most existing methods use the assumption that any shape is a linear combination of K fixed shape bases (Dai et al., 2014; Bregler et al., 2000). In learning the dictionary, we propose a new efficient solver based on the alternating direction method of multipliers (ADMM) (Boyd et al., 2011).

Each of these contributions addresses the issue of 3D reconstruction from 2D imagery. Following Chapter 2, which covers related works, the next three chapters describe each method in detail, and Chapter 6 concludes the dissertation with potential extensions to our works and possible future research directions.

CHAPTER 2: RELATED WORK

3D reconstruction from 2D imagery has been studied extensively by many researchers in the computer vision community. In this section, we first review work on camera parameter estimation and then survey research related to static and dynamic object reconstruction.

2.1 Camera Parameter Estimation

Camera parameters are generally considered a prerequisite for 3D reconstruction, since they provide the geometric relationships between multiple cameras. Specifically, with this geometric information, the mapping from a 3D point to an image pixel can be uniquely determined. Camera parameters are seperated into two parts: the internal (intrinsic) camera parameters consist of a focal length, principle point, skew parameter, and radial distortion that convert the normalized coordinates to image coordiantes, and the external (extrinsic) part describes a camera's rotation and translation relative to a global coordinate system (Hartley and Zisserman, 2004).

Given the importance of camera parameters in computer vision tasks such as 3D reconstruction, many works have focused on estimating camera parameters, a process also called camera calibration. Earlier works for camera calibration required a calibration object such as a planar checkerboard to be seen by the cameras (Sturm and Maybank, 1999; Zhang, 2000; Bouguet, 2000), which imposes a significant constraint for practical applications. Thanks to the recent development of techniques in structure from motion (SfM) (Snavely et al., 2006, 2008; Wu, 2013; Wilson and Snavely, 2013; Heinly et al., 2014; Schönberger et al., 2015; Heinly et al., 2015; Heinly, 2015; Zheng and Wu, 2015), camera calibration can be achieved by simply leveraging 2D correspondences among multiple images.

Structure from motion is a pipeline that targets estimating the camera parameters of the images observing a common static scene. A typical pipeline includes the main steps of feature extraction

(Lowe, 2004; Rublee et al., 2011; Bay et al., 2008), inlier correspondence search (Raguram et al., 2013), camera pose estimation (Nistér, 2003; Kneip et al., 2011; Zheng et al., 2014c; Zheng and Wu, 2015), and bundle adjustment (Agarwal et al., 2010; Wu et al., 2011). Recent works in structure from motion have exhibited enough accuracy, efficiency, and robustness to be applicable in most real scenarios (Snavely et al., 2006; Wu, 2013).

2.2 Static Scene Reconstruction

As a main research subject in 3D computer vision, there are a large number of works addressing issues in static scene reconstruction. Early works mainly focus on depthmap estimation on binocular images (Boykov et al., 2001; Sun et al., 2002; Scharstein and Szeliski, 2002; Scharstein and Pal, 2007). In these works, two images are rectified so that correspondence estimation for a pixel in one image can be simplified to search along a single row of the other image. In contrast, multiview depthmap estimation (MVDE) uses multiple images to reduce the ambiguities in searching for correspondences. Moreover, the redundant information among the estimated depthmaps can be leveraged to filter out outlier depths. This section first discusses the most related works for multiview depthmap estimation and the associated issues such as robustness and efficiency, and then discusses briefly the methods for generating a consistent point cloud or mesh.

2.2.1 Multiview Depthmap Estimation

Handling occlusion is important in depthmap estimation, and the first methods for addressing occlusion emerged in two view stereo (Sun et al., 2002, 2005; Xiao et al., 2008). However, in these methods, the occluded pixel region is only marked with unknown depth due to the unavailable correspondence in another image.

In principle, the additional view redundancy available to MVDE can be leveraged to resolve occlusions. Kang et al. (2001) explicitly address occlusion in multi-baseline stereo by only using the subset of the heuristically selected overlapping cameras with the minimum matching cost. The heuristic provides occlusion robustness as long as there is a sufficient number of unoccluded views (typically 50%). Campbell et al. (2008) choose the best few depth hypotheses for each pixel, following with an Markov random field (MRF) optimization to determine a spatially consistent depthmap. Their method chooses source images based on spatial proximity of cameras. Strecha et al. (2004) handle occlusion in wide-baseline multi-view stereo by including visibility within a probabilistic model, where the depth smoothness is enforced on neighboring pixels according to the color gradient. The work by Strecha et al. (2004) is further extended in Strecha et al. (2006) where the depth and visibility are jointly modeled by hidden Markov random fields. In the work by Strecha et al. (2006), the memory used for visibility configuration of each pixel is 2^K, which grows exponentially with respect to the number of input images K. Hence, the approach is limited to very few images (three images in their evaluation). Gallup et al. (2008) present a variable-baseline and variable-resolution framework for MVDE, exploring the attainment of pixel-specific data associations for capture from approximately linear camera paths. While that work illustrates the benefits of fine-grained data association strategies in multi-view stereo, it does not easily generalize to irregularly captured datasets.

Given the redundant information among multiple depthmaps, lightweight depthmap fusion removes outlier depths by leveraging the mutual depth consistency among multiple depthmaps. Shen (2013) computes the depthmap for each image using PatchMatch stereo, and enforces depth consistency over neighboring views. Hu and Mordohai (2012) follow a scheme similar to the work by Campbell et al. (2008) but select the final depth through a process enforcing mutual consistency across all depthmaps. These methods require the depthmaps of other views to be available, placing less emphasis on the accuracy of the individual depthmaps.

2.2.2 Robustness

Robust stereo performance for crowd-sourced data is an ongoing research effort. Images downloaded via keyword searches from the Internet (such as Flickr[1] or Panoramio[2]) typically

[1] https://www.flickr.com/

[2] http://www.panoramio.com/

consist of unstructured imagery with a large portion of unrelated images. To discern a suitable input datum for stereo, Frahm et al. (2010) use appearance clustering of a color augmented GIST descriptor (Oliva and Torralba, 2001) along with feature-based geometric verification. In contrast, the work by Heinly et al. (2015) discovers the relationships between images using in a streaming paradigm that registers images to a vocabulary tree built online. However, even when the unrelated images are purged, using the data for stereo is still challenging due to the heterogeneous capture characteristics.

To estimate the depthmap of an image, Frahm et al. (2010) select the most related images based on the number of sparse feature points shared in common. The depthmap is then estimated using the heuristic K-best planesweeping algorithm (Kang et al., 2001). Due to the issues such as illumination difference and occlusion, their estimated depthmaps are of low quality. Furukawa et al. (2010) use structure from motion (SFM) to purge redundant imagery but retain high-resolution geometry. Their iterative clustering merges sparse 3D points and cameras based on visibility analysis. Although intra-cluster image partitioning is not performed, the cluster size is limited in an effort to maintain computational efficiency. Goesele et al. (2007) address the viewpoint selection for crowd-sourced imagery by building small-sized image clusters using the cardinality of the set of common features among viewpoints and a parallax-based metric. This image-wide selection may not be robust to outlier camera pose estimates. After this, images are resized to the lowest common resolution in the cluster. Pixel depth is then computed using four images selected from the cluster based on local color consistency.

2.2.3 Efficiency

Efficiency is an important issue in depthmap estimation. Traditional methods on large baseline stereo generally involve exhaustive evaluations of a large number of depth hypotheses. The high complexity of computation is not only time-consuming (Yang and Pollefeys, 2003; Strecha et al., 2006; Gallup et al., 2007; Hu and Mordohai, 2012), but also prohibitive on less powerful devices such as smartphones and tablets.

To handle these issues, the recently proposed PatchMatch technique provides an efficient sampling scheme. Though the scheme has no strict theory or proof of its working mechanism, it has been empirically shown that it works very well in practice. PatchMatch was originally proposed to find approximate nearest neighbor matches between image patches in Barnes et al. (2009), and later Bleyer et al. (2011) introduce it to solve the two-view stereo problem. PatchMatch initializes each pixel with a random slanted plane at a random depth, then propagates high-confidence values to neighboring pixels. The nearby and the current pixels' slanted planes are tested, and the one with the best cost is kept. Besse et al. (2012) combine the PatchMatch sampling scheme and belief propagation to infer an MRF model that contains smoothness constraints. By combining guided filter and PatchMatch, Lu et al. (2013) provide an efficient edge-aware filtering for correspondence field estimation, which can be applied in two-view stereo. While the original PatchMatch stereo was a sequential method, Bailer et al. (2012) parallelize the algorithm by restricting the propagations to only horizontal and vertical directions. Our research further explores the potential of PatchMatch in wide baseline stereo with a large hypothesis space.

2.2.4 Point Cloud and Mesh Generation

So far, we have only discussed the works focusing on depthmap estimation. Other methods aim at generating a consistent 3D model (either point cloud or mesh) instead of depthmaps. Furukawa and Ponce (2010) aim at reconstructing a quasi-dense point cloud by densifying the sparse 3D points. They present an accurate patch-based multiview stereo approach that starts from a sparse set of matched keypoints, which are repeatedly expanded until visibility constraints are invoked to filter out false matches. Zaharescu et al. (2011) propose a mesh evolution framework based on a new self-intersection removal algorithm.

A typical approach for 3D mesh generation is to fuse the depthmaps into a consistent model by leveraging the redundant information across the depthmaps. Gallup et al. (2010b,a) develop heightmap-based fusion methods that work well for planar object surfaces such as building facades. Zach (2008) tackles the surface reconstruction task in a variational formulation. Given that all these methods are volumetric-based and hence memory-inefficient, Zheng et al. (2012) instead propose to compress the volume of interest using Haar wavelets, hence reducing the amount of memory required. Jancosek and Pajdla (2011) propose a method that reconstructs surfaces that do not have direct support in the input 3D points by exploiting visibility in 3D meshes. Their method has been shown to work robustly on textureless regions.

2.3 Dynamic Object Reconstruction

The following sections outline the related works of trajectory triangulation, image sequencing, articulated object reconstruction, non-rigid structure from motion (NRSFM), and single-view reconstruction.

2.3.1 Trajectory Triangulation

Avidan and Shashua (2000) first coined the task of trajectory triangulation, which is defined as reconstruction of a moving point from monocular images. That is, each dynamic point is observed only by one camera at a time. Their method assumes the dynamic point moves along a simple parametric trajectory, such as a straight line or a conic section. This is a rather strict constraint that impedes their method's application in real scenarios. In contrast, other methods (Park et al., 2010; Valmadre and Lucey, 2012; Zhu et al., 2011; Park et al., 2015) focus on a more general model by only assuming a smooth motion of dynamic objects.

Park et al. (2010) represent the trajectory with a linear combination of low-order discrete cosine transform (DCT) bases, and the trajectory is triangulated by estimating the coefficients of the linear combination. There are two fundamental limitations of their method as observed by Valmadre and Lucey (2012). First, there is no automated scheme to determine the optimal number (K) of DCT bases. Second, the correlation between the object trajectory and the camera motion inherently limits the reconstruction accuracy. To overcome the first limitation, Park et al. (2015) select K by checking the consistency of the reconstructed trajectory in an N-cross validation scheme. Alternatively, Valmadre and Lucey (2012) propose a new method without using DCT bases. They estimate the trajectory by minimizing the trajectory's response to a bank of high-pass filters. To overcome the second limitation, Zhu et al. (2011) propose to incorporate the 3D structures of a number of key frames to enhance the reconstructability. However, obtaining those key-frame 3D structures requires manual interaction. All the methods (Park et al., 2010; Valmadre and Lucey, 2012; Zhu et al., 2011) require the sequencing information of the images, but in natural capture setups, the availability of sequencing information and high reconstructability typically cannot be fulfilled simultaneously (Zhu et al., 2011; Park et al., 2015).

2.3.2 Sequencing and Synchronization

Sequencing information is important in trajectory triangulation. Recently, Basha et al. (2012, 2013) target the problem of determining the temporal order of a collection of photos without recovering the 3D structure of the dynamic scene. The method by Basha et al. (2012) relies on two images taken from roughly the same location to eliminate the uncertainty in the sequencing. Basha et al. (2013) later introduce a solution that leverages the known temporal order of the images within each camera. Both of these methods assume that dynamic objects move close to a straight line within a short time period, but in practice, points can deviate considerably from the linear motion model, especially when the temporal discrepancy between images is large.

Video synchronization has attracted much attention in the computer vision community (Tuytelaars and Gool, 2004; Shrestha et al., 2010; Rao et al., 2003). Those methods have various constraints such as camera motion, availability of sound, and number of videos.

2.3.3 Articulated Object Reconstruction

Trajectory triangulation suffers from the reconstructability problem of inaccurate reconstruction if the camera motion is relatively small compared to the object motion (Park et al., 2015). In the case of 3D reconstruction of articulated objects, we can enforce an additional constraint that the distances between joint points (according to the topology) are fixed, which helps to reduce ambiguities in reconstruction. Based on the previous work by Park et al. (2010), the authors further reconstruct 3D articulated motion with the constraint that a trajectory remains at a fixed distance with respect to its parent trajectory (Park and Sheikh, 2011). Their work shows the improvement of the reconstructibility over their earlier approach (Park et al., 2010). However, the formulation involves solving an NP-hard quadratic programming problem, which is intractable in the case of a large number of input images. To conquer the limitation, Valmadre et al. (2012) develop a dynamic programming approach that is guaranteed to solve the problem in a timely manner. As opposed to articulated object reconstruction, our research focuses on reconstructing more general dynamic objects.

2.3.4 Non-rigid SfM

One class of related works solve the non-rigid structure from motion (NRSFM) problem, which targets simultaneous recovery of camera motion and 3D structure using an image sequence. These methods typically start from a set of 2D correspondences across frames. As an important extension of the well-known Tomasi-Kanade factorization (Tomasi and Kanade, 1992), Bregler et al. (2000) tackle the NRSFM problem through matrix factorization, with the assumption that deforming non-rigid objects can be represented by a linear combination of low-order shape bases. It was later shown by Xiao et al. (2004) that utilizing only orthogonality constraints on the camera rotation is not enough, and a basis prior is required to uniquely determine the shape bases. However, Akhter et al. (2009a) discover that in spite of the inherent ambiguity in the shape bases, the 3D shape itself can be uniquely recovered without ambiguity. Recently, Dai et al. (2014) have proposed a new prior-free method that estimates the shape matrix without explicitly recovering the shape bases, which is achieved by minimizing the rank (nuclear norm) of the shape matrix.

As a dual method to the above shape-based methods, Akhter et al. (2009b) propose the first trajectory-based NRSFM approach, which leverages DCT bases to approximately represent point trajectories. While shape-based approaches typically do not require sequencing information, trajectory-based approaches completely fail if image frames are randomly shuffled (Dai et al., 2014).

At first glance, it seems that the NRSFM problem targets a more complete problem than the trajectory triangulation problem since the former additionally assumes unknown camera poses. However, these approaches assume orthographic or weak perspective camera models, and it has been shown empirically that the extension of these methods to the projective camera model is not straightforward (Park et al., 2010). There are works for projective non-rigid shape and motion recovery based on tensor estimation (Hartley and Vidal, 2008; Vidal and Abretske, 2006), but this challenging problem is still under ongoing research. Moreover, the NRSFM methods only recover the shape of the object without absolute translation due to the inherent ambiguity arising from the unknown shape translation and the unknown camera translation.

2.3.5 Single Image Reconstruction

While trajectory triangulation and NRSFM methods estimate 3D points from an image sequence, other works target the problem of 3D reconstruction from a single image. Since there is only one view of the object, the object motion, either static or dynamic, becomes irrelevant for the reconstruction.

Some works focus on 3D reconstruction of a Manhattan world (Coughlan and Yuille, 1999), which is defined as man-made scenes with mainly orthogonal facades. In this scenario, 3D reconstruction from a single image can be simplified to finding the 3D lines and planes within the scene. The work by Delage et al. (2005) uses an MRF model to identify the different planes and edges in the scene, as well as their orientations. Then, an iterative optimization algorithm is applied to infer the planes' positions. Ramalingam and Brand (2013) reconstruct the 3D lines in a Manhattan scene from a single image using linear programming that identifies a sufficient minimal set of least-violated line connectivity constraints.

There are other approaches mainly relying on supervised learning. Hoiem et al. (2005) label the image regions as ground, vertical, and sky with a pre-trained classifier, then "cut and fold" the image into a pop-up model like children's pop-up books. The method is limited to the application of outdoor scenes containing simple ground and vertical structures. Saxena et al. (2008) propose a method for computing a depthmap from a single still image by using a hierarchical multi-scale MRF that incorporates several features. The features are manually designed, and the parameters of the MRF model are trained using ground-truth depths. Instead of manually choosing features, Eigen et al. (2014) recently propose to estimate the depthmap of a single image by employing two deep network stacks: one that makes a coarse global prediction based on the entire image, and another that refines this prediction locally. Due to the wide applicability of the topic, single depthmap estimation using supervised learning is currently an active research topic.

CHAPTER 3: PATCHMATCH BASED JOINT VIEW SELECTION AND DEPTHMAP ESTIMATION

3.1 Introduction

Multi-view depthmap estimation (MVDE) methods strive to determine a view dependent depthfield by leveraging the local photoconsistency of a set overlapping images observing a common scene. Applications benefiting from high quality depthmap estimates include dense 3D modeling, classification/recognition (Shotton et al., 2011) and image based rendering (Chen and Williams, 1993). However, achieving highly accurate depthmaps is inherently difficult even for well controlled environments where factors such as viewing geometry, image-set color constancy, and optical distortions are rigorously measured and/or corrected. Conversely, practical challenges for robust depthmap estimation from non-controlled input imagery (*i.e.* Internet collected data) include mitigating heterogeneous resolution and scene illuminations, unstructured viewing geometry, scene content variability and image registration errors (*i.e.* outliers). Moreover, the increasing availability of crowd sourced datasets has explicitly brought efficiency and scalability to the forefront of application requirements, while implicitly increasing the importance of data association management when processing such large scale datasets.

The input for MVDE is commonly assumed to consist of a convergent set of images along with reliable estimates of their pose and calibration parameters. The extracted depthmap will correspond to the pixel-wise 3D structure hypotheses that best explain the available image observations in terms of some measure of visual similarity with respect to a reference image. Ironically, the potential robustness afforded by having multiple available images is compromised by the inherent variability in pairwise photoconsistency observations. In practice, correct depth hypotheses may provide low photoconsistency in a source image subset (e.g. occlusions or illumination aberrations), while incorrect depth hypotheses may register high image similarity (e.g. repetitive structure or

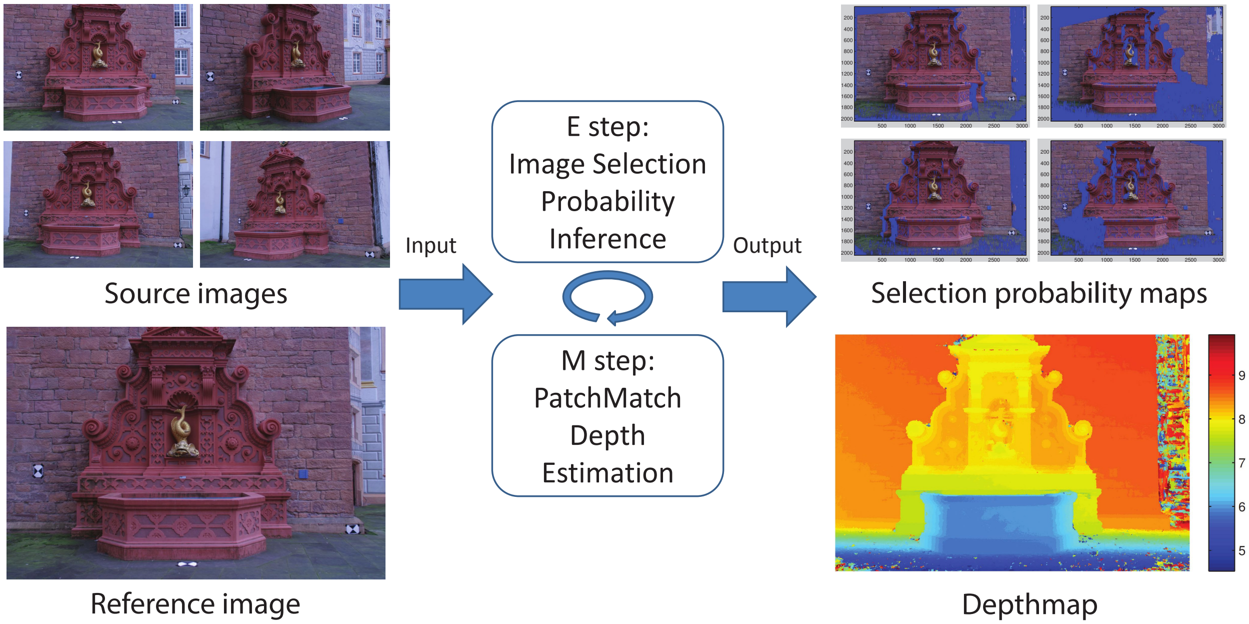


Figure 3.1: Overview of our approach. Input imagery is used to jointly estimate a depthmap and pixel level view associations. Blue regions in the view selection probability map indicate pixels in the reference image lacking reliable observations in the corresponding source image.

homogeneous texture). These technical challenges render multi-view depth hypothesis evaluation as a problem of robust model fitting, where a demarcation between inlier and outlier photoconsistency observations is required. We tackle this implicit data association problem by addressing the question: *What aggregation subset of the source image set should be used to estimate the depth of a particular pixel in the reference image*?

We propose a probabilistic framework for depthmap estimation that jointly models pixel-level view selection and depthmap estimation given pairwise image photoconsistency. An overview is depicted in Figure 3.1. The corresponding graphical model is trained using EM algorithm. The algorithm iterates between view selection by inference in the probabilistic model, and PatchMatch-like depth sampling and propagation (Bleyer et al., 2011; Bailer et al., 2012). The insight leveraged by our method is the spatial smoothness of the photoconsistency with respect to the good source images given the correct depth (Strecha et al., 2006; Goesele et al., 2007). Our expectation of having a high overlap of photoconsistent source images among neighboring pixels in the reference image, leads to modeling the depth estimation problem as a Markov chain where the unobserved states correspond to binary indicator variables for the selection probability of each source image.

We summarize the contributions and advantages of the framework as follows.

1. **Accuracy:** Mitigation of spurious data associations at the pixel level provides state-of-the-art accuracy results for single depthmap estimation.

2. **Efficiency:** Deployment of PatchMatch sampling and propagation enables reduced computational burden as well as GPU implementation.

3. **Scalability:** Linear storage requirement with respect to the number of source images, as opposed to the exponential growth in the joint view selection and depth estimation model by Strecha et al. (2006), enables handling selection instances comprising hundreds of images.

3.2 Joint View Selection and Depth Estimation

In this section we provide an overview of our PatchMatch propagation scheme (Section 3.2.1), describe our probabilistic graphic model (Section 3.2.2), describe our variational inference approximation to the model's posterior probability (Section 3.2.3 and Section 3.2.4) and finalize describing our implementation (Section 3.2.5).

3.2.1 PatchMatch Propagation for Stereo

Our algorithm uses single oriented planes instead of the multiple oriented planes (Bailer et al., 2012), to reduce the three-dimensional search space (depth and two angles for the orientated plane) to one dimension. We alternatively perform upward/downward propagations during the odd iterations and perform rightward/leftward propagations during even iterations. To calculate the depth at pixel (i, j) for the rightward propagation, only the depth at positions $(i, j-1)$ and (i, j) are tested on pixel (i, j) (Figure 3.2). Likewise, only one neighbor is considered for all other propagations. The propagation schemes of (Bleyer et al., 2011) and (Bailer et al., 2012) are shown in Figure 3.2.

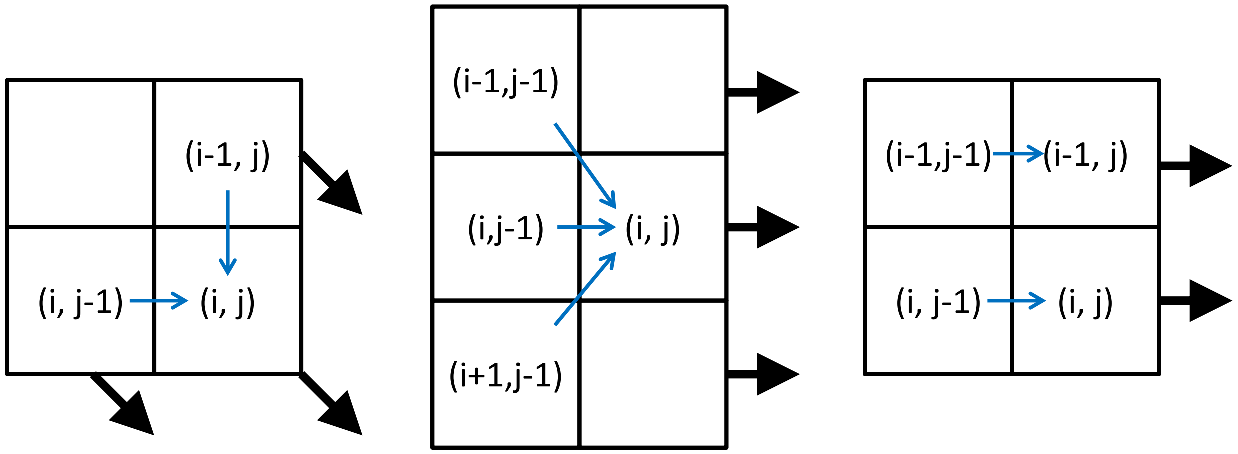


Figure 3.2: The black and blue arrows show the propagation directions and the sampling schemes. Left: Top left to bottom right propagation in (Bleyer et al., 2011). Middle: Rightward propagations in (Bailer et al., 2012). Right: Our rightward propagation.

In case of the absence of proper depth hypotheses, we can additionally draw and test H random depth hypotheses for each pixel during propagations. In this work, we use $H = 1$ and hence have 3 depth hypotheses tested per pixel in a propagation, i.e. the depths of current and the neighboring pixel along with one random depth. Without loss of generality, we limit our discussion henceforth to the rightward horizontal propagation.

3.2.2 Graphical Model

In our algorithm, the depth is estimated for a reference image X^{ref}, given a set of M (unstructured) source images $X^1, X^2, ...X^M$ with known camera calibration parameters, which are the output of a typical structure from motion system such as VisualSFM (Wu, 2013). We denote the correct depth associated with each pixel l on image X^{ref} as θ_l.

Photo-consistency values for the correct depth of a given pixel across a set of source images may be incongruent for some of the source images. This may be attributed to a diversity of factors such as occlusions, calibration errors, illumination aberration, etc. Therefore, depth estimation for a given pixel entails the determination of which subset of source images will provide the most robust estimate. Our model defines M binary variables $Z_l^m \in \{0, 1\}, m = 1, 2...M$ for each pixel l in the

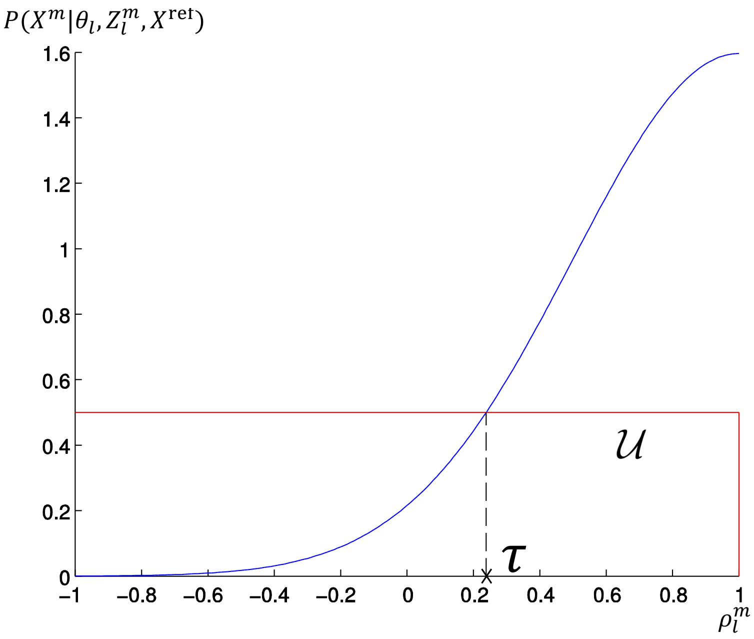


Figure 3.3: Distribution of Equation (3.1)

reference image X^{ref}, where Z_l^m is 1 if image X^m is selected for depth estimation of pixel l, and 0 otherwise.

We first define the likelihood function. We denote the color patch centered at pixel l in the reference image as X_l^{ref}. Given a pixel l and its correct depth θ_l in the reference image X^{ref}, a color patch X_l^m on source image m can be determined through homography warping (Shen, 2013). If $Z_l^m = 1$, the probability that the observed color patch X_l^m is color-consistent with X_l^{ref} should be high. We use NCC (normalized cross correlation) to compare the two color patches X_l^m and X_l^{ref} as a robust proxy to single pixel comparisons, and denote the NCC measurement as ρ_l^m. In the case when $Z_l^m = 0$, X_l^m has arbitrary colors due to factors such as occlusion or calibration errors, so the probability of observing X_l^m is unrelated to X_l^{ref} and considered uniformly distributed. Therefore we propose the following likelihood function

$$P(X_l^m|Z_l^m, \theta_l, X_l^{\text{ref}})= \begin{cases} \frac{1}{NA} e^{-\frac{(1-\rho_l^m)^2}{2\sigma^2}} & \text{if } Z_l^m = 1 \\ \frac{1}{N}\mathcal{U} & \text{if } Z_l^m = 0, \end{cases} \tag{3.1}$$

where A equals to $\int_{-1}^{1} exp\{-\frac{(1-\rho)^2}{2\sigma^2}\}d\rho$ and N is a constant. Note that NCC value ranges in $[-1, 1]$ and equals 1 with the best color consistency. Consistent with our intuition, a color patch X_l^m with high NCC value ρ_l^m has high probability $P(X_l^m|Z_l^m = 1, \theta_l, X_l^{\text{ref}})$. $\mathcal{U}$ is the uniform distribution in the range $[-1, 1]$ with probability density 0.5. Note that NCC computation is affine invariant and multiple pairs of color patches can generate the same NCC value. To simplify the analysis without affecting depthmap quality, Equation (3.1) assumes the number of color patches X_l^m that can generate any specific NCC value is the same and equals to N. Since only the ratio $P(X_l^m|Z_l^m = 1, \theta_l, X_l^{\text{ref}})/P(X_l^m|Z_l^m = 0, \theta_l, X_l^{\text{ref}})$ matters in the model inference discussed in Section 3.2.3 and Section 3.2.4, we can safely ignore the constant N in Equation (3.1).

In Equation (3.1) σ is the parameter determining the suitability of an image based on NCC measurement ρ_l^m. As seen in Figure 3.3, a soft threshold τ is determined by σ. If ρ_l^m is larger than τ, it is more likely that image m is selected, and vice versa. Since X_l^{ref} is observed for each pixel, $P(X_l^m|Z_l^m, \theta_l, X_l^{\text{ref}})$ is simply denoted as $P(X_l^m|Z_l^m, \theta_l)$ in the rest of the paper.

The depths of nearby pixels are considered independent, while the pairwise smoothness is put on the nearby selection variables along the current propagation direction (Figure 3.4) through the transition probabilities:

$$P(Z_l^m|Z_{l-1}^m) = \left(\begin{smallmatrix} \gamma & 1-\gamma \\ 1-\gamma & \gamma \end{smallmatrix}\right). \tag{3.2}$$

Setting γ close to 1 encourages neighboring pixels to have similar selection preference for source images X^m. To enable parallel computation, we only enforce pairwise constraint on the pixels of the same row in the horizontal propagations. Note Figure 3.4 only shows one row of selection variables for each of the source images.

Finding the optimal selection $\boldsymbol{Z}$ and depth $\boldsymbol{\theta}$ given all the images $\boldsymbol{X}$ equates to computing the maximum of the posterior probability (MAP) $P(\boldsymbol{Z}, \boldsymbol{\theta}|\boldsymbol{X})$. The Bayesian approach firstly computes the joint probability based on the graphical model (Figure 3.4) and normalizes over $P(\boldsymbol{X})$. The

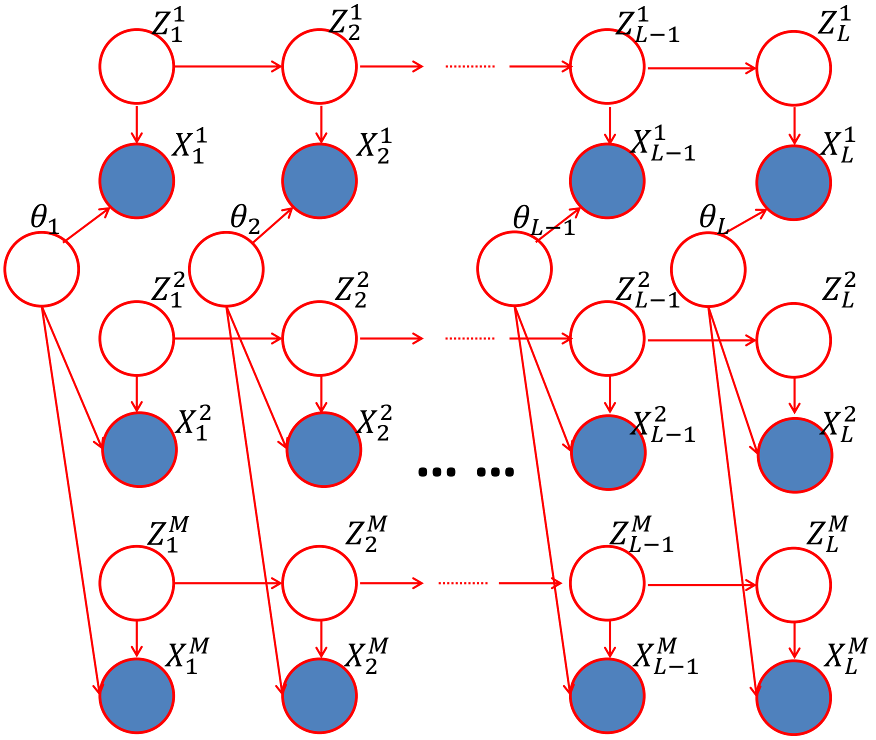


Figure 3.4: The graphical model. θ_l is the depth of pixel l. Z_l^m is the selection of image m at pixel l. X_l^m is the observation (colors) on the source image m given depth θ_l.

joint probability is

$$P(\boldsymbol{X}, \boldsymbol{\theta}, \boldsymbol{Z}) = \prod_{m=1}^{M} [P(Z_1^m) \prod_{l=2}^{L} P(Z_l^m | Z_{l-1}^m) \prod_{l=1}^{L} P(X_l^m | Z_l^m, \theta_l)] \prod_{l=1}^{L} P(\theta_l), \tag{3.3}$$

where L is the number of pixels along the propagation direction of the reference image. We use an uninformative uniform distribution for prior $P(Z_1^m)$ as well as depth prior $P(\theta_l)$ since we have no preference without observations. However, computing $P(\boldsymbol{X})$ is intractable as it requires to sum over all possible values of $\boldsymbol{Z}$ and $\boldsymbol{\theta}$.

We interleave pixel level inference of image selection probability with fixed depth, and depth updating with fixed image selection probability. Our approach is a variant of the generalized EM (GEM) (Neal and Hinton, 1998). Similarly to the work by Neal and Hinton (1998), we use variational inference theory to justify our algorithm.

3.2.3 Variational Inference

Variational inference selects a member of a *restricted* family of distributions $q(\boldsymbol{Z}, \boldsymbol{\theta})$ to approximate the true posterior distribution $P(\boldsymbol{Z}, \boldsymbol{\theta}|\boldsymbol{X})$, in the sense that the KL divergence between these two is minimized (Bishop, 2006). *The restriction is imposed purely to achieve tractability.* The real posterior distribution is over the set of unobserved variables $\boldsymbol{\theta} = \{\theta_l | l = 1, ..., L\}$ and $\boldsymbol{Z} = \{\boldsymbol{Z}^m | m = 1, ..., M\}$, where $\boldsymbol{Z}^m = \{Z_1^m, Z_2^m, ..., Z_L^m\}$ is a chain in the graph. We put restrictions on the family of distributions $q(\boldsymbol{Z}, \boldsymbol{\theta})$, assuming that it is factorizable into a set of distributions (Bishop, 2006):

$$q(\boldsymbol{Z}, \boldsymbol{\theta}) = \prod_{m=1}^{M} q_m(\boldsymbol{Z}^m) \prod_{l=1}^{L} q_l(\theta_l). \tag{3.4}$$

For tractability, we further constrain each $q_l(\theta_l)$, $l = 1, 2, ..., L$ to the family of Kronecker delta functions:

$$q_l(\theta_l) = \delta(\theta_l = \theta_l^*) = \begin{cases} 1, & \text{if } \theta_l = \theta_l^* \\ 0, & \text{otherwise} \end{cases} \tag{3.5}$$

where θ_l^* is a parameter to be estimated. This assumption is in contrast to most other works (Strecha et al., 2004, 2006; Sun et al., 2002, 2005), which discretize the depth as a means to recover the whole posterior distribution of the depth. Once the distribution $q_l(\theta_l)$ is determined, θ_l is set to θ_l^* to maximize the approximate posterior distribution Equation (3.4), so θ_l^* is actually the final estimated depth. Conversely, the depths $\boldsymbol{\theta}$ can be considered as parameters shared by different chains instead of as variables. This assumption seamlessly combines the PatchMatch sampling scheme in the graphic model inference.

The variational method seeks to find a member $q^{\text{opt}}(\boldsymbol{Z}, \boldsymbol{\theta})$=$\prod_{m=1}^{M} q_m^{\text{opt}}(\boldsymbol{Z}^m) \prod_{l=1}^{L} q_l^{\text{opt}}(\theta_l)$ from the family $q(\boldsymbol{Z}, \boldsymbol{\theta})$, minimizing the KL divergence between $q(\boldsymbol{Z}, \boldsymbol{\theta})$ and $P(\boldsymbol{Z}, \boldsymbol{\theta}|\boldsymbol{X})$ under the constraint that $q_m(Z^m), m = 1, ...M$ are normalized ($q_l(\theta_l)$ is guaranteed to be normalized as it is constrained to be a Kronecker delta function):

$$\begin{aligned} \underset{q(\boldsymbol{Z},\boldsymbol{\theta})}{\text{minimize}} \quad & \mathrm{KL}(q(\boldsymbol{Z},\boldsymbol{\theta})||P(\boldsymbol{Z},\boldsymbol{\theta}|\boldsymbol{X})) \\ \text{subject to} \quad & \sum\nolimits_{Z^m} q_m(Z^m) = 1, m = 1, \ldots, M. \end{aligned} \tag{3.6}$$

Note the optimization is performed over distributions, but not over variables. To optimize over $q_m(\boldsymbol{Z}^m)$, the standard solution (Bishop, 2006) is $\log\left(q_m(\boldsymbol{Z}^m)\right) = \mathbb{E}_{\setminus m}[\log\left(P(\boldsymbol{X},\boldsymbol{\theta},\boldsymbol{Z})\right)] + const$, where $\mathbb{E}_{\setminus m}$ is the expectation of $\log\left(P(\boldsymbol{X},\boldsymbol{\theta},\boldsymbol{Z})\right)$ taken over all variables not in $q_m(\boldsymbol{Z}^m)$ (Bishop, 2006). Then we have

$$q_m^{\text{opt}}(\boldsymbol{Z}^m) \propto \Psi(\boldsymbol{Z}^m) \prod\nolimits_{l=1}^{L} P(X_l^m|Z_l^m, \theta_l = \theta_l^*), \tag{3.7}$$

where $\Psi(\boldsymbol{Z}^m)$=$P(Z_1^m) \prod_{l=2}^{l=L} P(Z_l^m|Z_{l-1}^m)$. The right side of Equation (3.7) has form of joint probability of a Hidden Markov Chain with fixed transition probability from Equation (3.2) and fixed emission probability Equation (3.1). The probability of each hidden variable $q(Z_l^m)$ can be efficiently inferred by forward-backward algorithm (Bishop, 2006). See Section 3.2.4 for more details. This corresponds to the E step of the GEM algorithm.

To optimize over $q_l(\theta_l)$ we seek an optimal parameter θ_l^{opt} for the distribution $q_l(\theta_l)$ that minimizes Equation (3.6). Suppressing the terms not involving θ_l gives

$$\theta_l^{\text{opt}}\text{= } \underset{\theta_l^*}{\text{argmax}} \sum_{m=1}^{M} q(Z_l^m\text{=}1) \ln P(X_l^m|Z_l^m\text{=}1, \theta_l\text{=}\theta_l^*). \tag{3.8}$$

By substituting Equation (3.1) into Equation (3.8), we get

$$\theta_l^{\text{opt}} = \underset{\theta_l^*}{\text{argmin}} \sum\nolimits_{m=1}^{M} q(Z_l^m = 1)(1 - \rho_l^m)^2, \tag{3.9}$$

where ρ_l^m is a function of θ_l^*. To find θ_l^{opt} in the above equation, 3 depth hypotheses sampled based on PatchMatch are tested, and the one that maximizes Equation (3.9) is assigned to the parameter of the distribution $q_l(\theta_l)$. This step is the M step of the GEM algorithm. Note that the righthand side of Equation (3.9) is a weighted sum of $(1-\rho_l^m)^2$ with weight equal to the image selection probability. Hence, a small value of $q(Z_l^m=1)$, designating image m as not favorable, contributes less in evaluating the parameter θ_l^*.

Improvement: Equation (3.9) is computationally expensive for hundreds of source images. Based on Equation (3.9), it is unnecessary to compute ρ_l^m if the corresponding image selection probability $q(Z_l^m=1)$ is very small. Hence, we propose a Monte Carlo based approximation (Bishop, 2006). Rewriting Equation (3.9) as

$$\theta_l^{\text{opt}} = \underset{\theta_l^*}{\text{argmin}} \sum\nolimits_{m=1}^{M} P(m)(1-\rho_l^m)^2 \tag{3.10}$$

where the new distribution $P(m) = \frac{q(Z_l^m=1)}{\sum_{m=1}^{M} q(Z_l^m=1)}$ can be deemed as the probability of image m being the best for depth estimation of pixel l. We draw samples based on the distribution $P(m)$ to obtain a subset S, then

$$\theta_l^{\text{opt}} = \underset{\theta_l^*}{\text{argmin}} \frac{1}{|S|} \sum\nolimits_{m\in S} (1-\rho_l^m)^2. \tag{3.11}$$

Empirically, 15 samples suffice to attain good results.

Both distributions $q_m^{\text{opt}}(\boldsymbol{Z})$ and $q_l^{\text{opt}}(\theta_l)$ are coupled. The computation of θ_l^* requires $q(Z_l^m)$ to be known (Equation (3.9)), but to infer $q(Z_l^m)$ in Equation (3.7), we need θ_l^* available. The next subsection introduces the update scheme that computes the distributions iteratively.

3.2.4 Update Schedule

The common way to compute approximate distributions is coordinate descent optimization method. Namely, one distribution is optimized while other distributions remain fixed. Choosing which distribution to optimize over in each step is arbitrary or scheduled based on application, but it always decreases the cost function in Equation (3.6). We choose to interleave updates of $q_l(\theta_l)$ and $q_m(\boldsymbol{Z}^m)$ as it is able to quickly propagate the correct depth into nearby pixels. For clarity, our explanations below use one chain and omit the image index m for each variable.

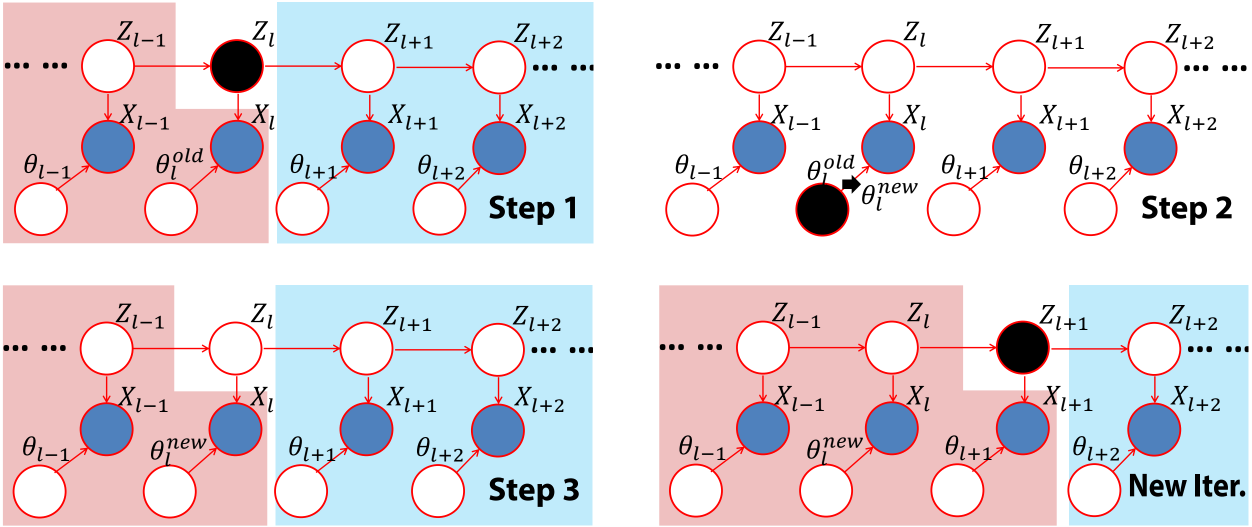


Figure 3.5: Update schedule. See text for more details.

For more details on Hidden Markov Chain inference, we refer the reader to text (Bishop, 2006). The forward-backward algorithm is used to infer the probability of hidden variables Z_l.

$$q(Z_l) = \frac{1}{A}\alpha(Z_l)\beta(Z_l), \tag{3.12}$$

where A is the normalization factor. $\alpha(Z_l)$ and $\beta(Z_l)$ are the forward and backward message for variable Z_l computed using the following Equations,

$$\alpha(Z_l) = p(X_l|Z_l, \theta_l) \sum_{Z_{l-1}} \alpha(Z_{l-1})P(Z_l|Z_{l-1}), \tag{3.13}$$

$$\beta(Z_l) = \sum_{Z_{l+1}} \beta(Z_{l+1})P(X_{l+1}|Z_{l+1}, \theta_{l+1})P(Z_{l+1}|Z_l). \tag{3.14}$$

Both the forward and backward messages are computed recursively (e.g. $\alpha(Z_l)$ is computed using $\alpha(Z_{l-1})$). In Figure 3.5, the variables covered in red area and blue area contribute to the forward and backward messages respectively.

We perform the following update schedule as is shown in Figure 3.5. In step 1, compute $q(Z_l)$ using Equation (3.12), (3.13) and (3.14) for each source image (*i.e.* $q(Z_l^m)$,$m = 1...M$). In step 2, update the depth from θ_l^{old} to θ_l^{new} using Equation (3.9) or Equation (3.11). In step 3, with θ_l^{new},

Input: All images, depthMap (randomly initialized or from previous propagation) **Output**: Updated depthMap m – image index, l – pixel index	Eq.	Step
For $l = L$ to 1		
For $m = 1$ to M		
Compute backward message β_l^m	(3.14)	1
For $l = 1$ to L		
For $m = 1$ to M		
Compute forward message α_l^m	(3.13)	1
Compute $q(Z_l^m)$	(3.12)	1
Draw depth hypotheses by PatchMatch		
Estimate θ_l^* for $q_l(\theta_l)$	(3.9 or 3.11)	2
For $m = 1$ to M		
Recompute forward message α_l^m	(3.13)	3

Table 3.1: The algorithm of a row/column propagation.

we recompute forward message $\alpha(Z_l)$, which is further used to compute $\alpha(Z_{l+1})$ recursively in Equation (3.13). Next we start at variable Z_{l+1} with the same process until reaching the end of the row in the image. Before the update process, the backward message for each variable can be computed recursively (Equation (3.14)) and stored in memory.

3.2.5 Algorithm Integration

We now describe the computational framework implementing our depth estimation and view selection formulation. The depthmap is initialized with random values within the depth range. Alternatively, sparse 3D measurements may be included within our initialization. Next, the rightward, downward, leftward and upward propagations are applied in sequence. Each propagation (except in the first iteration) uses the depth results of the former propagation. Within each propagation, updates of the depth and the selection probability are interleaved as described in Section 3.2.4. After two or three sweeps, each containing the four direction propagations, the depthmap reaches a stable state. Convergence may alternatively be verified through tracking the number of modified depth estimates up to a threshold. As each row is independent from other rows given our graphical model and processed in exactly the same way during one propagation, it can be easily parallelized for leveraging GPUs. We describe the algorithm for processing one row within rightward propagation in Table 3.1.

Discussion. The estimation of the exact image-wide MAP for our graphical model would require a Hidden Markov Random Field (MRF) formulation instead of our Hidden Markov Chain approximation. Our choice of using propagation direction specific chain models was driven by computational efficiency/tractability. The proposed framework enables us to easily interleave the propagation with hidden variable inference while fostering implementation parallelism. The enforcement of smoothness constraints on the hidden variables enables non-oscillating behavior of our evolving depth estimates. Our PatchMatch based framework has linear computational and storage complexity with respect to to input data size while being independent of the size of the depth search space. Namely, since the number of tested depth hypotheses (3 for each propagation) is small and constant, the computation complexity of our method is $O(WHM)$, where W, H, and M are the width, height and number of images. Methods using complete hypotheses search, (e.g. Sun et al. (2002); Strecha et al. (2006)), require $O(WHMD)$ computations, where D is the size of hypotheses space normally reaching up to thousands of hypotheses.

3.3 Experiments

We evaluate the accuracy of our method on standard ground truth benchmarks and highlight our robustness on multiple crowd sourced datasets. In both evaluation scenarios we juxtapose our results with current state-of-the-art methods. We implemented our method in CUDA and executed on a Nvidia GTX-Titan GPU. For all experiments, the total number multi-directional propagations is set to 3 and we use $\sigma = 0.45$ in the likelihood function (Equation (3.1)) and $\gamma = 0.999$ in the transition probabilities (Equation (3.2)).

Ground truth evaluation. We evaluated on the Strecha datasets (Fountain-P11 and Herzjesu-P9) presented in Strecha et al. (2008) as they include ground truth 3D structure measurements. We use all dataset images full resolution, set the NCC patch size to 15 by 15 and approximate the depth range from sparse 3D points. We measure pixel-wise depth errors as our goal is to generate a single

	2cm	10cm	2 cm	10cm
Error	fountain-P11		Herzjesu-P9	
Ours	0.732	0.911	0.619	0.833
Ours(P)	0.769	0.929	0.650	0.844
LC (Hu and Mordohai, 2012)	0.754	0.930	0.649	0.848
FUR (Furukawa and Ponce, 2010)	0.731	0.838	0.646	0.836
ZAH (Zaharescu et al., 2011)	0.712	0.832	0.220	0.501
TYL (Tylecek and Sara, 2010)	0.732	0.822	0.658	0.852
JAN (Jancosek and Pajdla, 2011)	0.824	0.973	0.739	0.923

Table 3.2: The percentage of pixels with absolute error less than 2cm and 10cm. Entries *Ours(P)* and *Ours* denote our results with and without postprocessing. Reported values are from the work by Hu and Mordohai (2012).

depthmap instead of one consistent 3D scene model. We calculate the number of pixels with the depth error less than 2cm and 10cm from the ground truth and compare with (Hu and Mordohai, 2012; Furukawa and Ponce, 2010; Zaharescu et al., 2011; Tylecek and Sara, 2010; Jancosek and Pajdla, 2011). All the pixels with accessible ground truth depth are evaluated to convey both the accuracy and the completeness of the estimated depthmaps. We omit evaluation of the dataset's two extremal views as done by Hu and Mordohai (2012).

We use slanted planes of single orientation instead of fronto-parallel planes. The single dominant orientation direction can be estimated by projecting sparse 3D points onto the ground plane as described in Gallup et al. (2007). We further apply two optional depthmap refinement schemes to increase the final accuracy. Our basic depth refinement uses a smaller NCC patch (5x5), while eliminating random depth sampling, during an additional propagation sweep. We then use deterministic fine-grain sampling (20 hypotheses) in the depth neighborhood (± 1 cm.) of each pixel's depth estimate as proposed in Shen (2013). Finally, a median filter of size 9x9 is applied to each raw depthmap. Table 3.2 shows our method is comparable to the state-of-the-art methods. Note the results of Hu and Mordohai (2012); Tylecek and Sara (2010); Jancosek and Pajdla (2011) are obtained through multi-depthmap fusion, while our method directly estimates individual depthmaps.

Advantages of pixel level view selection. Figure 3.6 shows our comparison to the occlusion-robust best-K planesweeping method (Kang et al., 2001), where for a given depth hypothesis, the cost is the average of the best K costs, with K being predefined. When K is set to the number of

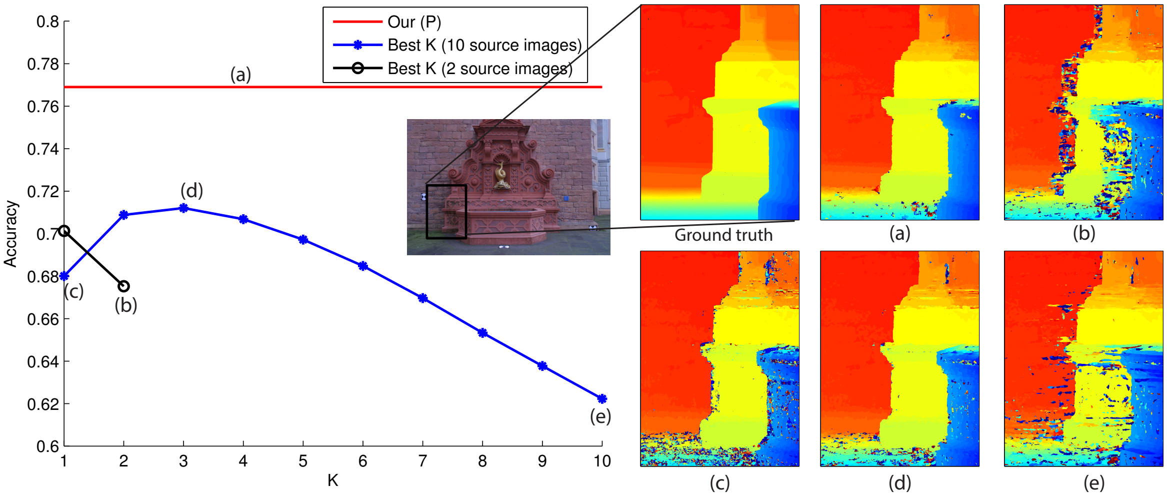


Figure 3.6: Left: Comparison against best-K aggregation. Right: Raw depthmap output of a partially occluded subregion with results for different dataset-aggregation combinations.

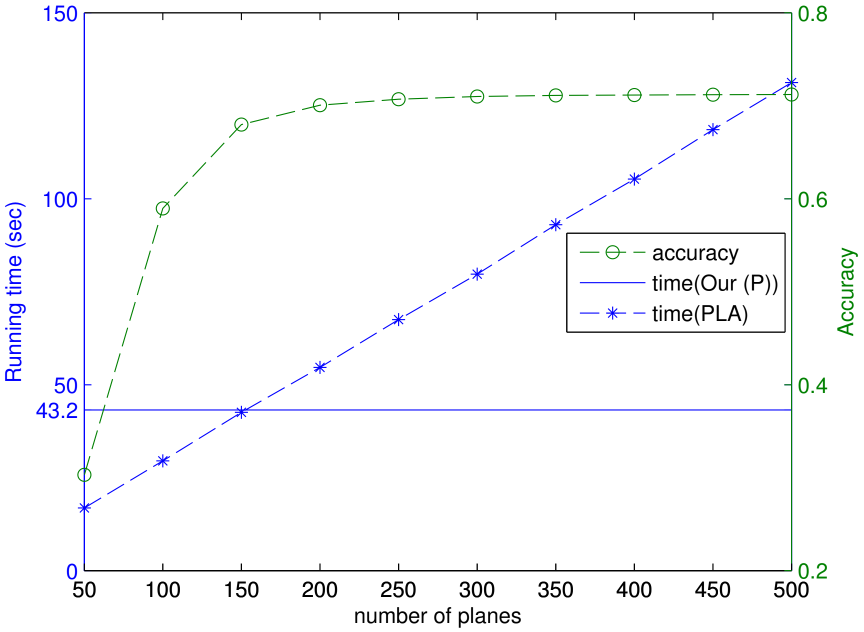


Figure 3.7: Fountain dataset performance. Left: Average running time. Right: Percentage of pixels given different thresholds. PLA is the planesweep algorithm with all source images and K=3, while GOS is the method by Goesele et al. (2007).

source images, it degenerates to the basic planesweeping algorithm that computes the cost using all source images. As opposed to our method with dynamic weights of images used for depth recovery, this method has a worse ability to handle occlusion. We compute depthmaps of the fountain-P11 data with varying K and otherwise fixed parameters, using 2000 planes. The percentage of pixels within 2cm difference from the ground truth is taken as a measure of the error. We run the planesweeping using two different dataset types. In the first case all 10 source images are used. Alternatively, we use the neighboring left and the right images. Figure 3.6 shows our results outperform all fixed aggregation schemes and illustrates the raw depthmap output of a partially occluded subregion.

Run times for our method are compared with an optimized GPU planesweeping code. Figure 3.7 shows the linear dependence of computation time to the number of planes as well the diminishing accuracy improvements provided by increasing the search space resolution. Our PatchMatch sampling and propagation scheme only requires depth range specification, foregoing explicit search space discretization.

Robustness to noisy SfM estimates. The advantage of pixel-level view selection across the entire dataset is highlighted in Figure 3.8, where we compare our results for corrupted SFM estimates against those obtained using the approach by Goesele et al. (2007). Figure 3.8 depicts Alexander Nevsky Cathedral in Sofia having indistinguishable structure in the tower structure (*i.e.* view invariant appearance due to structural symmetry). A set of 136 images, comprised of two mutually exclusive subsets observing the front or back, was fed into VisualSFM (Wu, 2013) yielding a corrupted 3D model where symmetric structure is fused along with the disjoint camera clusters. The approach by Goesele et al. (2007) initially selects a global subset of 20 images based on the corrupted SFM estimates and select independently for each pixel's depth estimation a fixed number (typically 4) of images from the global subset (similar to using K-best aggregation with K=4). If the global subset is unbalanced or is contaminated by corrupted estimates the completeness of the model is compromised, as shown in Figure 3.8 where the background dome is missing. We consider the entire dataset and implicitly mitigate such outliers. Moreover, we re-executed the code by Goesele et al. (2007) with manually filtered camera poses and indeed achieved correct results.

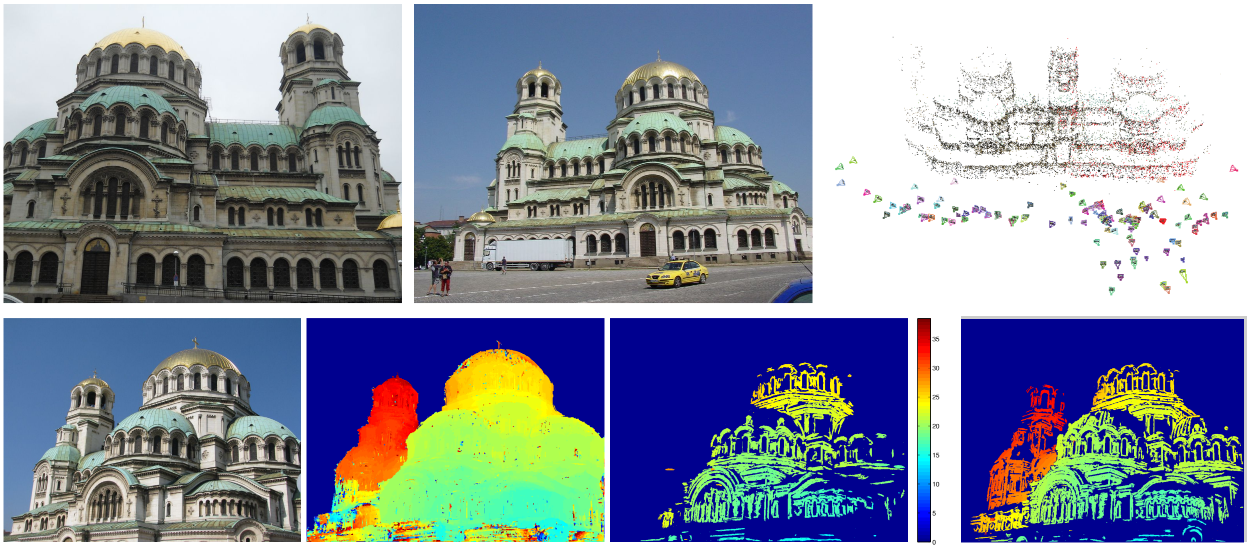


Figure 3.8: Top: Front and back of Alexander Nevsky Cathedral and estimated 3D model. Bottom: original image, depthmap of our method and the method by Goesele et al. (2007) with wrong and correct camera poses.

Robustness to varying capture characteristics. We tested our algorithm on Internet photo collections (IPC) downloaded from the Flickr for six different scenes: Paris Triumphal Arch (195 images), Brandenburg Gate (300 images), Notre Dame de Paris (300 images), Great Buddha (212 images), Mt. Rushmore (206 images) and Berlin Cathedral (500 images). In order to control GPU memory, we optionally resize imagery to no more than 1024 pixels for each dimension. Camera poses were calculated using VisualSFM (Wu, 2013). The average run time for Berlin Cathedral is 98.3 secs/image. For illustration, sky region pixels are masked out using the method in Derek Hoiem (2005) as post-processing. To compare with the method by Goesele et al. (2007), we run the author's code [1] on the same dataset with default parameters except for setting the matching window size to the same as ours (7x7). The results shown in Figure 3.9 illustrate that, while both approaches are robust to wide variations in illumination, scale and scene occlusions across the datasets, our approach tends to provide increased completeness of depthmap estimates. We attribute this to our more flexible view selection framework. In contrast to the method by Goesele et al. (2007), we avoid making initial hard image discriminations through an initial global image subset.

[1] http://www.gris.informatik.tu-darmstadt.de/projects/multiview-environment/

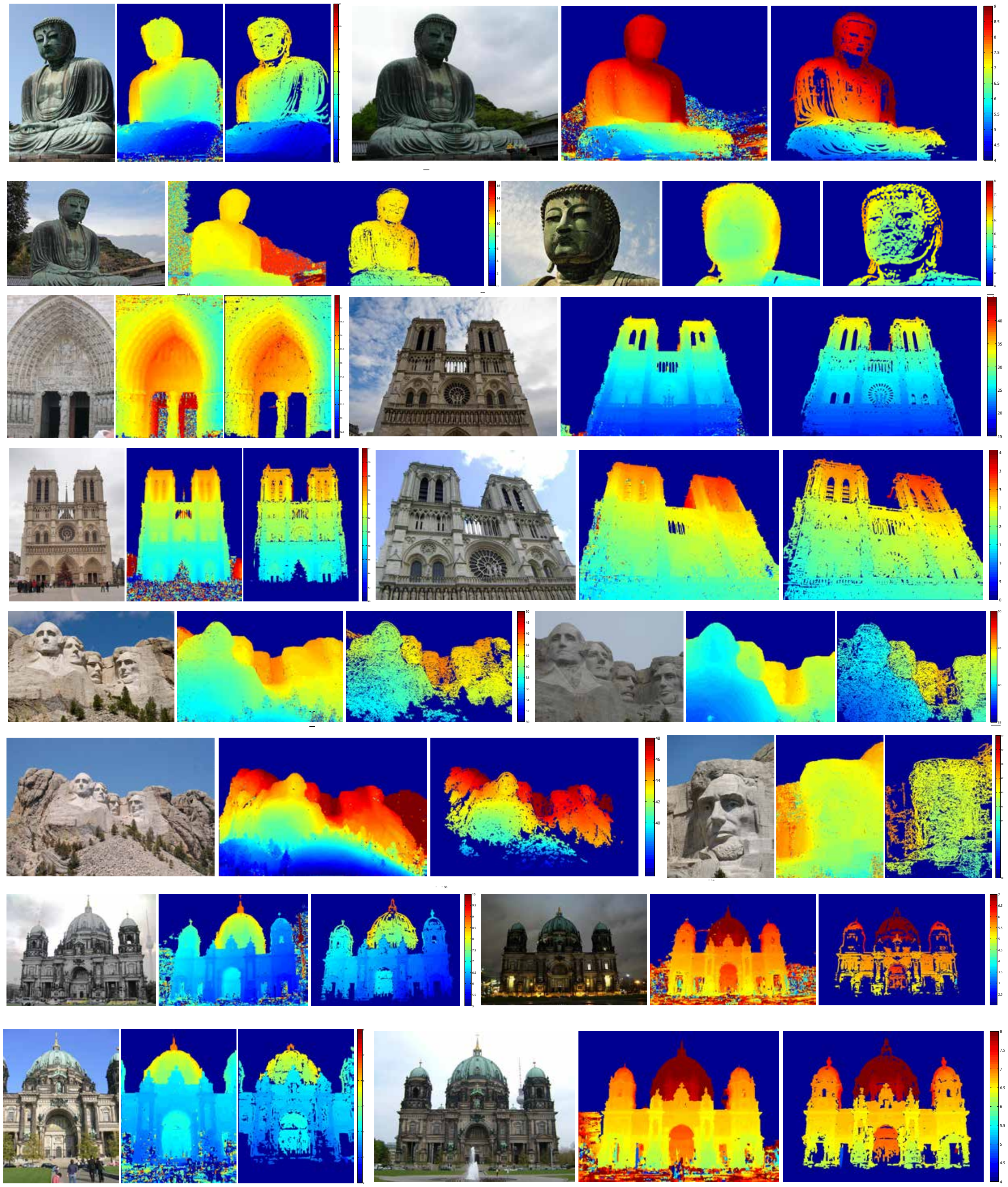


Figure 3.9: Each image triplet depicts a reference image along with our and Goesele's ((Goesele et al., 2007)) depthmap output (Best viewed in color).

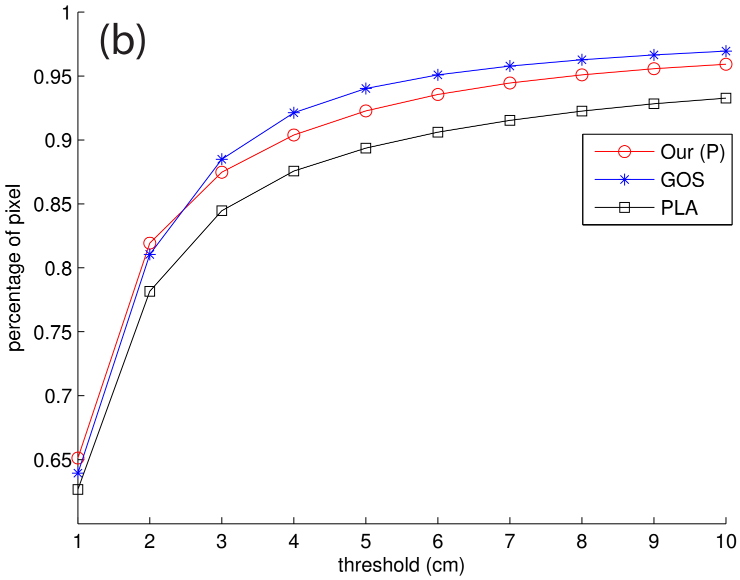


Figure 3.10: Fountain dataset performance.Percentage of pixels given different thresholds. PLA is the planesweep algorithm with all source images and K=3, while GOS is the method by Goesele et al. (2007).

To quantitatively compare the accuracy of our results with the work by Goesele et al. (2007), in the absence of ground truth geometry for crowd source datasets, we revisit the accuracy of both methods in the Strecha Fountain dataset. The method by Goesele et al. (2007) rejects outlier depth estimates based on the NCC values and the viewing angles. Hence, we only compare the accuracy of the reliable pixels as classified by Goesele et al. (2007) (comprising 75.4% of total image pixels). Figure 3.10 shows our approach outperforming both the method in Goesele et al. (2007) and planesweep for high accuracy thresholds. We expect the same accuracy ranking to carry over to the crowd sourced data results.

3.4 Conclusion

We have presented an efficient and effective joint solution to the view selection and depth estimation problem in multi-view stereo. Our solution relies on estimating a selection probability of each source image at the pixel level. The selection probability encodes the existence of contingency issues such as occlusions, specular aberrations and calibration errors. Moreover, by automatically determining reference image data associations with respect to a general source image dataset, we can encompass a larger range input imagery while increasing overall system robustness. Our approach has also extended the PatchMatch algorithm to encompass robust multi-view depth estimation within a probabilistic framework. Reported results achieve state-of-the-art accuracy in ground truth benchmarking while enabling robust operation in crowd-sourced datasets.

CHAPTER 4: JOINT OBJECT CLASS SEQUENCING AND TRAJECTORY TRIANGULATION (JOST)

4.1 Introduction

Techniques of 3D reconstruction from crowd-sourced imagery have developed rapidly over the past decade (Agarwal et al., 2011; Frahm et al., 2010; Zheng et al., 2014a; Heinly et al., 2015). Despite these advances, the state-of-the-art methods only target the static parts of a scene, treating the dynamic elements as hindrances to reconstruction. Since dynamic objects are typically the major focus of real-life images, recovering their 3D information enables applications such as better scene visualizations and dynamic event analysis. Therefore, it is of great interest to reconstruct these dynamic objects.

In this chapter, we propose a method to estimate the 3D positions of dynamic objects of the same class moving in a common path given a set of unstructured images as input. Figure 4.1a shows example input images in a dateset that captures pedestrians walking on a sidewalk. We assume no temporal correlation among the images, and that no two images observe the same dynamic object instance. The main challenge of the reconstruction problem resides in recovering 3D positions given noncurrent captures (or even single observations) of the dynamic objects, which invalidates the use of traditional 3D triangulation. The only constraint available for our problem is the fact that all observed instances of an object class move along a possibly diverging path in the 3D scene, which we define as an object class trajectory. Figure 4.2 shows one example of object class trajectory.

In this chapter, we define the spatial ordering of the objects along the trajectory as sequencing information. If the trajectory is modeled as a graph, this information can also be regarded as the topology of the trajectory (see Figure 4.2 for an example of a cross-shaped trajectory). This sequencing information captures the spatial proximity of the dynamic objects in 3D space, and therefore triangulating the object class trajectory necessarily involves learning a trajectory topology.

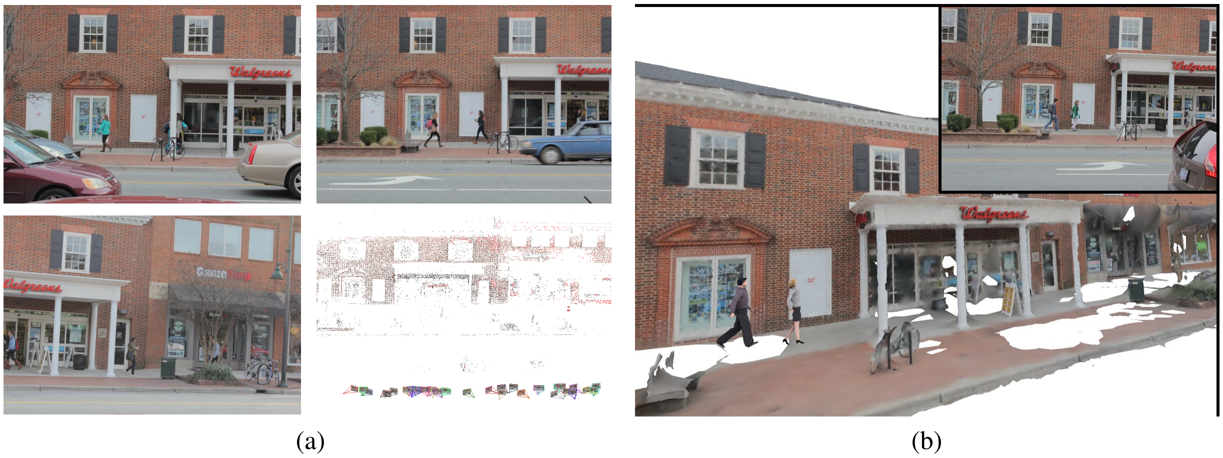


Figure 4.1: Left: Tree images of the pedestrian dataset and the output of structure from motion. Right: Estimated 3D positions of two pedestrians that are captured in the image. Note we only reconstruct one 3D position for each dynamic object instance instead of a dense 3D model. For visualization purposes, a general mesh model is inserted into each estimated position.

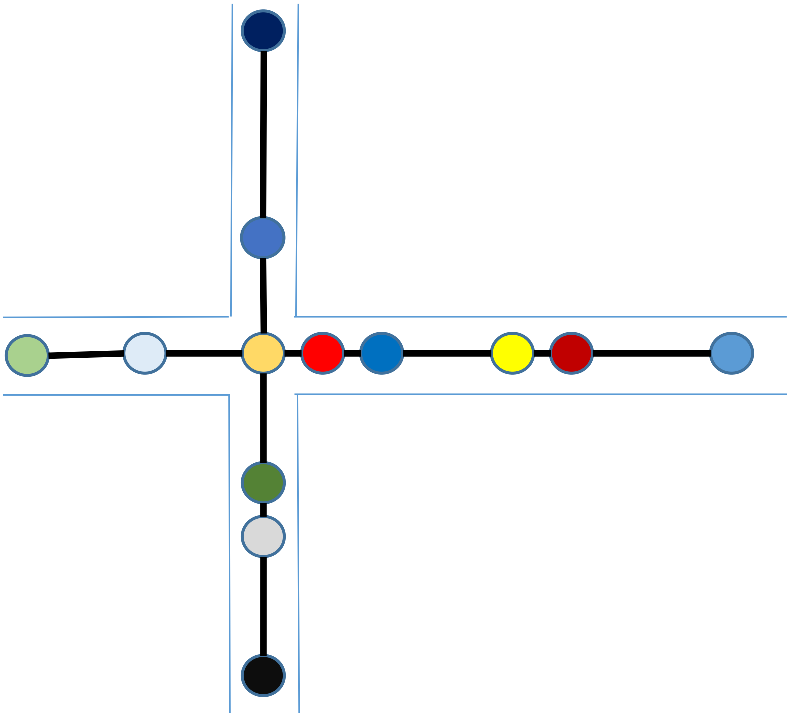


Figure 4.2: Example of cross-shaped object class trajectory. The circles of different colors represent object instances of the same object class in the path. Note the topology of the trajectory is tree-structured. Each image only observes one or a few object instances, and we use all the observations to recover the object class trajectory.

To recover the object class trajectory, our method simultaneously determines the sequencing information of the objects and their 3D positions on the path, which we call joint object class sequencing and trajectory triangulation (JOST). We leverage all the observations on different images to recover the object class trajectory, which in turn provides an estimate for the 3D positions of the dynamic objects in each image (see Figure 4.1b).

4.2 Joint Object Class Sequencing and Trajectory Triangulation

We now detail our method for joint object class sequencing and trajectory triangulation from unstructured images. Our method includes three steps:

1. Spatially register the cameras to a common 3D coordinate system using structure from motion (SfM).
2. Detect object instances and estimate motion tangents from input imagery as the 2D observations of the dynamic objects.
3. Leverage the observations of the object instances to simultaneously
   (a) determine the sequencing information of the objects along a trajectory (*i.e.*, the topology of the trajectory), and
   (b) triangulate the geometry of the corresponding object class trajectory.

While we exploit known methods to solve for camera registration, object detection, and motion tangents in the images, our main contribution is an algorithm for tackling challenge 3. To this end, we model our problem as a nonconvex optimization problem, and develop a novel solver involving a step of discrete optimization followed by another step of continuous refinement. Next, we introduce our system in detail.

4.2.1 Spatial Registration

The goal of the initial spatial registration in our method is to establish camera registration in a common coordinate system. Given that in all our datasets a fair portion of the images contains static background structures, we use the publicly available structure from motion tool VisualSFM (Wu, 2013) to register all the cameras. See Figure 4.1a for an example.

The obtained camera registration determines the camera center $\tilde{\mathbf{C}}_j$ of the j-th camera. With known camera parameters, each pixel in a camera defines a viewing ray with direction $\mathbf{r}$ in the 3D scene space. For our object class trajectory, we are only interested in the ray direction $\mathbf{r}_i$ associated with the object instance i of the desired class (for simplicity we refer to them as objects), where $i = 1, \ldots, N$, and N is the total number of detected objects over all frames. The ray $\mathbf{X}_i(t_i)$ in the 3D space represents a 1D subspace on which the imaged object has to lie and is described by

$$\mathbf{X}_i(t_i) = \mathbf{C}_i + t_i \mathbf{r}_i, \tag{4.1}$$

where $t_i \geq 0$ is the positive distance from the camera center $\mathbf{C}_i$ along the ray $\mathbf{X}_i(t_i)$. In the following, we implicitly assume the condition $t_i \geq 0$. We denote the camera center associated with an object instance i as $\mathbf{C}_i$ with $\mathbf{C}_i = \tilde{\mathbf{C}}_j$, where $\tilde{\mathbf{C}}_j$ is the center of the camera j in which the object instance i is detected. This means if more than one object is detected in camera j, there will be multiple $\mathbf{C}_i$ with identical positions. Once we obtain the value for t_i, the object position can be uniquely determined.

4.2.2 Object Detection and Motion Tangent Estimation

Our proposed method leverages object detection techniques to determine the 2D observations of the dynamic objects. We identify one 2D position of each detected object on the image by the center of the detection bounding box. These object detections provide us the viewing rays where the dynamic objects are placed.

To robustly perform joint object class sequencing and trajectory triangulation, our proposed method also uses the motion tangent of each object, which is defined as the moving direction of the dynamic object in the 3D space. The problem of motion tangent estimation has been solved for videos (Zhao et al., 2003), but in the absence of temporal coherence among the images, our method needs to estimate the motion tangent based on a single image.

The particular choice of object detection and motion tangent estimation methods depends on the specific object class and the scenes. We discuss our choices in Section 4.3, and for now we assume we have at our disposal the 2D observation defining the ray $\mathbf{X}_i(t_i)$, as well as a coarse estimate of the motion tangent $\mathbf{d}_i$ for each object i.

4.2.3 Object Class Trajectory Triangulation

Assuming known viewing rays $\mathbf{X}_i(t_i)$ and the motion tangent $\mathbf{d}_i$, we now define the object class trajectory estimation problem before delving into our data representation and our estimation framework. For ease of description, we directly leverage the viewing rays $\mathbf{X}_i(t_i)$ of the detected objects i and thereby implicitly use the camera parameters and the 2D observations.

For a particular class of objects, an object class trajectory describes a path taken by the dynamic objects of the desired class through the 3D scene. Each observation (object detection) is a sample of the point on the trajectory. Since there are only a finite number of observations of objects along the path, we only sample a discrete set of 3D points on the path, and the combination of piecewise linear functions between the true object positions $\mathbf{X}_i^*$ represents the object class trajectory.

An important principle for obtaining an object class trajectory is that sampling along a path results in a collection of spatially adjacent points. A trajectory should therefore connect all observed points in such a way that total (spatial) traversal between the points is minimized. We formulate this as a minimization of the following cost function:

$$\min_{\mathbf{p}} \sum_{(i,j)\in\mathbf{p}} \|\mathbf{X}_i^* - \mathbf{X}_j^*\|_2^2. \tag{4.2}$$

here $\mathbf{p}$ defines the topology spanning the path with minimum cost, given as a list of adjacency relationships between all the points $\mathbf{X}_i^*$, $i = 1, \ldots, N$.

While the trajectory above is based on the ground truth 3D object positions $\mathbf{X}_i^*$, we can only observe the rays $\mathbf{X}_i(t_i)$. To recover the object class trajectory, we also need to determine the position of each object i along its viewing ray $\mathbf{X}_i(t_i)$. We propose to find the adjacency relation by optimizing over variables $\mathbf{t} = [t_1, \ldots, t_N]$ and $\mathbf{p}$ jointly as

$$\min_{\mathbf{p},\mathbf{t}} \sum_{(i,j)\in\mathbf{p}} \|\mathbf{X}_i(t_i) - \mathbf{X}_j(t_j)\|_2^2. \tag{4.3}$$

To robustly recover both $\mathbf{t}$ and $\mathbf{p}$ simultaneously, we further leverage the information of motion tangent. The direction of the local trajectory should be the same or similar to the motion tangent of the dynamic objects. Given the motion tangents $\mathbf{d}_i$ estimated from the images, we can further constrain the trajectory, obtaining an optimization problem:

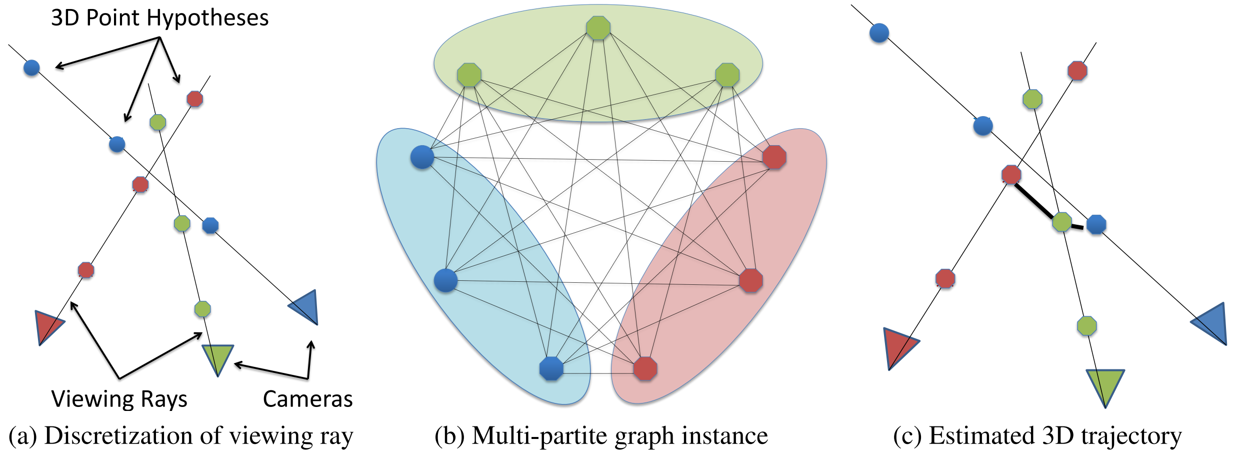
$$\min_{\mathbf{p},\mathbf{t}} \sum_{(i,j)\in\mathbf{p}} \|\mathbf{d}_{i,j} \times (\mathbf{X}_i(t_i) - \mathbf{X}_j(t_j))\|_2^2 + \lambda\|\mathbf{X}_i(t_i) - \mathbf{X}_j(t_j)\|_2^2, \tag{4.4}$$

where the operator × is the vector cross product, and λ is a positive weight (discussed at length in Section 4.2.7). The direction $\mathbf{d}_{i,j}$ is selected from $\mathbf{d}_i$ and $\mathbf{d}_j$ as the motion tangent that is closest to the 3D motion direction $\mathbf{X}_i(t_i) - \mathbf{X}_j(t_j)$. More details about computation of $\mathbf{d}_{i,j}$ will be illustrated in Section 4.2.6. The first cost term in Equation (4.4) adds the penalization if the local direction of the recovered trajectory deviates from the motion tangent. The optimization procedure simultaneously determines both the adjacency $\mathbf{p}$ and the object positions through $\mathbf{t}$.

Optimization of the non-convex function in Equation (4.4) is inherently difficult. To achieve this, we propose a new discrete-continuous optimization strategy using a generalized minimum spanning tree (GMST).

4.2.4 Generalized Trajectory Graph

To determine the object class trajectory, we conceptually have to choose for each ray $\mathbf{X}_i(t_i)$ the 3D point, and simultaneously determine the adjacency $\mathbf{p}$ representing the adjacency relations of the rays $\mathbf{X}_i(t_i)$, which defines the topology of the object class trajectory. Our discrete-continuous optimization strategy first uses a generalized minimum spanning tree (GMST) to find the adjacency list $\mathbf{p}$, and followed by a convex optimization over $\mathbf{t}$ with $\mathbf{p}$ being fixed.



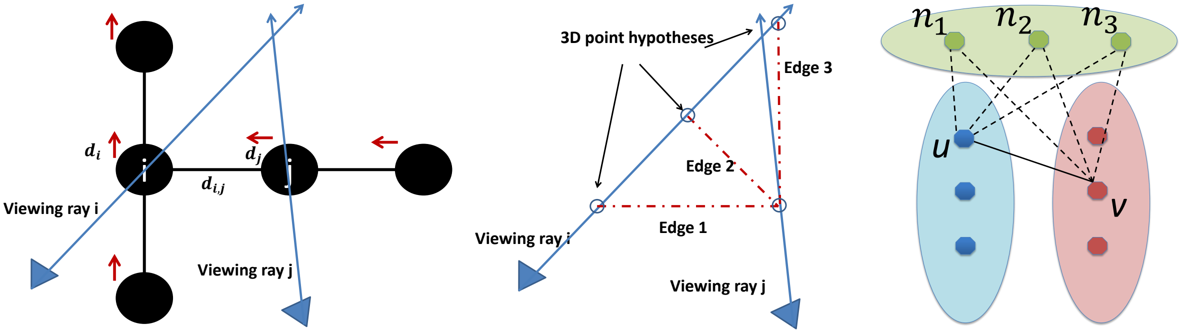
(a) Discretization of viewing ray (b) Multi-partite graph instance (c) Estimated 3D trajectory

Figure 4.3: Illustration of GMST. See the text for more details.

In the discrete optimization step, we map the continuous problem of finding the 3D point along each ray to a discrete problem of selecting a 3D point out of a set of discrete 3D points (see Figure 4.3a). Using this formulation, we determine one 3D point along each ray and the adjacency $\mathbf{p}$ by computing the GMST on an undirected multipartite graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ (Myung et al., 1995). This allows us to simultaneously determine the topology and the discrete 3D object positions.

An undirected multipartite graph is a graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ whose vertices are partitioned into N partite sets $\{V_1,\ldots,V_N\}$ with the number of partite sets $|V_i| = k$, while fulfilling $\mathcal{V} = V_1 \cup V_2 \cup \cdots \cup V_N$ and $V_o \cap V_p = \emptyset, \forall o \neq p$, with $o, p \in \{1,\ldots,N\}$. The multipartite graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ has only edges between the different partite sets of vertices V_o, and all edge costs are non-negative (see Figure 4.3b for an example). Next, we will explain on how we define the graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ based on Equation 4.4.

Each ray $\mathbf{X}_i(t_i)$ defines a one dimensional constraint on the 3D position of the object. We discretize the ray to obtain a discrete set of potential depth estimates. This leads to a finite set of possible 3D positions along the ray (see Figure 4.3a for an illustration), defining a finite set of 3D point hypotheses $\{\hat{\mathbf{X}}_i^o \mid o = 1,\ldots,k\}$, where k is the number of the discrete hypotheses along the ray. In our representation, each 3D point $\hat{\mathbf{X}}_i^o$ establishes a node V_i^o in the graph. The set of nodes $\{V_i^o \mid o = 1,\ldots,k\}$ related to the ray $\mathbf{X}_i(t_i)$ of object i defines a partite set of nodes V_i in the graph $\mathcal{G}(\mathcal{V},\mathcal{E})$. Given that no nodes within a group have any connecting edges, the resulting multipartite graph will contain no edges between the different depth hypotheses of object i.



(a) (b) (c)

Figure 4.4: In Figure 4.4a, the black nodes shows the real positions of dynamic objects. The red vector represents the direction associated with each object. In the shown example, $\mathbf{d}_{i,j}$ equals $\mathbf{d}_i$.

We now define the edge cost of the multipartite graph based on Equation (4.4). The multipartite graph only has edges between the nodes from different partite sets. We define the edge direction $\mathbf{d}_{i,j}$ between any two nodes V_i^o and V_j^p in the partite set i and partite set j, respectively, as the consistency of the 3D motion with the motion tangents $\mathbf{d}_i$ or $\mathbf{d}_j$ (see Section 4.2.2). This definition comes from the intuition that the edge direction should be compliant with the motion tangent observed in the images. Given the motion of two objects i and j, and their respective motion tangents $\mathbf{d}_i$ and $\mathbf{d}_j$, it is clear that the edge direction between the points $\hat{\mathbf{X}}_i^o$ and $\hat{\mathbf{X}}_j^p$ (associated with the nodes V_i^o and V_j^p) should be close to at least one of the motion tangents $\mathbf{d}_i$ and $\mathbf{d}_j$. Therefore, we define the edge cost $e(V_i^o, V_j^p)$ of the edge between the nodes V_i^o and V_j^p as

$$e(V_i^o, V_j^p) = \min(\|\mathbf{d}_i \times (\hat{\mathbf{X}}_i^o - \hat{\mathbf{X}}_j^p)\|_2^2, \|\mathbf{d}_j \times (\hat{\mathbf{X}}_i^o - \hat{\mathbf{X}}_j^p)\|_2^2) + \lambda\|\hat{\mathbf{X}}_i^o - \hat{\mathbf{X}}_j^p\|_2^2. \tag{4.5}$$

If only considering the first term in Equation (4.5), edges with 3D motion directions that are approximately parallel to $\mathbf{d}_i$ or $\mathbf{d}_j$ have lower cost than those are at an angle to both $\mathbf{d}_i$ and $\mathbf{d}_j$. For instance, Edge 1 and Edge 3 in Figure 4.4b have a relatively lower cost than Edge 2 because Edge 1 is parallel to $\mathbf{d}_j$ and Edge 3 is parallel to $\mathbf{d}_i$.

4.2.5 GMST

A generalized minimum spanning tree (GMST) on the graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is a tree of minimal cost that spans *exactly one node* from each partite set V_i. The GMST problem degenerates to a typical minimum spanning tree problem (Cormen et al., 2009) if each of the partite sets contains only one node. For our proposed graph, it means a GMST includes exactly one hypothesized 3D point from each observation. Furthermore, a GMST prefers the edge $e(V_i^o, V_j^p)$ that has a small cost and is compliant with the motion tangents in the images, as those edges have lower edge cost. Accordingly, a GMST is our desired solution for estimating the object class trajectory. Note that if we sample an infinite number of 3D points along each viewing ray, the corresponding GMST problem is equivalent to the original formulation in Equation (4.4).

The multipartite graph defined above contains a large number of edges, which increases the complexity of computing the GMST. We use a deterministic method introduced by Ferreira et al. (2012) to remove the redundant edges that are guaranteed not to be included in the GMST. We show a specific toy example in Figure 4.4c to illustrate the method. If the cost of edge (u, v) is larger than any cost of the 6 edges (u, n_l) and (v, n_l), $l = 1, 2, 3$, the edge (u, v) is safe to be removed. A simple proof is that if edge (u, v) exists in the computed GMST, we could remove edge (u, v) and replace it with one of the 6 edges to obtain a new GMST with lower cost. Therefore, edge (u, v) can not be present in the GMST. Moreover, it is plausible to explore other ways to remove edges based on given prior information. For instance, if it is known the pairwise neighboring 3D objects are close in 3D space, we can safely remove the edges that connect two spatially distant point hypotheses by applying a predefined threshold.

The GMST problem was first introduced by Myung et al. (1995) and has been extensively studied in the past two decades (Myung et al., 1995; Dror et al., 2000; Feremans et al., 2002; Oncan et al., 2008; Ferreira et al., 2012) due to its wide applications in telecommunications, agriculture watering, and facility distribution design (Myung et al., 1995; Dror et al., 2000). Unlike the minimum spanning tree (MST) problem, which can be solved in polynomial time, finding the

GMST is proved to be NP-hard (Myung et al., 1995). Myung *et al.* (Myung et al., 1995) and Feremans *et al.* (Feremans et al., 2002) propose several integer programming formulations for the GMST problem. However, those methods provide no guarantee of efficiency, especially when the problem scale is large. The computational challenge of the GMST problem has led to the development of metaheuristics (Oncan et al., 2008; Ferreira et al., 2012) that search the hypothesis space and are empirically shown to be effective.

We exploit the state-of-the-art GRASP-based approach proposed by Ferreira *et al.* (Ferreira et al., 2012). GRASP (Greedy Randomized Adaptive Search Procedure) is a metaheuristic that consists of iterations comprising two phases: 1) solution construction and 2) solution improvement through local search. Ferreira et al. (2012) propose a method that considers several solution construction algorithms, a local search procedure, and two additional mechanisms: path-relinking and iterative local search. We refer readers to their paper (Ferreira et al., 2012) for more details.

4.2.6 Continuous Refinement

The output of GMST computation is the estimation of the 3D points (denoted as $\widehat{\mathbf{X}}_i$ for object i) and the adjacency topology $\mathbf{p}$ of the object class trajectory. Then, $\mathbf{d}_{i,j}$ is chosen to be one of $\mathbf{d}_i$ and $\mathbf{d}_j$ that has smaller angle to the vector $\hat{\mathbf{X}}_i - \hat{\mathbf{X}}_j$,

$$\mathbf{d}_{i,j} = \underset{\mathbf{d}\in\{\mathbf{d}_i,\mathbf{d}_j\}}{\operatorname{argmax}}(|\mathbf{d}\cdot(\hat{\mathbf{X}}_i - \hat{\mathbf{X}}_j)|), \tag{4.6}$$

where operator $\cdot$ is the vector dot product. We fix the adjacency $\mathbf{p}$ given by the GMST and continue with a final continuous refinement step for the 3D object position, through a convex program optimization over variable $\mathbf{t}$

$$\min_{\mathbf{t}} \sum_{(i,j)\in\mathbf{p}} \|\mathbf{d}_{i,j} \times (\mathbf{X}_i(t_i) - \mathbf{X}_j(t_j))\|_2^2 + \lambda\|\mathbf{X}_i(t_i) - \mathbf{X}_j(t_j)\|_2^2. \tag{4.7}$$

4.2.7 Reconstructability Analysis

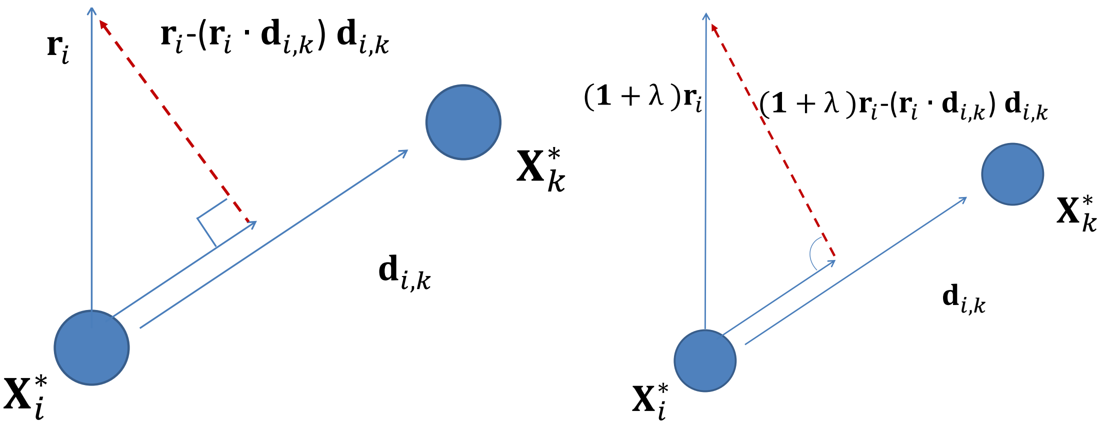
Now, we analyze the reconstructability of the proposed method. That is, we determine under which conditions the solution of Equation (4.4) generates accurate 3D points. The direct analysis of Eq (4.4) is difficult, since it needs to determine in which situation the adjacency $\mathbf{p}$ with minimum cost, out of N^{N-2} possible adjacencies (Wikipedia, 2014), corresponds to the real object class trajectory. However, we find that having the motion tangent constraint reduces the possibility of finding the incorrect adjacency $\mathbf{p}$. Hence, we focus on the reconstructability of the continuous method in Equation (4.7) given the adjacency $\mathbf{p}$.

Assume we already know the ground truth 3D point $\mathbf{X}_i^*$ of object i, $i = 1, \ldots, N$. Given that $\mathbf{X}_i^*$ is present on the viewing ray $\mathbf{X}_i$, we move the camera center $\mathbf{C}_i$ to $\mathbf{X}_i^*$ along the ray $\mathbf{X}_i(t)$ in direction $\mathbf{r}_i$. Then any point on the line that passes through $\mathbf{X}_i^*$ and has ray direction $\mathbf{r}_i$ can be represented as $\mathbf{X}_i(s_i) = \mathbf{X}_i^* + s_i\mathbf{r}_i$, where s_i is the signed distance along the viewing ray (not the positive distance as defined by the t_i). Then Equation (4.7) can be reformulated as

$$\min_{\mathbf{s}} \sum_{(i,j)\in\mathbf{p}} \|\mathbf{d}_{i,j} \times (\mathbf{X}_i(s_i) - \mathbf{X}_j(s_j))\|_2^2 + \lambda\|\mathbf{X}_i(s_i) - \mathbf{X}_j(s_j)\|_2^2, \tag{4.8}$$

where $\mathbf{s} = [s_1, \ldots, s_N]$. Though s_i is signed distance and t_i is positive distance, minimizing Equation (4.7) and Equation (4.8) still output the same 3D point positions, as long as the computed 3D points in Equation (4.8) are in front of the camera centers. We will see that this is normally true, since the computed 3D points are typically close to their ground truth position if the system is well-conditioned.

We denote the solution of Equation (4.8) as s^{opt}. The true 3D points are ideally reconstructed if $\text{s}^{\text{opt}} = 0$, since $\mathbf{X}_i(0)$ equals to $\mathbf{X}_i^*$ given $\text{s}^{\text{opt}} = 0$. More specifically, s^{opt} equals the signed Euclidean distance between the 3D points produced by Equation (4.7) and the ground truth X_i^*. Therefore, $||\text{s}^{\text{opt}}||$ is the Euclidean error of the estimated 3D points by Equation 4.7. In the remainder of this section, we further analyze in which situations $||\text{s}^{\text{opt}}||$ is small to better understand the quality of the estimated 3D points.


$\mathbf{r}_i$
$\mathbf{r}_i$-($\mathbf{r}_i \cdot \mathbf{d}_{i,k}$) $\mathbf{d}_{i,k}$
$\mathbf{X}_k^*$
$\mathbf{d}_{i,k}$
$\mathbf{X}_i^*$
$(\mathbf{1}+\lambda)\mathbf{r}_i$
$(\mathbf{1}+\lambda)\mathbf{r}_i$-($\mathbf{r}_i \cdot \mathbf{d}_{i,k}$) $\mathbf{d}_{i,k}$
$\mathbf{X}_k^*$
$\mathbf{d}_{i,k}$
$\mathbf{X}_i^*$

(a) $\lambda = 0$

(b) $\lambda > 0$

Figure 4.5: Plot of Equation (4.10) with $\lambda = 0$ and $\lambda > 0$.

The minimum value of Equation (4.8) is achieved at the point where the first derivative relative to s equals 0. This produces a linear equation system $\mathbf{A}\mathrm{s}^{\mathrm{opt}} = \mathbf{b}$, where the ith row and jth column of matrix $\mathbf{A}$ is

$$A_{ij} = \begin{cases} [(\mathbf{r}_i \cdot \mathbf{d}_{i,j})\mathbf{d}_{i,j} - (1+\lambda)\mathbf{r}_i] \cdot \mathbf{r}_j & \text{if } i \neq j \text{ and } (i,j) \in \mathbf{p} \\ 0, & \text{if } i \neq j \text{ and } (i,j) \notin \mathbf{p} \\ \sum_{(i,k)\in\mathbf{p}} [1 + \lambda - (\mathbf{r}_i \cdot \mathbf{d}_{i,k})^2] & \text{if } i = j. \end{cases} \tag{4.9}$$

The ith element of vector $\mathbf{b}$ is

$$b_i = \sum\nolimits_{(i,k)\in\mathbf{p}} (\mathbf{X}_k^* - \mathbf{X}_i^*) \cdot [(1+\lambda)\mathbf{r}_i - (\mathbf{r}_i \cdot \mathbf{d}_{i,k})\mathbf{d}_{i,k}]. \tag{4.10}$$

Next, we explain that if the adjacency $\mathbf{p}$ is correctly found, the reconstructabililty of the object class trajectory mainly depends on the condition number of the linear system defined by $\mathbf{A}$. With careful observation, we can see Equation (4.9) and Equation (4.10) have the following interesting properties:

1. If $\mathbf{b}$ is 0, $\mathrm{s}^{\mathrm{opt}}$ equals 0, which means the solution of Equation (4.7) recovers the ground truth 3D points. There are a few situations where $\mathbf{b}$ equals 0. (1) In the case of a static object

$\mathbf{X}_i^* = \mathbf{X}_k^*$, $\mathbf{b}$ equals 0 based on Equation (4.10). (2) Careful observation reveals that if λ is set to 0, in Equation (4.10) the vector $(1+\lambda)\mathbf{r}_i - (\mathbf{d}_{i,k} \cdot \mathbf{r}_i)\mathbf{d}_{i,k}$ is perpendicular to vector $\mathbf{X}_i^* - \mathbf{X}_k^*$ (Figure 4.5a), hence $b_i = 0$. However, we will show that with $\lambda = 0$, the linear system $\mathbf{A}\mathbf{s} = \mathbf{b}$ is unstable due to the high condition number of matrix $\mathbf{A}$. (3) Furthermore, when λ increases from 0, the two vectors slowly deviate from being perpendicular, as shown in Figure 4.5b. Therefore, b_i is likely to be small if λ is close to 0.

2. Since we can not control 3D positions and there are typically small measurement errors in $\mathbf{d}_{ij}$, $\mathbf{b}$ does not exactly equal to 0. This can be regarded as a small disturbance of $\mathbf{b}$ around $\mathbf{0}$. For the linear system $\mathbf{A}\mathbf{s}^{\text{opt}} = \mathbf{b}$, one can think of the condition number $\kappa(\mathbf{A})$ as being (roughly) the rate at which the solution, $\mathbf{s}^{\text{opt}}$, will change with respect to a change in $\mathbf{b}$. $\kappa(\mathbf{A})$ is available because it solely depends on $\mathbf{r}_i$, $\mathbf{d}_{i,j}$ and λ, but not on the ground truth 3D points $\mathbf{X}^*$. Therefore, we can estimate the reliability of the reconstructed 3D points by computing $\kappa(\mathbf{A})$. Moreover, we empirically found that the condition number of matrix $\mathbf{A}$ is inversely related to λ. The condition number shown in Figure 4.6 is computed using 100 random cameras, and averaged over 200 trials. We can see $\kappa(\mathbf{A})$ is large if λ is close to 0 and drops dramatically with small λ. Then, $\kappa(\mathbf{A})$ decreases monotonically and slowly as λ increases. In our experiments, we choose $\lambda = \frac{1}{15}$ as a balance of having good chance of small $\mathbf{b}$ without decreasing the stability of the linear system.

In conclusion, given the well-conditioned system and correct motion tangent $\mathbf{d}_{i,j}$, we are able to reconstruct the 3D positions close to the ground truth.

4.3 Object Detector and Motion Tangent Estimation

Before presenting our experimental evaluation, we first briefly describe the particular object detector we use in our experiments. Single-image-based object detection is a well studied problem in computer vision with a wide variety of methods readily available (Zhang et al., 2006; Dalal and Triggs, 2005; Felzenszwalb et al., 2010). Similarly, there are a large number of motion tangent

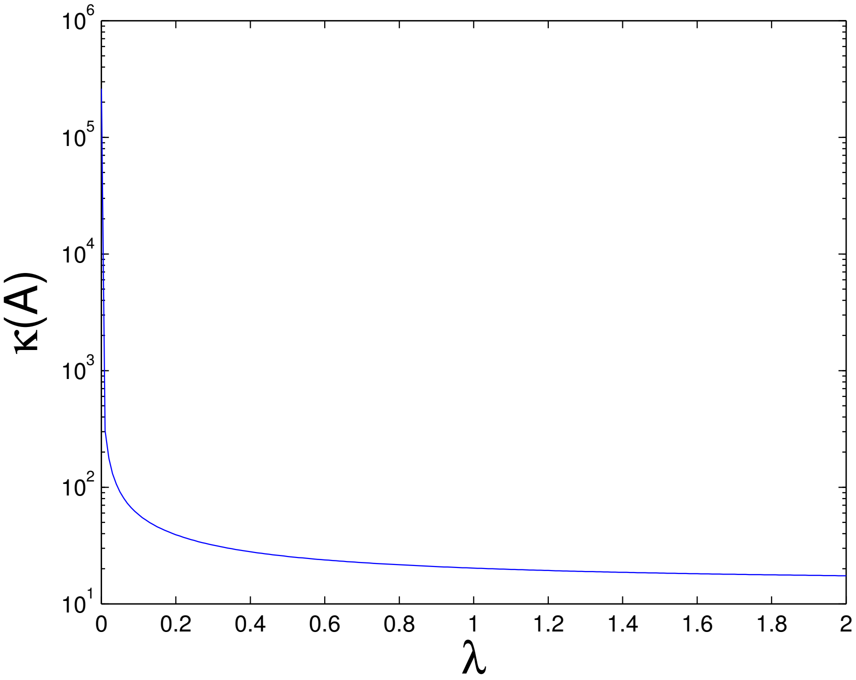


Figure 4.6: The condition number of the system $\kappa(\mathbf{A})$ increases as λ decreases.

estimation methods in the literature (Blanz and Vetter, 2003; Gu and Kanade, 2006; Jain and Learned-Miller, 2010; Jones and Viola, 2003).

For images containing large faces, we opt for leveraging the method that jointly determines the face position and its motion tangent direction (Zhu and Ramanan, 2012). In our experiments, the detection threshold is set to -0.35 to avoid false detections, as the false alarm may disturb our algorithm. Our chosen detectors provide a motion tangent of object i that is quantized every $\theta = 15^\circ$ in the range of -90° and 90°.

For cars and pedestrians with small faces in the images, we default to the deformable parts detector (Felzenszwalb et al., 2010; Girshick et al., 2012). We used the pre-trained model with detection threshold 0.35. The motion tangent of the pedestrians and cars are estimated using the 3D point cloud (output of VisualSFM) of the background wall by assuming the dynamic objects move parallel to the wall. This is normally true in Manhattan scenes. Some of the detection results are shown in Figure 4.7.

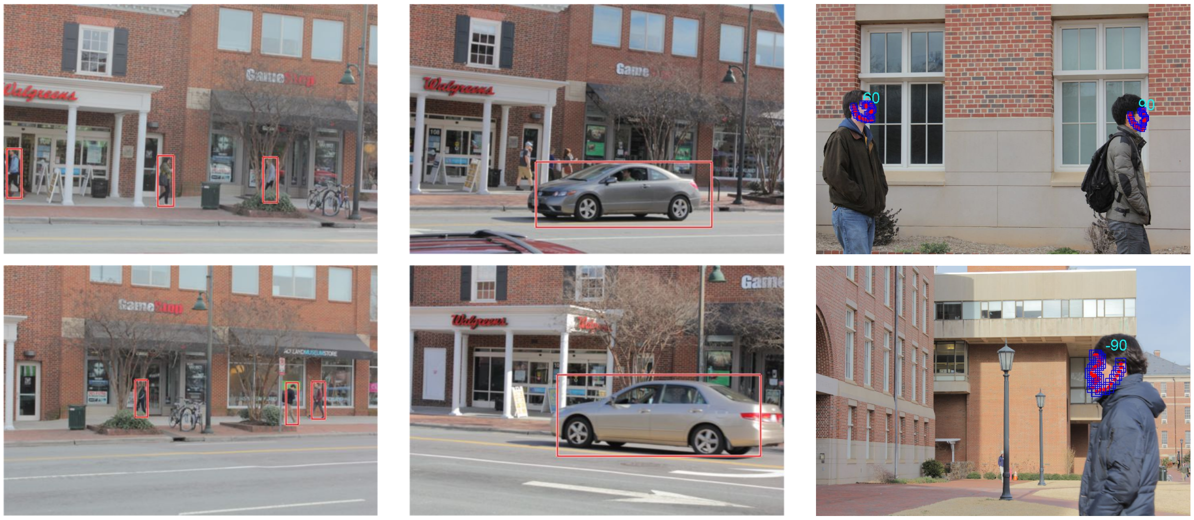


Figure 4.7: Detected objects and estimated motion tangents using different detectors.

	single line	T junction	double lines	half circle	sine wave	cross
error$_A$	0.5963	1.9688	1.5169	2.3751	2.3705	3.4111
error$^*_A$	0.4263	1.9148	1.4982	2.3340	2.3516	3.4030
error$_B$	0.2151	0.2126	0.7824	0.2281	0.2578	0.2251
error$^*_B$	0.0287	0.0944	0.7692	0.1074	0.2305	0.1308

Table 4.1: The table shows the average errors. The subscript represents camera setup. The absence of an asterisk represents the GMST algorithm output, and the asterisk is the refined output of Equation (4.7). Notice that for the ground truth 3D points, the average distance between every pair of nearest points equals 1.

4.4 Experiments

We evaluate our algorithm on both synthetic and real datasets. The GMST algorithm used in our method (Ferreira et al., 2012) searches the hypothesis space, which stops iterating when either the GMST cost is under a preset value, or the run time reaches a preset limit. For all experiments, we only use the time limit to stop searching, given the lack of an adequate *a priori* approximation of the true GMST cost for each dataset.

Synthetic datasets. Our first experiment uses synthetic data, with six different object class trajectory shapes on a plane, including a single line path, a T-junction path, a path with two parallel lines, a half circle path, a sine-wave-shaped path, and a cross-shaped path. To have a sense of the

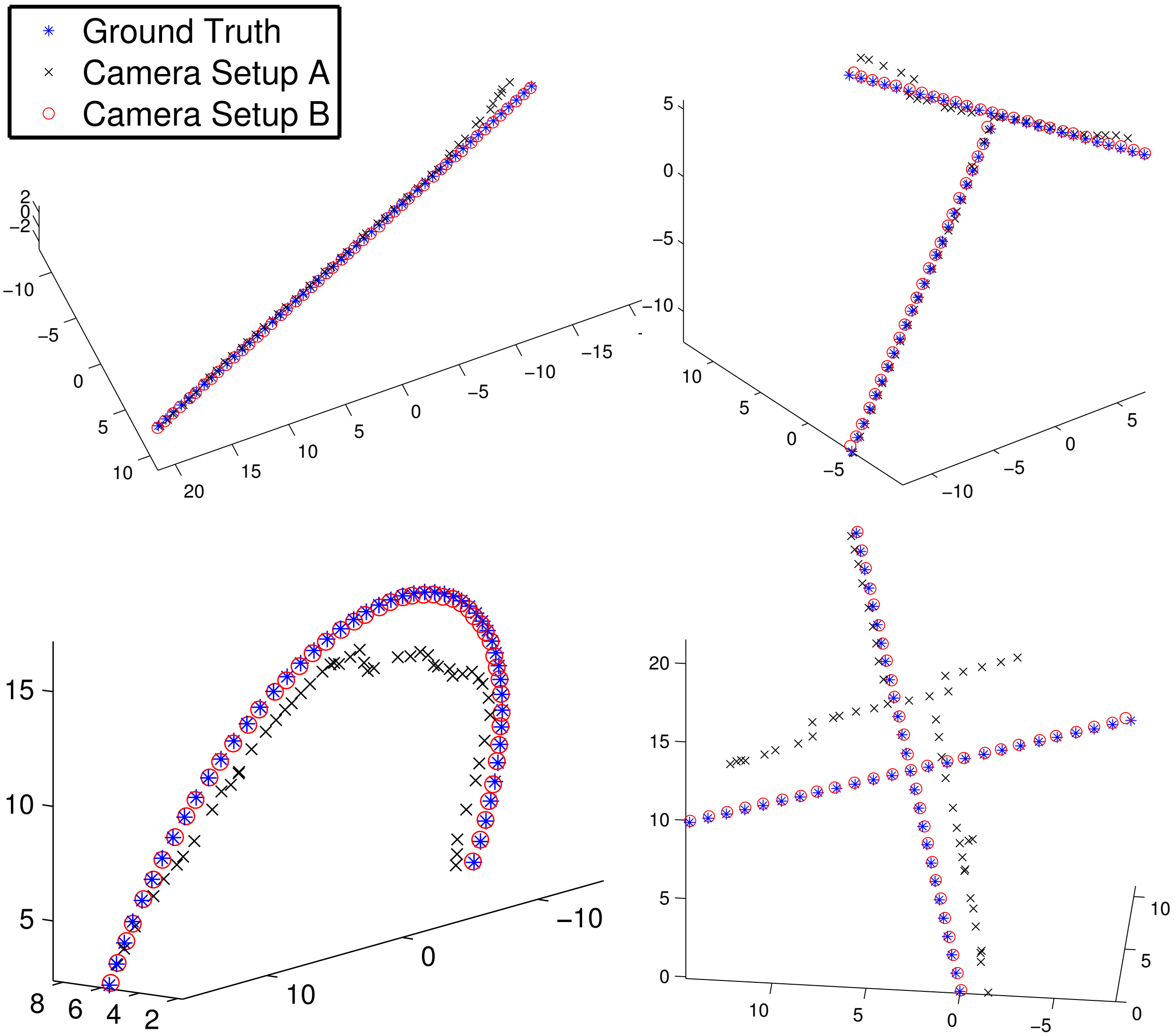


Figure 4.8: Example results for line path, T-junction path, half circle and crossed paths.

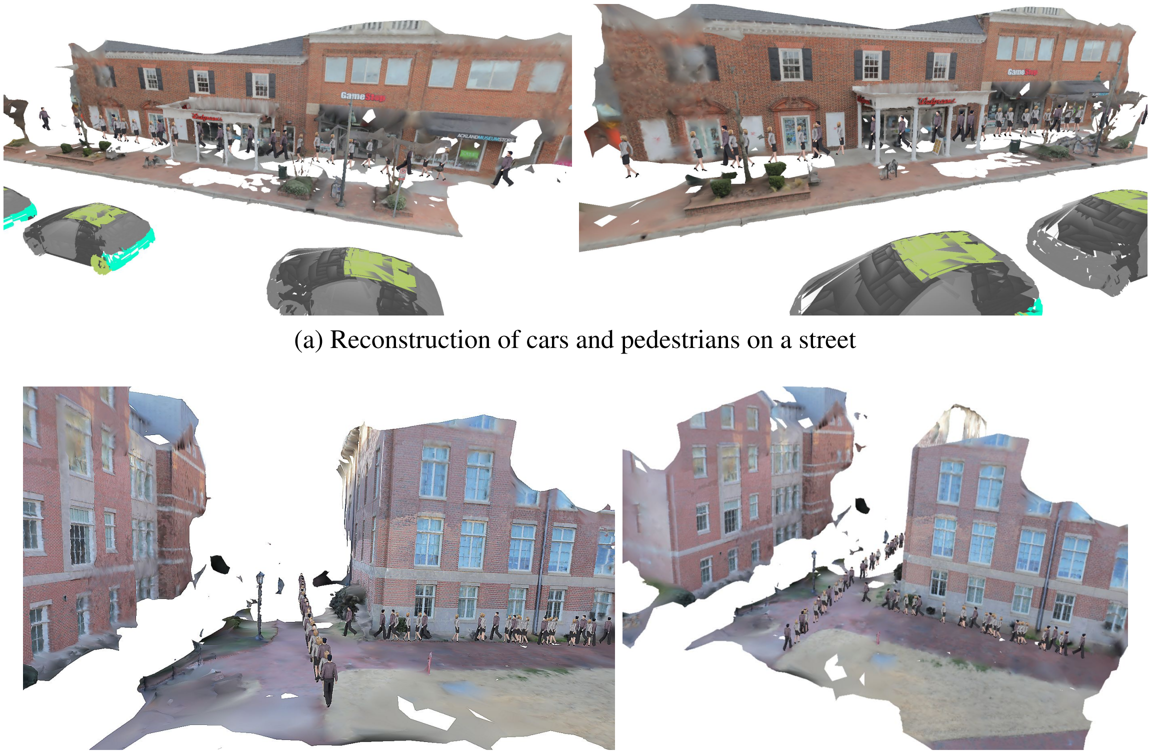
output errors, we normalize the 3D points so that the average distance between every pair of nearest points equals 1.

The virtual cameras are randomly generated around the 3D object points with two different configurations. In camera configuration A, all the camera centers stay in the same plane as the 3D points, which is more difficult since each viewing ray may intersect the ground truth path several times. In camera configuration B, the camera centers are set randomly off the plane, with the angle between the viewing ray and the plane being at most 10° and the camera distance being 2-3 times the length of the path.

We choose $k = 100$ uniformly distributed discrete 3D hypotheses $\mathbf{X}_i^o$ along each viewing ray $\mathbf{X}_i$ in a range that contains the ground truth 3D point. The size of the range is set as 1.5 times the length of the path. Notice that while the ground truth 3D point lie in the range for a given image, there is no guarantee that any of the discrete hypothesis samples $\mathbf{X}_i^o$ will exactly match the true depth.

Errors are measured using the Euclidean distance between the estimated 3D points and the ground truth. We run 32 instances for each shape with randomly generated virtual cameras. The average errors over the 32 instances for each shape category are listed in Table 4.1. We report both the errors of the GMST output and the errors after the continuous refinement using Equation (4.7). Table 4.1 shows our continuous refinement always improves the reconstruction accuracy over the GMST approximation. The results demonstrate *off-plane* cameras yield improved results than *in-plane* cameras for complex paths (e.g. crossed paths), due to the multiplicity of ray-to-path intersections. In these cases, the GMST solution has a more complex search space and yields a sub-optimal solution. However, the condition number of the linear system does not vary significantly across configurations. Figure 4.8 shows the estimated 3D points overlaid onto the ground truth.

Real datasets. We evaluate our method on two image datasets registered by VisualSFM (Wu, 2013). The detection confidence threshold is set high in order to decrease the false alarm rate. However, a very small amount of false alarms were purged manually, as they may affect the



(a) Reconstruction of cars and pedestrians on a street

(b) Reconstruction of people walking on a T-junction path

Figure 4.9: Two views for each of the reconstructed results.

reconstruction. We sample 100 samples along the viewing ray in the range $[0, far]$, where far is estimated using the model scale. The run time for each object class trajectory is set to 3 hours.

The first dataset captures random pedestrians walking on a sidewalk, plus random cars driving on an adjacent road. It contains 135 images with 82 valid car detections and 137 valid pedestrian detections. The scene and the reconstructed object class trajectory are shown in Figure 4.10. The second dataset captures several people who are walking on a T-junction shaped path at the corner of a building. It contains 47 images with 66 valid detections. Using the camera positions, we convert the face directions into the global coordinate system to obtain the motion tangents $\mathbf{d}_i$ of the moving people. For illustration, we construct the background static scene using CMPMVS (Jancosek and Pajdla, 2011). The general 3D human and car mesh models are inserted into each of the estimated 3D positions. We show different views of the reconstructed results in Figure 4.9.

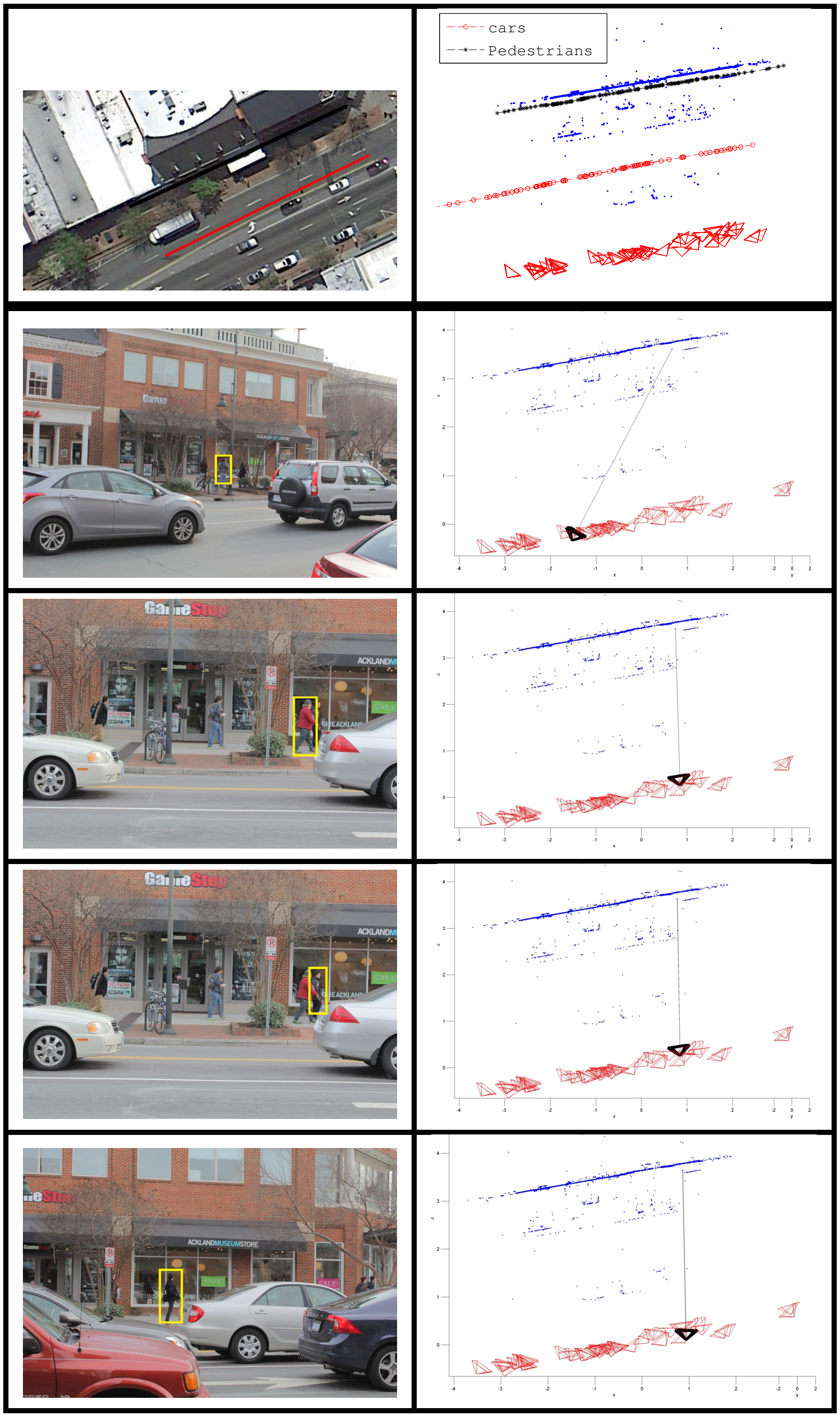


Figure 4.10: Top row: An aerial image showing the scene and a figure showing the cameras and reconstructed cars and pedestrians. Bottom four rows: Four pedestrian detections (shown in yellow rectangles) and the poses of the corresponding cameras. These four pedestrians are adjacent in the reconstructed object class trajectory. Notice that the second and the third images are the same image but with different detections.

4.5 Conclusion

We target the problem of reconstructing the 3D positions of dynamic objects from a set of unordered images, with the assumption that the objects of the same class move in a common path in the scene. We propose a framework of joint object class sequencing and trajectory triangulation and solve the associated non-convex optimization problem through a new discrete-continuous optimization scheme based on the generalized minimum spanning tree (GMST). The promising results on synthetic and real datasets demonstrate the solvability of the difficult problem and the effectiveness of our approach.

CHAPTER 5: SELF-EXPRESSIVE DICTIONARY LEARNING FOR DYNAMIC 3D RECONSTRUCTION

5.1 Introduction

Thanks to the rapid development of mobile technology, it has become common that many people use their own mobile cameras to capture a common event of interest, such as a concert or a wedding. These real-life videos and photos usually have the dynamic objects as the main focus of the scene. With the bursting growth of such crowd-sourced data, it is of interest to develop methods of dynamic object 3D reconstruction that enables understanding and visualization of the captured events.

In this work, we target the problem of dynamic 3D object reconstruction from multiple unsynchronized videos. More specifically, the method takes as input a collection of video streams without inter-sequence temporal information. The video streams could potentially have different, irregular, and unknown frame rates (see Figure 5.1). As output, the method reconstructs the 3D positions of sparse feature points at each time instance (*e.g.*, Figure 5.2). Dynamic object reconstruction from unsynchronized videos is a challenging problem due to various factors, such as unknown temporal overlap among video streams, possible non-concurrent captures, and dynamic object motion. Any of these factors impedes the valid reconstruction from traditional 3D triangulation, which relies on the assumption of concurrent captures or a static scene.

Despite the ubiquity of uncontrolled video collections, there are currently no methods that can successfully address our problem. Static scene reconstruction from photo collections has reached a high level of maturity (Snavely et al., 2006; Zheng et al., 2014a; Heinly et al., 2015) thanks to the development of structure from motion and depth estimation, but the reconstruction of dynamic objects using videos currently falls far behind the maturity of reconstruction of static scene elements. Existing methods of trajectory triangulation (Park et al., 2010; Valmadre and Lucey, 2012) from

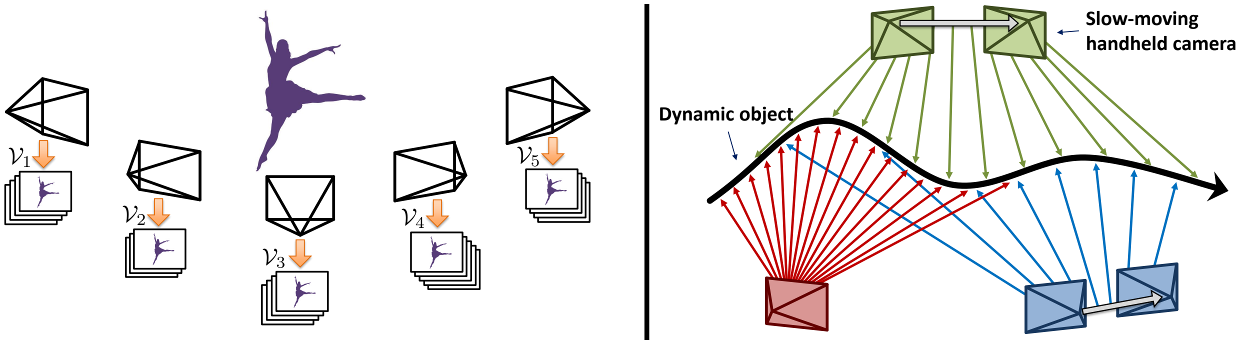


Figure 5.1: Left: Multiple videos capture a performance. The corresponding set of independent image streams serves as input to our method. Right: Each input video has a different sampling of a 3D point's trajectory.

monocular image sequences inherently require temporal order information (sequencing information). However, with independently captured videos, it is challenging to obtain this information across videos. In Chapter 4, we propose to jointly estimate the sequencing and 3D points by solving a generalized minimum spanning tree (GMST) problem. However, the NP-hard GMST problem itself limits the scalability of the approach. Also in this vein, the non-rigid structure from motion (NRSFM) problems have received extensive study over the two decades (Carlo and Takeo, 1992; Hartley and Vidal, 2008; Dai et al., 2014), but such methods are still under further exploration, especially if a perspective camera model is applied.

To solve the problem, we observe that, given the smooth motion of a dynamic object, any 3D shape at one time instance can be sparsely approximated by other shapes across time. Based on this self-expressive representation, our solution leverages the compressive sensing technique (l_1 norm), and tackles the problem in a dictionary learning framework (Aharon et al., 2006; Elad and Aharon, 2006), where the dictionary is defined by the temporally varying 3D structure. Though the self-expression technique has been previously used in subspace clustering for motion segmentation (Elhamifar and Vidal, 2009), and dictionary learning has been used in other applications such as image denoising (Elad and Aharon, 2006), we are the first to explore learning a self-expressive dictionary for the problem of dynamic object reconstruction.

The remainder of this chapter is organized as follows. After introducing the notations in Section 5.2, we begin describing foundations of our proposed approach in Section 5.3. Section 5.4 presents

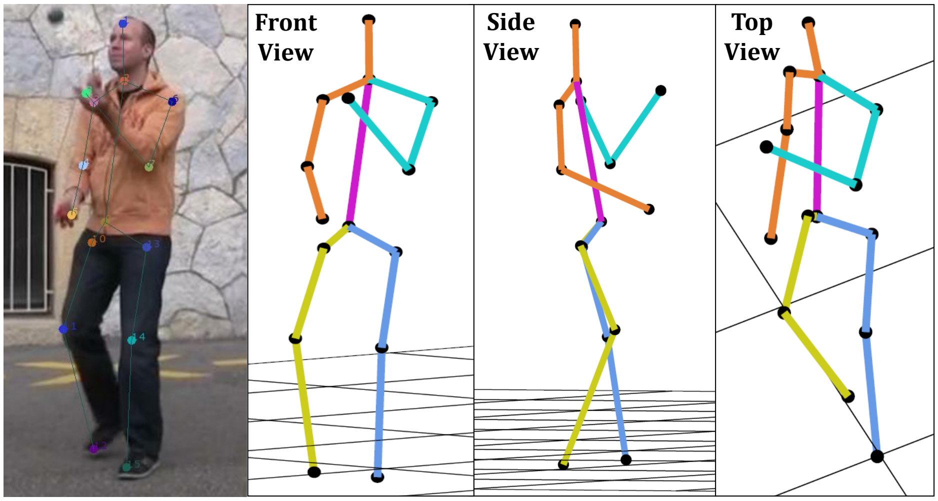


Figure 5.2: Example frame (left image) from the multiple videos capturing a performance serving as input to our method, with overlaid structure (points), and (right three images) different views of the reconstructed 3D points. Note our method only estimates the 3D points but no topology. The skeleton lines are plotted for visualization purposes.

our model for dynamic object reconstruction without sequencing information, followed by the parameterization of the 3D structure given different kinds of 2D measures in Section 5.5. Section 5.6 describes our ADMM-based optimization solver to minimize the model. Then, Section 5.7 illustrates the reconstructablity of our algorithm. We provide experimental evaluations in Section 5.8 and conclude the paper in Section 5.9.

5.2 Problem and Notation

We now describe the notations of our problem. Let $\mathcal{I}$ denote an aggregated set of images obtained from N video sequences $\mathcal{V}_n$, where $n = 1, \ldots, N$. Assuming a total of F available images, we can denote each individual image as $I_f \in \mathcal{I}$, where $f = 1, \ldots, F$. Alternatively, we can refer to the m-th frame in the n-th video as $I_{(n,m)} \in \mathcal{V}_n$, where $n = 1, \ldots, N$ and $m = 1, \ldots, |\mathcal{V}_n|$.

We assume an *a priori* camera registration through structure-from-motion analysis of static background structures within the environment (Wu, 2013). Accordingly, for each available image I_f, we know the capturing camera's pose matrix $\mathbf{M}_f = [\mathbf{R}_f \mid -\mathbf{R}_f\mathbf{C}_f]$, along with its intrinsic camera matrix $\mathbf{K}_f$.

Without loss of generality, we first assume each image I_f captures a common set of P 3D points $\{\mathbf{X}_{(p,f)} \mid p = 1, \ldots, P\}$, and the 2D measure of each point is denoted as $\mathbf{x}_{(p,f)}$. We also assume the correspondences of image measures $\mathbf{x}_{(p,f)}$ across images are available. Then for each measure $\mathbf{x}_{(p,f)}$, we can compute a viewing ray with direction by

$$\mathbf{r}_{(p,f)} = \mathbf{R}_f^{\mathrm{T}} \mathbf{K}_f^{-1} \begin{bmatrix} \mathbf{x}_{(p,f)} \\ 1 \end{bmatrix}, \tag{5.1}$$

and followed by a normalization into a unit vector.

Hence, the position of the dynamic 3D point $\mathbf{X}_{(p,f)}$ corresponding to $\mathbf{x}_{(p,f)}$ can be described by the distance along the viewing ray $\mathbf{r}_{(p,f)}$ given by

$$\mathbf{X}_{(p,f)} = \mathbf{C}_f + d_{(p,f)} \mathbf{r}_{(p,f)}, \tag{5.2}$$

where $d_{(p,f)}$ is the unknown distance of the 3D point from the camera center.

Given F frames with each frame observing P dynamic 3D points, we denote our aggregated observed 3D datum as

$$\mathbb{X} = \begin{bmatrix} \mathbf{X}_{(1,1)} & \cdots & \mathbf{X}_{(1,F)} \\ \vdots & \ddots & \vdots \\ \mathbf{X}_{(P,1)} & \cdots & \mathbf{X}_{(P,F)} \end{bmatrix} = [\mathbf{S}_1 \quad \cdots \quad \mathbf{S}_F], \tag{5.3}$$

where the f-th column of the matrix $\mathbb{X}$, denoted as $\mathbf{S}_f$, is obtained by stacking all the P 3D points observed in the f-th frame.

Then by defining $\mathbb{C}$, $\mathbb{r}$, and $\mathbb{d}$ as follows,

$$\mathbb{C} = \begin{bmatrix} \mathbf{C}_1 & \cdots & \mathbf{C}_F \end{bmatrix}, \tag{5.4}$$

$$\mathbb{r} = \begin{bmatrix} \mathbf{r}_{(1,1)} & \cdots & \mathbf{r}_{(1,F)} \\ \vdots & \ddots & \vdots \\ \mathbf{r}_{(P,1)} & \cdots & \mathbf{r}_{(P,F)} \end{bmatrix}, \tag{5.5}$$

$$\mathbb{d} = \begin{bmatrix} d_{(1,1)} & \cdots & d_{(1,F)} \\ \vdots & \ddots & \vdots \\ d_{(P,1)} & \cdots & d_{(P,F)} \end{bmatrix}, \tag{5.6}$$

Equation (5.2) for all the points can be rewritten in matrix form as

$$\mathbb{X} = \mathbf{1}_{P\text{x}1} \otimes \mathbb{C} + (\mathbb{d} \otimes \mathbf{1}_{3\text{x}1}) \odot \mathbb{r}, \tag{5.7}$$

where $\mathbf{1}_{P\text{x}1}$ is a P-by-1 matrix with values equal to 1, $\otimes$ is the Kronecker product, and $\odot$ is the component-wise matrix product.

Our task is to recover $\mathbb{X}$ from the 2D measures without image sequencing information across the videos.

5.3 Principle

The key observation driving our approach is that dynamic shape exhibits temporal coherence. In this section, we demonstrate how this principle can be leveraged to recover local temporal ordering with known shapes. Our proposed method will extend these ideas to situations with unknown structures.

For our method, we assume a smooth 3D motion under the sampling provided by the videos. Hence, we can approximate the 3D structure $\mathbf{S}_f$ observed in image f in terms of a linear combination of the structures corresponding to the set of immediately preceding ($\mathbf{S}_{prev}$) and succeeding ($\mathbf{S}_{next}$) frames in time. That is, we have

$$\mathbf{S}_f \approx w \cdot \mathbf{S}_{prev} + (1 - w) \cdot \mathbf{S}_{next}, \tag{5.8}$$

with $0 \leq w \leq 1$. If our structure matrix $\mathbb{X}$ from Equation (5.3) was temporally ordered, which it is not in general, the two neighboring frames would be $\mathbf{S}_{f-1}$ and $\mathbf{S}_{f+1}$. Clearly, such perfect temporal order can be extracted from a single video sequence. However, the reconstructability constraints make single-camera structure estimation ill-posed (see Section 5.7.2 for details). Hence, *we rely on inter-sequence temporal ordering information to solve the dynamic structure estimation problem.* The absence of a global temporal ordering requires us to search for temporal adjacency relations across the different video streams having potentially different frame rates.

In the most simple scenario, the pool of candidate neighboring frames is comprised by all other frames except f. Writing the 3D points of the current frame $\mathbf{S}_f$ as a linear combination of other frames, we have

$$\mathbf{S}_f = \mathbb{X}\mathbf{W}_f, \tag{5.9}$$

where $\mathbf{W}_f = \left(w_{(1,f)}, \ldots, w_{(f-1,f)}, 0, w_{(f+1,f)}, \ldots, w_{(F,f)}\right)^{\mathrm{T}}$ is a vector of length F representing the coefficients for the linear combination. Note that the f-th element in $\mathbf{W}_f$ equals 0, since the f-th column of $\mathbb{X}$ (corresponding to $\mathbf{S}_f$) is not used as an element of the linear combination.

Moreover, since only a few shapes in the close temporal neighborhood of $\mathbf{S}_f$ are likely to provide a good approximation, we expect the vector $\mathbf{W}_f$ to be sparse. Accordingly, we propose to find the local temporal neighborhood of a shape $\mathbf{S}_f$ through a compressive sensing formulation leveraging the l_1 norm:

$$\underset{\mathbf{W}_f}{\text{minimize}}\ ||\mathbf{S}_f - \mathbb{X}\mathbf{W}_f||_2^2 + \lambda||\mathbf{W}_f||_1, \tag{5.10}$$

where λ is a positive weight. Here, the l_1 norm serves as an approximation of the l_0 norm and favors the attainment of sparse coefficient vectors $\mathbf{W}_f$ (Bach et al., 2012). Moreover, we incorporate the desired properties of our linear combination framework (Equation (5.8)) and reformulate Equation (5.10) as

$$\begin{aligned} \underset{\mathbf{W}_f}{\text{minimize}} \quad & ||\mathbf{S}_f - \mathbb{X}\mathbf{W}_f||_2^2 \\ \text{subject to} \quad & \mathbf{W}_f \cdot \mathbf{1}_{F\times 1} = 1 \\ & \mathbf{W}_f \geq 0. \end{aligned} \tag{5.11}$$

The affine constraints of Equation (5.11) constrain the variable $\mathbf{W}_f$ to reside in the simplex Δ_f defined as

$$\Delta_f \triangleq \{\mathbf{W}_f \in \mathbb{R}^F \text{ s.t. } \mathbf{W}_f \geq 0, w_{(f,f)} = 0 \text{ and } \sum_{j=1}^{F} w_{(j,f)} = 1\}. \tag{5.12}$$

Despite the lack of an explicit l_1-norm regularization term in Equation (5.11), as a variant of compressive sensing, the formulation still keeps the sparsity-inducing effect (Bach et al., 2012; Chen et al., 2014). This is true for the present problem, since we know a shape can be well represented by temporally close shapes. A similar formulation has been used in modeling archetypal analysis for representation learning (Chen et al., 2014). There, the authors also provide a new efficient solver for this kind of problem.

Finally, we generalize our formulation from Equation (5.11) to include all available structure estimates $\mathbf{S}_f$, with $f = 1, \ldots, F$, into the following equation

$$\begin{aligned} &\underset{\mathbb{W}}{\text{minimize}} \quad ||\mathbb{X} - \mathbb{X}\mathbb{W}||_{\mathrm{F}}^2 \\ &\text{subject to} \quad \mathbf{W}_f \in \Delta_f, f = 1, \cdots, F, \end{aligned} \tag{5.13}$$

where $|| \cdot ||_{\mathrm{F}}$ denotes the Frobenius norm and $\mathbb{W} = [\mathbf{W}_1 \ldots \mathbf{W}_F]$ is an $F \times F$ matrix with the f-th column equal to $\mathbf{W}_f$. By construction, $\mathbb{W}$ has all its diagonal elements equal to zero.

As an illustration of the validity of our compressed sensing formulation, Figure 5.3 shows the output of Equation (5.13) on a real motion capture dataset given known 3D points $\mathbb{X}$. Although image sequencing is assumed unknown, we show results in temporal order for visualization purposes. The coefficients in $\mathbb{W}$ approximate a matrix having non-vanishing values only on the locations directly above and below the main diagonal. This indicates that the 3D points $\mathbf{S}_f$ are a linear combination of $\mathbf{S}_{f-1}$ and $\mathbf{S}_{f+1}$.

Minimizing Equation (5.11) is equivalent to finding the most related shapes to linearly represent $\mathbf{S}_f$. It is usually true that the temporally close shapes $\mathbf{S}_{f-1}$ and $\mathbf{S}_f$ are most related, and therefore local temporal information is recoverable from the non-vanishing values in $\mathbb{X}$. However, if object motion is repetitive or if the object is static for a period of time, there is no guarantee that the most

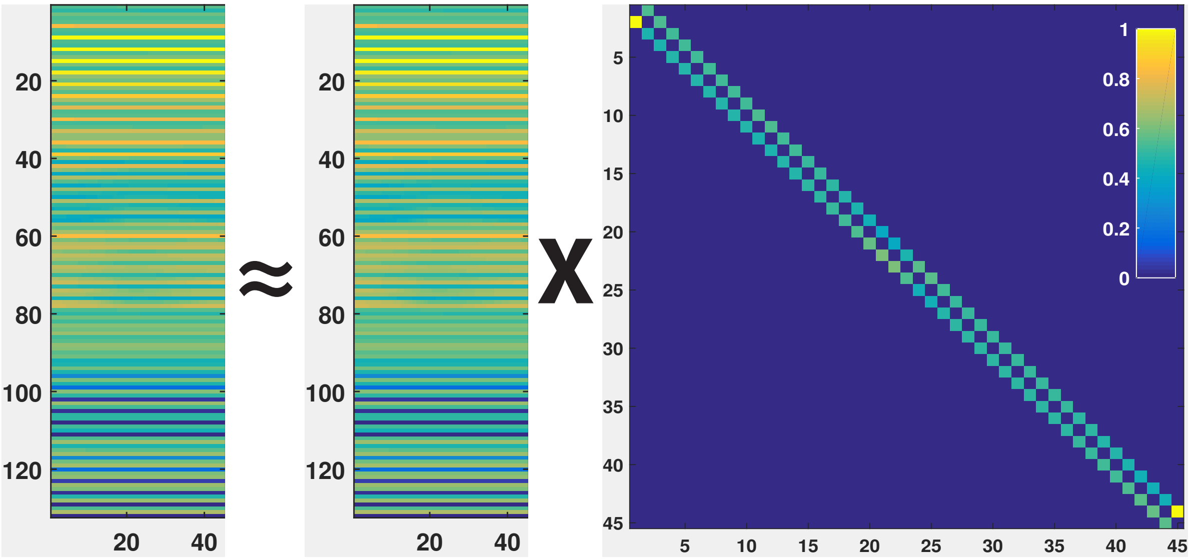


Figure 5.3: We illustrate the output of Equation (5.13) on a real motion capture dataset “Clap1Rep”. For easy visualization, the shortest motion capture dataset (45 frames) presented in the work by Müller et al. (2007) is used. Each element/column in $\mathbb{X}$ corresponds to ground truth 3D structure. The estimation of $\mathbb{W}$ through Equation (5.13) approximates the correct ordering after enforcing all elements in the diagonal to be 0.

related shapes are the temporally closest ones. Even though this is true, the analysis in Section 5.7.3 shows that this does not cause any problem for our method in regard to 3D reconstruction.

To validate our prior of sparse representation for real motion, we quantitatively evaluate the estimated coefficients $\mathbb{W}$ by minimizing Equation (5.13) on all 130 real motion capture datasets presented in the work by Müller et al. (2007). For a shape at a given time sample, we measure the sum of the two largest estimated coefficient values for this sample, and the frequency with which these top two coefficients correspond to the ground truth temporally neighboring shape samples. Given our prior, values of 1 for both measures are expected. The average values we obtain are 0.9972 and 0.9994, supporting the validity of our prior.

5.4 Method

We address the problem of estimating sparse dynamic 3D structure from a set of spatially registered video sequences with unknown temporal overlap. Section 5.3 presented a compressive sensing formulation leveraging the self-expressiveness of all the shapes in the context of known 3D geometry. However, our goal is to estimate the unknown structure without sequencing information. To this end, we define our dictionary as the temporally varying 3D structure and propose a compressive sensing framework which poses the estimation of 3D structure as a dictionary learning problem. We solve this problem in an iterative and alternating manner, where we optimize for 3D structure while fixing the sparse coefficients, and *vice versa*. This is achieved through the optimization of a biconvex cost function that leverages the compressed sensing formulation described in Section 5.3 and, additionally, enforces both structural dependence coherence across video streams and motion smoothness among estimates from common video sources.

5.4.1 Cost Function

To achieve the stable estimation of both the structure $\mathbb{X}$ and the sequencing information $\mathbb{W}$, we extend our formulation from Equation (5.13) to the following cost function:

$$\begin{aligned} \underset{\mathbb{X},\mathbb{W}}{\text{minimize}} \quad & \frac{1}{FP}||\mathbb{X} - \mathbb{X}\mathbb{W}||_{\mathrm{F}}^2 + \lambda_1\Psi_1(\mathbb{W}) + \lambda_2\Psi_2(\mathbb{X}) \\ \text{subject to} \quad & \mathbf{W}_f \in \Delta_f, f = 1, \cdots, F; \end{aligned} \tag{5.14}$$

where $\Psi_1(\mathbb{W})$ and $\Psi_2(\mathbb{X})$ are two convex cost terms regulating the spatial relationships between 3D observations within and across video streams. We also add the normalization term FP to cancel the influence of number of frames and number of points per shape. Next, we describe each of the cost terms in detail.

5.4.2 Dictionary Space Reduction in Self-representation

The first cost term in Equation (5.14) serves to find shapes in the dictionary to sparsely represent each shape. The search space can be reduced if some elements of $\mathbb{W}$ are forced to be 0. As mentioned, the diagonal elements of $\mathbb{W}$ are forced to be 0, since a shape is not used to represent itself. Moreover, it is possible that if *a priori* knowledge of rough temporal information across video steams is available, we can also leverage that knowledge to reduce the search space.

In our solution, we explicitly enforce that the shape observed by one video is not used to represent the shape observed in the same video, because the reconstructibility analysis in Section 5.7.2 shows such estimation is ill-posed. In our implementation, enforcing this constraint is achieved by not defining the corresponding variables in $\mathbb{W}$ during the optimization.

5.4.3 Coefficient Relationships: $\Psi_1(\mathbb{W})$

As described in Section 5.3, a given structure $\mathbf{S}_f$ in frame f can be obtained from the linear combination of the 3D shapes captured in other frames. The coefficients or weights of the linear combination are given by the elements of the matrix $\mathbb{W}$. In particular, the element in the j-th row and f-th column of $\mathbb{W}$ is denoted as $w_{(j,f)}$, and it describes the relative contribution (weight) from $\mathbf{S}_j$ in estimating $\mathbf{S}_f$. Similarly, $w_{(f,j)}$ represents the contribution of $\mathbf{S}_f$ towards the 3D points in $\mathbf{S}_j$. Accordingly, a value of $w_{(f,j)} = 0$ indicates the absence of any contribution from $\mathbf{S}_f$ to $\mathbf{S}_j$, which is desired for tempo-spatially non-proximal 3D shapes.

We note that, if $\mathbf{S}_f$ contributes to $\mathbf{S}_j$, it means the two sets of points are highly correlated, which further implies that $\mathbf{S}_j$ should reciprocally contribute to estimating $\mathbf{S}_f$. We deem this reciprocal influence within our estimation process as *structural dependence coherence* and develop a cost term that contributes toward enforcing this property within the estimation of $\mathbb{W}$. We encode this relationship into our cost function as an additional term of the form

$$\Psi_1(\mathbb{W}) = \frac{1}{F}||\mathbb{W} - \mathbb{W}^\top||_{\mathrm{F}}^2. \tag{5.15}$$

A strict interpretation of the above formulation aims to identify symmetric matrices. In general, the reciprocal influence between $\mathbf{S}_f$ and $\mathbf{S}_j$ does not imply symmetric contribution, as the values of $w_{(f,j)}$ and $w_{(j,f)}$ depend on the actual 3D motion being observed. More specifically, these values describe the linear structural dependencies between two different, but overlapping, 3-tuples of 3D points, *e.g.* ($\mathbf{S}_i$,$\mathbf{S}_f$,$\mathbf{S}_j$) and ($\mathbf{S}_f$,$\mathbf{S}_j$,$\mathbf{S}_k$) as illustrated in Figure 5.4. In the toy example of Figure 5.4, it can be seen that $\mathbf{S}_i$ and $\mathbf{S}_j$ are at equal distance to $\mathbf{S}_f$ and hence equally contribute to

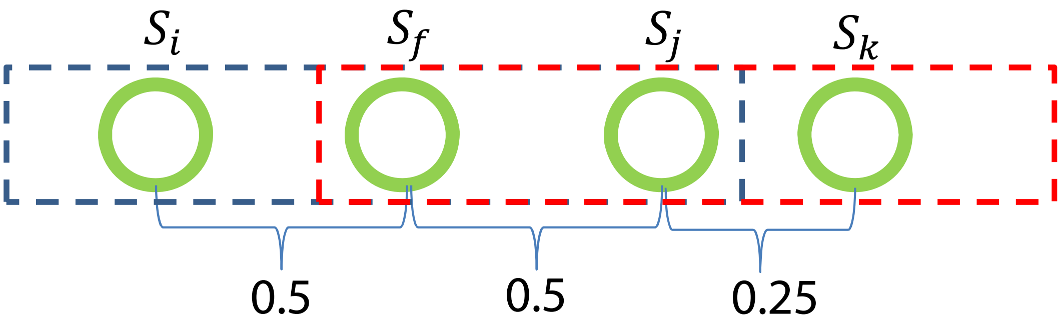


Figure 5.4: Illustration of the triplets influencing the weights for $\mathbf{S}_f$ and $\mathbf{S}_j$ leading to an asymmetric $\mathbb{W}$. The values in the figure represent the distance between adjacent points.

it, *i.e.* $w_{(i,f)} = w_{(j,f)} = \frac{1}{2}$. However, in order to determine the linear combination weights for specifying $\mathbf{S}_j$, we need to consider $\mathbf{S}_f$ and $\mathbf{S}_k$. Here, $\mathbf{S}_f$ is twice as far from $\mathbf{S}_j$ as $\mathbf{S}_k$, and thus $w_{(f,j)} = \frac{1}{3}$, which is lower than $w_{(j,f)}$. Accordingly, we do not expect a fully symmetric weight matrix $\mathbb{W}$. However, given our expectation of a sparse coefficient matrix $\mathbb{W}$, we can focus on finding congruence between the zero-value elements of the $\mathbb{W}$ and $\mathbb{W}^\top$, which $\Psi_1(\mathbb{W})$ effectively encodes. Moreover, $\Psi_1(\mathbb{W})$ is convex, which enables its use within our biconvex optimization framework.

5.4.4 Sequencing Information: $\Psi_2(\mathbb{X})$

Under the assumption of sufficiently smooth 3D motion w.r.t. the frame-rate of each video capture, we define a 3D spatial smoothness term that penalizes large displacements among successive frames from the same video. Therefore, we define a pairwise term over the values of $\mathbb{X}$

$$\Psi_2(\mathbb{X}) = \frac{1}{M} \sum_{n=1}^{N} \sum_{m=1}^{|\mathcal{V}_n|-1} \left|\left|\mathbf{X}_{(n,m)} - \mathbf{X}_{(n,m+1)}\right|\right|_2^2, \tag{5.16}$$

where n is the video index, m is the image index within a video, $|\mathcal{V}_n|$ denotes the number of video frames within each sequence, and $M = \sum_{n=1}^{N}(|\mathcal{V}_n| - 1)$ is a normalization factor. Note that $\Psi_2(\mathbb{X})$ does not explicitly enforce ordering information across video sequences, but instead fosters a compact 3D motion path within a sequence. Moreover, $\Psi_2(\mathbb{X})$ is a convex term.

However, this regularization term $\Psi_2(\mathbb{X})$ is a double-edged sword. Since this term minimizes the sum-of-squared distances, if a video camera is static or has small motion, the estimated 3D points are likely to be pulled towards the camera center. This typically biases the estimated 3D points slightly away from their real positions. Therefore, we propose to first minimize Equation (5.14) until convergence to obtain values for $\mathbb{X}$ and $\mathbb{W}$, and then taking those values as initialization, we further optimize the problem with weight of $\Psi_2(\mathbb{X})$ (*i.e.* λ_2) set to 0.

5.5 Parameterization of $\mathbb{X}$

Given accurate 2D measurements, the 3D structures $\mathbb{X}$ are constrained to lie on the viewing rays defined by the 2D measures and camera poses. Therefore, we can use Equation (5.7) to represent $\mathbb{X}$. This is deemed as a hard constraint, as the points have to lie on the viewing ray. However, in practice, the measures are typically noisy or unavailable due to, for example, inaccurate feature detection or motion blur. Next, we discuss the parameterization of $\mathbb{X}$ given noisy and missing 2D observations.

5.5.1 Noisy Observations

The parameterization using Equation (5.2) enforces the hard constraint that 3D points lie on the viewing rays. Given that this may not be appropriate under the circumstance of noisy measurements, we can change this hard constraint to a soft constraint by adding a regularization term into the original Equation (5.14). Defining the objective function in Equation (5.14) as $\Phi(\mathbb{X}, \mathbb{W})$, we propose a revised version as

$$\begin{aligned} \underset{\mathbb{X},\mathbb{W},\mathbb{d}}{\text{minimize}} \quad & \Phi(\mathbb{X}, \mathbb{W}) + \lambda_3 ||\mathbf{1}_{P\text{x}1} \otimes \mathbb{C} + (\mathbb{d} \otimes \mathbf{1}_{3\text{x}1}) \odot \mathbb{r} - \mathbb{X}||_{\text{F}}^2 \\ \text{subject to} \quad & \mathbf{W}_f \in \Delta_f, f = 1, \cdots, F. \end{aligned} \tag{5.17}$$

The formulation converts the hard constraint of Equation (5.7) as a soft constraint by adding a penalization if the 3D points deviate away from the viewing ray. The value of λ_3 controls how much a point can deviate away from the viewing ray, and it depends on the noise level of the 2D observations. A larger value should be used when the level of noise is lower. Note the new formulation is the same to the hard constraint if the weight λ_3 is set to ∞. Moreover, in Equation

(5.17), $\mathbb{d}$ is an auxiliary variable solely depending on $\mathbb{X}$. More details about the optimization of Equation (5.17) are presented in Section 5.6.1.

5.5.2 Missing Data

Each 3D point, given its accurate 2D measurement, lies on the corresponding viewing ray. Hence, the 3D point has one degree of freedom – depth along the viewing ray. However, in the absence of 2D observations, which can happen in the case of occlusion, the 3D points are no longer constrained by the viewing ray and thus have three degrees of freedom.

In our method, the 3D points with missing 2D observations are interpolated by the estimated linear coefficients $\mathbb{W}$. Therefore, this scheme is likely to produce larger errors if a dynamic 3D point is not observed by multiple consecutive frames across time. In our experiments, we test the accuracy of our algorithm under different missing-data rates.

5.6 Optimization

The biconvex function in Equation (5.14) is non-convex, but it is convex if one set of the variables $\mathbb{X}$ or $\mathbb{W}$ is fixed. The optimization scheme employed for Eq. (5.14) alternates the optimizations over $\mathbb{X}$ and $\mathbb{W}$. We preferred this approach due to its relative simplicity over elaborate dictionary update schemes such as K-SVD (Aharon et al., 2006). Nevertheless, since the alternating optimization steps need to be performed until convergence, each step must be reasonably fast. Although optimizing over $\mathbb{X}$ is easy, optimizing over $\mathbb{W}$ is relatively more difficult due to the simplicial constraint. We find that optimizing over $\mathbb{W}$ with a general solver, such as CVX (Grant and Boyd, 2014), is too slow even for a moderate number of frames F. Moreover, during our iterative optimization, the output of the previous step can be fed into the current step for better initializaiton (hot start), but typical general solvers, such as those based on the interior point algorithm, do not allow for a hot start. To solve the problem with speed and scalability, we propose a new solver based on alternating direction method of multipliers (ADMM) (Boyd et al., 2011).

5.6.1 Optimize Over $\mathbb{X}$

If $\mathbb{W}$ in Equation (5.14) is fixed, the optimization over $\mathbb{X}$ is straightforward, as the problem is quadratic programming without any constraint, regardless of the difficulties discussed in Section 5.5.

1. If the data are noise-free, we can substitute Equation (5.7) into Equation (5.14), and obtain a quadratic programming problem without any constraint on the unknown variable $\mathbb{d}$.

2. In the case of noisy measurements, $\mathbb{d}$ are dependent on $\mathbb{X}$. More specifically, $d_{(p,f)}$ is given by

$$d_{(p,f)} = (\mathbf{X}_{(p,f)} - \mathbf{C}_f)^{\mathrm{T}} \mathbf{r}_{(p,f)}, \tag{5.18}$$

   i.e. the projection of $\mathbf{X}_{(p,f)} - \mathbf{C}_f$ onto the viewing ray. Then, after replacing $\mathbb{d}$ with $\mathbb{X}$, we obtain a quadratic programming problem over unknown $\mathbb{X}$.

3. For the case of missing observations, the corresponding 3D points are unknown variables. Therefore, for a given miss rate, the problem is quadratic over some unknown variables both in $\mathbb{d}$ and in $\mathbb{X}$.

For the quadratic programming without constraints, the solution can be found at the zero value of the derivative of the cost function over the unknown variables.

5.6.2 Optimize Over $\mathbb{W}$

The optimization over $\mathbb{W}$ is more complex mainly due to the simplex constraints. By fixing the variable $\mathbb{X}$ in Equation (5.14), the cost function becomes,

$$\begin{aligned} \underset{\mathbb{W}}{\text{minimize}} \quad & \frac{1}{FP}||\mathbb{X} - \mathbb{X}\mathbb{W}||_{\mathrm{F}}^2 + \frac{\lambda_1}{F}||\mathbb{W} - \mathbb{W}^\top||_2^2 \\ \text{subject to} \quad & \mathbf{W}_f \in \Delta_f, f = 1, \cdots, F. \end{aligned} \tag{5.19}$$

Notice that if the term $||\mathbb{W} - \mathbb{W}^\top||_{\mathrm{F}}^2$ vanishes, the cost function is the same to Equation (5.13), which can be decomposed into Equation (5.11), and optimized over $\mathbf{W}_f$ for each $f = 1, \ldots, F$ independently. Advantageously, the number of variables for each subproblem is much smaller compared to the total number of variables in $\mathbb{W}$, and it can be parallelized on the level of subproblems. Moreover, Chen et al. (2014) propose a fast solver to the optimization problem in Equation (5.11) based on an active-set algorithm that can benefit from the solution sparsity. However, the cost term $||\mathbb{W} - \mathbb{W}^\top||_{\mathrm{F}}^2$ prevents the decomposition.

In this work, we propose an ADMM algorithm that enables the decomposition. By introducing a new auxiliary variable $\mathbb{Z}$, Equation (5.19) can be rewritten as

$$\begin{aligned} \underset{\mathbb{W}}{\text{minimize}} \quad & \frac{1}{FP}||\mathbb{X} - \mathbb{X}\mathbb{W}||_{\mathrm{F}}^2 + \frac{\lambda_1}{F}||\mathbb{Z} - \mathbb{Z}^\top||_{\mathrm{F}}^2 \\ \text{subject to} \quad & \mathbf{W}_f \in \Delta_f, f = 1, \cdots, F \\ & \mathbb{W} = \mathbb{Z}. \end{aligned} \tag{5.20}$$

Though this change may seem trivial, the objective function is now separated in $\mathbb{W}$ and $\mathbb{Z}$. The ADMM technique allows this problem to be solved approximately by first solving for $\mathbb{W}$ with $\mathbb{Z}$ fixed, then solving for $\mathbb{Z}$ with $\mathbb{W}$ fixed, and next proceeding to update a dual variable $\mathbb{Y}$ (introduced below). This three-step process is repeated until convergence. Next, we describe each step of our ADMM-based algorithm.

In step 1, $\mathbb{W}$ is updated by

$$\mathbb{W}^{k+1} = \underset{\mathbf{T}_f \in \Delta_f, \,\text{for}1 \leq f \leq F}{\text{argmin}} \frac{1}{FP}||\mathbb{X} - \mathbb{X}\mathbb{W}||_{\mathrm{F}}^2 + \text{vec}(\mathbb{Y}^k)^\top \text{vec}(\mathbb{W}) + \frac{\rho}{2}||\mathbb{W} - \mathbb{Z}^k||_{\mathrm{F}}^2, \tag{5.21}$$

where the superscript k is the iteration index. $\mathbb{Y}^k$ is the matrix of dual variables and is initialized with 0. Note that the values of $\mathbb{Y}^k$ and $\mathbb{Z}^k$ are known during this step – we only optimize over the variable $\mathbb{W}$. The optimization can be decomposed into optimizing over $\mathbf{W}_f$ independently and in parallel, and we employ the fast solver proposed by Chen et al. (2014).

In step 2, we update the auxiliary variable $\mathbb{Z}$ according to

$$\mathbb{Z}^{k+1} = \underset{\mathbb{Z}}{\operatorname{argmin}} \frac{\lambda_1}{F}||\mathbb{Z} - \mathbb{Z}^\top||_{\mathrm{F}}^2 - \mathrm{vec}(\mathbb{Y}^k)^\top \mathrm{vec}(\mathbb{Z}) + \frac{\rho}{2}||\mathbb{W}^{k+1} - \mathbb{Z}||_{\mathrm{F}}^2. \tag{5.22}$$

This is a quadratic programming problem in the unknown variable $\mathbb{Z}$ without constraint and can be easily solved by setting the derivative of Equation (5.22) with respect to $\mathbb{Z}$ equal to 0.

In step 3, the dual variables $\mathbb{Y}$ are updated directly by

$$\mathbb{Y}^{k+1} = \mathbb{Y}^k + \rho(\mathbb{W}^{k+1} - \mathbb{Z}^{k+1}). \tag{5.23}$$

The three Equations (5.21), (5.22), and (5.23) iterate until the stop criterion is met. We use the stop criterion described by Boyd et al. (2011).

5.6.3 Initialization of the Optimization

Given the non-convexity of our original cost function (Equation (5.14)), the accuracy of our estimates is sensitive to the initialization values used by our iterative optimization. Hence, we design a 3D structure (*i.e.*, $\mathbb{X}$) initialization mechanism aimed at enhancing the robustness and accelerating the convergence of our biconvex framework. While our approach explicitly encodes the absence of concurrent 2D observations, we aim to leverage the existence of nearly-incident corresponding viewing rays as a cue for the depth initialization of a given 3D point $\mathbf{X}_{(p,f)}$. To this end, we identify for each bundle of viewing rays captured in I_f (*i.e.* associated with a given shape structure $\mathbf{S}_f$) an alternative structure instance captured at I_j that minimizes the Euclidean 3D triangulation error across all corresponding viewing rays. In order to avoid a trivial solution arising from the small-baseline typically associated with consecutive frames of a single video, we restrict our search to ray bundles captured from distinct video sequences.

The position of each point $\mathbf{X}_{(p,f)}$ in $\mathbf{S}_f$ is determined by $d_{(p,f)}$ as in Equation (5.2). Denoting $\mathbf{d}_f = [d_{(1,f)}, \ldots, d_{(P,f)}]$, we can find the distance between shapes of $\mathbf{S}_f$ and $\mathbf{S}_j$ by minimizing the

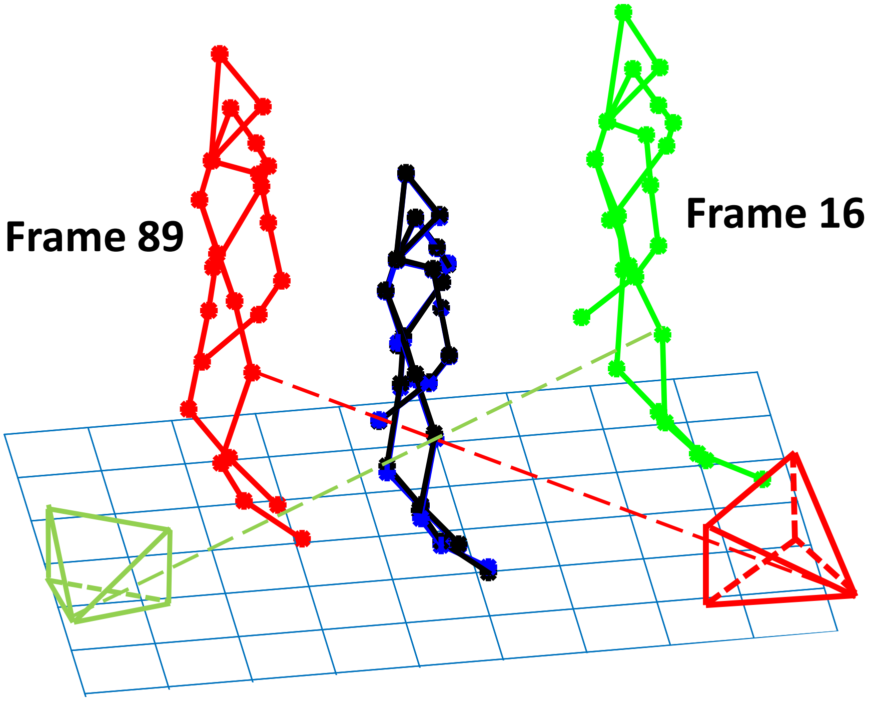


Figure 5.5: Example of incorrect initialization. The dataset ‘hopBothLegs3hops’ (Müller et al., 2007) has the motion of hopping forward three times. The black and blue shapes (almost overlapped) are the incorrect initialization of the real shapes (shown in green and red) of frames 16 and 89 due to the accidental ray intersections. This typically happens in the case of periodic motion such as walking or jogging. In the figure, only one set of nearly intersecting rays is plotted.

following cost function over the unknown variables $\mathbf{d}_f$ and $\mathbf{d}_j$

$$\{\mathbf{d}_f^*, \mathbf{d}_j^*\} = \underset{\mathbf{d}_f, \mathbf{d}_j}{\text{argmin}} ||\mathbf{S}_f - \mathbf{S}_j||_2^2. \tag{5.24}$$

This is a quadratic cost function with a closed-form solution.

We then build a symmetric distance matrix $\mathbf{D}$ with element $D_{(f,j)}$ equal to the minimum cost of Equation (5.24). If the frames f and j are from the same video, $D_{(f,j)}$ is set to infinity (or a very large number). Next, we identify many pseudo-intersection points with negative depth (*i.e.* , divergent pairs of viewing rays), and set the corresponding element in $\mathbf{D}$ to infinity. Finally, we determine the minimum element of each f-th row in our distance matrix $\mathbf{D}$ and assign the corresponding depth values $\mathbf{d}_f^*$ as our initialization for the definition of our 3D structure $\mathbf{S}_f$.

The above initialization is done regardless of available measurements, since we only look for an approximate initialization for the solver. In the case of missing data, the corresponding 3D points in the shape are simply ignored when minimizing Equation (5.24).

The output of the initialization is typically close to the ground truth, but may fail occasionally, as is shown in Figure 5.5. This kind of incorrect initialization may lead to poor estimation of the two shapes if the smoothness term $\Phi_2(\mathbb{X})$ in Equation (5.14) is not present, because these two shapes can well represent each other. Our cost term $\Phi_2(\mathbb{X})$ helps to pull the occasional incorrect shapes out of local minima.

5.7 Analysis and Discussion

This section provides key insight to our algorithm for dynamic object reconstruction without sequencing. The following statements will be illustrated in detail.

1. Interleaved 2D measures across video streams yields favorable viewing ray geometry for 3D shape estimation.

2. High-frequency 2D observations and smooth object motion jointly validate our self-expressive structure prior for accurate shape estimation.

3. No dependence on the availablity of sequencing information as opposed to existing approaches (Park et al., 2010; Valmadre and Lucey, 2012).

Next, we first describe the formulation of reconstruction errors by our method, based on which the above statements are illustrated at length in the subsequent three subsections.

5.7.1 Representation of Reconstruction Errors

Our solution computes 3D structure by minimizing the non-convex function Equation (5.14). Since direct analysis of the non-convex function is difficult, we only analyze the problem with the assumption that the ground truth of $\mathbb{W}$, which is defined as the output of Equation (5.14) given ground truth structure, is already known. Without loss of generality, we also assume the 2D observations are noise-free.

Given that in our method λ_2 is set to 0 in the end, and $\mathbb{W}$ is known and fixed, Equation (5.14) is equivalent to

$$\underset{\mathbb{X}}{\text{minimize}} \quad ||\mathbb{X} - \mathbb{X}\mathbb{W}||_{\mathrm{F}}^2. \tag{5.25}$$

From Equation (5.25), it can be seen when $\mathbb{W}$ is fixed, all points in a shape are computed independently, and computing one 3D point per shape versus multiple points per shape basically follows the same routine. Therefore, for the sake of more concise presentation, the analysis in this section assumes only one point per shape, and the point index p for the shape is omitted.

To analyze the reconstruction error, we assume that the ground truth of the 3D points is already known, and then analyze how much the computed structure deviates away from the ground truth, which is deemed as reconstruction error. We denote the ground truth 3D point as $\mathbb{X}^* = [\mathbf{X}_1^*, \cdots, \mathbf{X}_f^*, \cdots, \mathbf{X}_F^*]$. Then, any point $\mathbf{X}_f$ on the viewing ray that passes through $\mathbf{X}_f^*$ can be parameterized as

$$\mathbf{X}_f = \mathbf{X}_f^* + l_f \mathbf{r}_f, \tag{5.26}$$

where the unknown l_f is the signed distance from the ground truth along the viewing ray.

When minimizing Equation (5.25), using either Equation (5.26) or Equation (5.2) to represent $\mathbf{X}_f$ in practice generates different values of d_f and l_f, but the estimated 3D points are actually identical. Therefore, $|l_f|$ represents the Euclidean error of our method.

Equation (5.25) is a quadratic objective function without any constraint and has a closed-form solution. We use Equation (5.26) to represent the 3D point, and by setting the derivative of Eq. (5.25) over variables $\mathbf{l} = [l_1, \ldots, l_f, \cdots, l_F]$ to 0, we obtain a linear equation system denoted as

$$\mathbf{A}\mathbf{l} = \mathbf{b}, \tag{5.27}$$

where $\mathbf{A}$ is an $F \times F$ matrix with the f-th row given by

$$\mathbf{A}_{:f} = (\mathbf{I} - \mathbb{W})_{:f}(\mathbf{I} - \mathbb{W})^{\mathrm{T}}\mathrm{diag}([\mathbf{r}_1^T\mathbf{r}_f, \cdots, \mathbf{r}_F^T\mathbf{r}_f]), \tag{5.28}$$

and $\mathbf{b}$ is an $F \times 1$ vector with the f-th element given by

$$\mathbf{b}_f = \mathbf{r}_f^T\mathbb{X}^*(\mathbf{I} - \mathbb{W})(I - \mathbb{W})_{:f}^{\mathrm{T}}. \tag{5.29}$$

In Equations (5.28) and (5.29), the subscript $_{:f}$ denotes the f-th row of a matrix, and $\mathbf{I}$ is an identity matrix. Then the solution for $\mathbf{l}$ is

$$\mathbf{l} = \mathbf{A}^{-1}\mathbf{b}. \tag{5.30}$$

As mentioned, $\mathbf{l}$ is the reconstruction error, which is bounded by

$$||\mathbf{l}||_2 = ||\mathbf{A}^{-1}\mathbf{b}||_2 \leq ||\mathbf{A}^{-1}||_2||\mathbf{b}||_2. \tag{5.31}$$

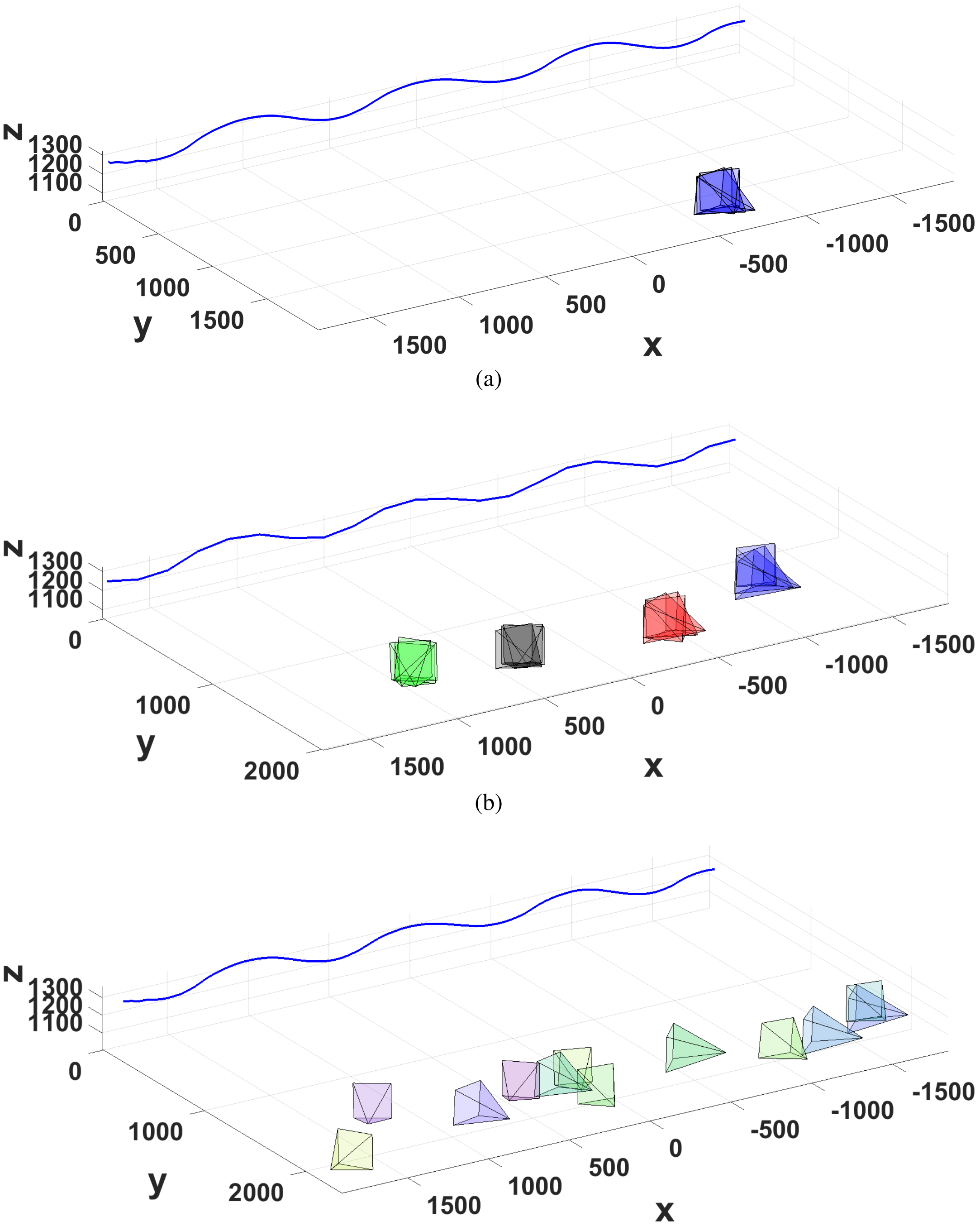
In this work, we use the term reconstructability (first defined in (Park et al., 2010)) as a criterion to characterize the reconstruction accuracy of our algorithm. In our case, in order to achieve high reconstructability, $||\mathbf{A}^{-1}||_2$ and $||\mathbf{b}||_2$ should be small. Next, we discuss $||\mathbf{A}^{-1}||_2$ and $||\mathbf{b}||_2$ in detail.

5.7.2 System Condition

Based on the definition of the matrix Euclidean norm, we have

$$||\mathbf{A}^{-1}||_2 = 1/\sigma_{\text{min}}, \tag{5.32}$$

where σ_{min} is the smallest singular value of matrix $\mathbf{A}$. With fixed $\mathbb{W}$, we observe from Equation (5.28) that $\mathbf{A}$ solely relies on the viewing ray directions and does not depend on the exact positions of the 3D points $\mathbb{X}^*$ along the viewing rays. Since σ_{min} is closely related to reconstruction errors and is determined by the camera system setup, we call it system condition. Note the system condition



(a)

(b)

(c)

Figure 5.6: Simulated camera setups. The blue curve is a trajectory of a 3D point obtained from motion capture data. Figures 5.6a and 5.6b depict the camera setups of one and four slow-moving handheld cameras. Figure 5.6c depicts a scenario where each random camera only captures one image. Figure 5.6b and Figure 5.6c show the camera setups used in our method and (Zheng et al., 2014b), respectively. Coordinates are in millimeters (mm).

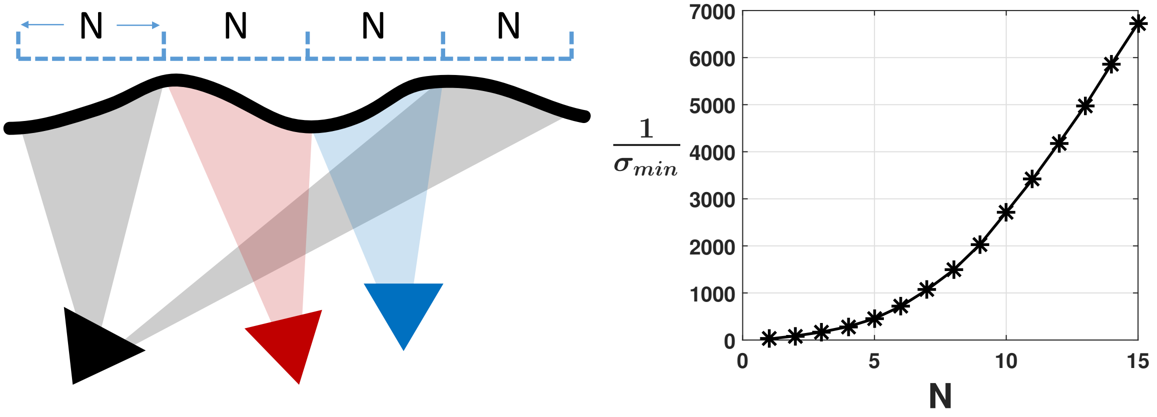
introduced here is in essence very similar to the system condition number described in the works (Valmadre and Lucey, 2012; Zheng et al., 2014b).

Since direct analysis of the system condition given viewing ray directions $\{\mathbf{r}_1, \ldots, \mathbf{r}_F\}$ based on Equation (5.28) is difficult, we next use empirical simulation to demonstrate the system condition under different camera setups.

In the experiments, we simulate scene captures close to real life. We use motion capture datasets that sample the 3D structure of real dynamic objects at 40 Hz. Figures 5.6a and 5.6b simulate setups of one handheld camera and multiple handheld cameras that record videos of a person walking. To mimic small random motion in each handheld camera, the camera centers at different time instances are Gaussian with standard deviation of 10 mm around a fixed center. We also test the case of completely random cameras (Figure 5.6c), with each taking one photo. The 3D structure at each time instance is projected to one of the virtual cameras to generate a set of 2D observations. For the scenario in Figure 5.6b, we ensure no two shapes at consecutive time instances are projected into the same video stream.

We estimate the system condition using Equation (5.32) on 500 trials with random cameras. The average system conditions for the cases of Figures 5.6a, 5.6b, and 5.6c are 1.48e+04, 22.3, and 29.0 respectively. It is evident the setup with one handheld camera has very low reconstructability. Note that even though the system conditions of the camera setups in Figures 5.6b and 5.6c are favorable, in practice the important sequencing information (see Section 5.7.4) across different cameras for these two cases is not readily available.

To illustrate the importance of cross-sequence 2D observations for our structure estimation process (statement 1), we evaluate system condition as a function of increased temporal gaps between cross-sequence samples. As shown in Figure 5.7a, the dynamic object is observed by one camera for N frames, and then observed by another camera for N frames. We show empirically that as N increases, the system condition increases monotonically (Figure 5.7b), which indicates more unstable reconstruction and typically larger errors (see experiments in Section 5.8.1.3), even under the assumption that $\mathbb{W}$ can be correctly estimated. This also illustrates that temporally consecutive



(a) Observation without overlap

(b) System condition

Figure 5.7: The reconstructability of the system is lower if the period of single-camera capture is longer.

shapes observed by the same video stream should not be used to represent each other, as is done in Section 5.4.2.

In fact, we observe that that reconstructability is closely related to the camera motion and the object motion. Specifically, if shape $\mathbf{S}_j$ is the most related shape to $\mathbf{S}_f$, as indicated by $\mathbb{W}$, the relative directions of viewing rays $\mathbf{r}_f$ and $\mathbf{r}_j$ (note we only have one point per shape in this analysis), determine the reconstructability. If the directions of $\mathbf{r}_f$ and $\mathbf{r}_j$ converge, *i.e.* the camera motion is relatively larger than the object motion, the reconstructability is higher. In the case of one handheld camera, the camera motion can be much smaller than the dynamic objects, and the viewing rays diverge, yielding low reconstructability. In contrast, if $\mathbf{r}_j$ and $\mathbf{r}_f$ are associated with different video cameras, the distance between the camera centers is much larger than the motion of the object. Hence the reconstructability is high. This observation is analogous to the classic triangulation of static scenes, where small baselines produce inaccurate reconstruction. Note the same conclusion was also made by Park et al. (2015), though their reconstruction algorithm is different from ours.

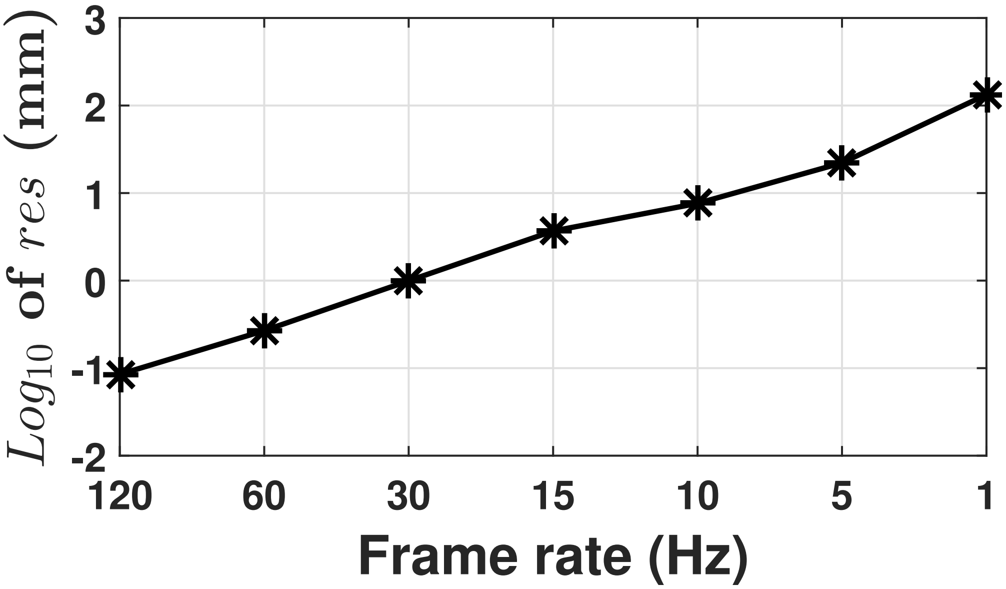


Figure 5.8: Average residuals res at different camera frame rates. Results are attained from 130 motion capture datasets in the work by Müller et al. (2007).

5.7.3 Shape Approximation Residual

While $\mathbf{A}$ depends on the viewing ray directions, which are available before reconstruction, $\mathbf{b}$ relies on the actual unknown positions of the ground truth structure $\mathbb{X}^*$ (Equation (5.29)). To achieve accurate reconstruction, each value in the vector $\mathbf{b}$ should be close to 0.

Since in Equation (5.29), $(I - \mathbb{W})^{\mathrm{T}}_{:f}$ is sparse, $\mathbf{b}_f$ can be considered as a linear combination of a few columns of matrix $\mathbb{X}^*(I - \mathbb{W})$ multiplied using dot product with the unit vector $\mathbf{r}_f$. Therefore, the value of $\mathbf{b}_f$ mainly relies on $||\mathbb{X}^*(I - \mathbb{W})||_{\mathrm{F}}$. Accordingly, we define the residual per point as

$$res = \frac{1}{PF}||\mathbb{X}^*(I - \mathbb{W})||_{\mathrm{F}}. \tag{5.33}$$

The residual res is small if all the shapes can be well represented by other shapes. It relies on speed of object motion and the capturing frame rate. We test the residual res given motion capture data sampled at different frame rates. Figure 5.8 shows res becomes larger as the frame rate goes down. This fits the intuition that shapes that are tempo-spatially farther away are less correlated. This also implies that our method cannot achieve accurate reconstruction from discrete images with large temporal discrepancy.

5.7.4 Importance of Image Sequencing

The temporal order of images, *i.e.*, image sequencing, plays an important role in dynamic object reconstruction (Park et al., 2010; Valmadre and Lucey, 2012). The work by (Valmadre and Lucey, 2012) generalizes the method by (Park et al., 2010) in a new framework based on high-pass filters. Here, we briefly describe the method by Valmadre and Lucey (2012) and its relation to our method, from which it can be revealed why their methods (Park et al., 2010; Valmadre and Lucey, 2012) require sequencing information as opposed to ours.

Assuming the object moves smoothly in the space, Valmadre and Lucey (2012) triangulate the 3D trajectory of an 3D point by minimizing its response to a set of high-pass filters. Given a predefined high pass filter $g = [g_M, \ldots, g_1]$, the trajectory is estimated by

$$\underset{\mathbb{X}}{\text{minimize}} \; ||\mathbb{X}G||_{\mathrm{F}}^2, \tag{5.34}$$

where G is defined as

$$G = \begin{bmatrix} g_M & & \\ \vdots & \ddots & \\ g_1 & \ddots & g_M \\ & \ddots & \ddots \\ & & g_1 \end{bmatrix}. \tag{5.35}$$

Each column of G is a high-pass filter for the local region of a trajectory. From the formulation, it is required for all the shapes (columns of $\mathbb{X}$) to be ordered temporally.

Comparing Equation (5.34) with Equation (5.25), we can see the two equations are the same if $\mathbf{G}$ equals $\mathbf{I} - \mathbb{W}$. In effect, the method by (Valmadre and Lucey, 2012) can be regarded as our method with a predefined $\mathbb{W}$. For instance, if the high pass filter is set to $\mathbf{g} = [1, -1]$, it is equivalent (ignoring the difference at boundary) that $\mathbb{W}$ is set to

$$\mathbb{W} = \begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ & 1 & \ddots & \\ & & \ddots & \end{bmatrix}. \tag{5.36}$$

Therefore, an alternative interpretation of their method (Valmadre and Lucey, 2012) using the high-pass filter $g = [1, -1]$ in terms of our theory is approximating the current shape using only the temporally closest shape.

Another high-pass filter proposed by Valmadre and Lucey (2012) is $[-1, 2, -1]$, which in our case is equivalent to fixing the weights of two neighboring shapes to 0.5. In effect, their method can be deemed as our method with predefined $\mathbb{W}$.

The importance of sequencing can also be revealed from analysis of residual defined by Equation (5.33). For the method by (Valmadre and Lucey, 2012) with predefined $\mathbf{G}$, the residual will be large if columns of $\mathbb{X}^*$ are randomly shuffled. In contrast, our method leverages compressive sensing to estimates $\mathbb{W}$ (instead of predefined), which automatically picks the most related shapes to produce small residuals.

5.8 Experiments

In our experiments, we evaluate our algorithm on both synthetic and real datasets. λ_1 and λ_2 in Equation (5.14) are set empirically to 0.05 and 0.1 for all the experiments. To alleviate the influence of different camera system scales (*i.e.* differing the scale of $\mathbb{X}$), the average distance between camera centers is normalized to 1 before applying our method. The soft constraint parameterization is used only in the presence of noisy measurements.

5.8.1 Simulation

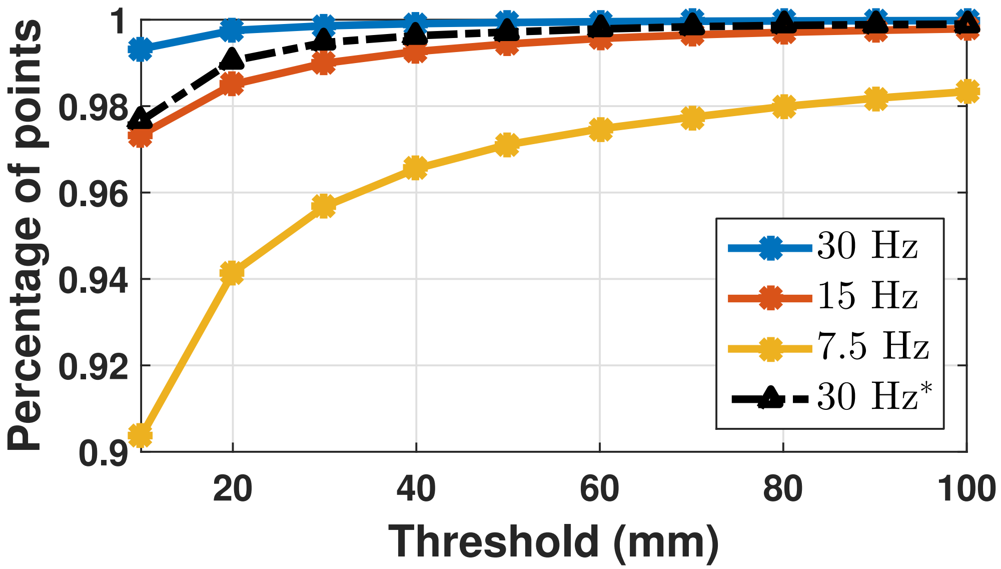
We use synthetic datasets to evaluate the accuracy and robustness of our proposal, and also compare against two state-of-the-art methods (Valmadre and Lucey, 2012; Dai et al., 2014). To generate synthetic data, we use the real motion capture datasets in the work by Müller et al. (2007), and leverage them as ground truth structure for our estimation. The whole datasets contain 130 different real motions including hopping, jogging, cartwheel, punching, *etc*. Each motion capture dataset is comprised of the temporal sequences of a common set of 44 3D points in real scale, which corresponds within our framework to ground truth structure $\mathbb{X}_{GT}$. The frame rate of the motion datasets, *i.e.* the sampling rate of the real continuous motion, is 120 Hz. The length of each dataset ranges from 45 to 701 frames, and with an average of 273 frames.

These 3D points are projected onto virtual cameras to generate input 2D measures into our methods. We select 4 virtual cameras with a resolution of 1M and focal length of 1000, and we position the static cameras around the centroid defined by $\mathbb{X}_{GT}$. The distance of the camera to the centroid is approximately twice the scale of $\mathbb{X}_{GT}$, and on average the distance is 2.7 meters. Considering the frame rate of the motion capture datasets is 120 Hz and there are 4 virtual cameras, the average frame rate for each camera is 30 Hz. Every temporal 3D capture is randomly assigned to each camera to build 4 disjoint image sequences. Unless otherwise mentioned, we enforce that no temporally consecutive captures are assigned to the same image sequence.

To evaluate our method, we compute the Euclidean errors between the ground truth and the estimated 3D points. We define the accuracy by counting the percentage of points having errors less than thresholds of 10, 20, 30, 40, 50, and 100 mm.

5.8.1.1 Accuracy

Different frame rates. We first evaluate how the algorithm behaves under different capture frame rates. 2D measures without noise are used to evaluate the accuracy of our method. In addition to the original motion capture data at 120 Hz, we also downsample the data to 60 and 30 Hz, so that each camera has frame rate of 15 and 7.5 Hz on average. As shown in Figure 5.9, the accuracy becomes worse as the frame rate gets slower. The main reason is that the self-representation residual is larger at lower frame rate. We notice that at a frame rate of 7.5 Hz, our method does not work well on the quick motions with large and nonlinear shape deformation, such as hopping or arms



Threshold Frame rate	10	20	30	40	50	100
30	0.9933	0.9975	0.9986	0.9991	0.9994	0.9998
15	0.9734	0.9850	0.9899	0.9926	0.9944	0.9979
7.5	0.9036	0.9415	0.9568	0.9655	0.9711	0.9833
30* (unconstrained assignment)	0.9766	0.9905	0.9947	0.9963	0.9971	0.9990

Figure 5.9: The reconstruction accuracy given different camera frame rates. We also test the case that the captures of object motion are randomly assigned to any of the image sequences without any constraint. 30 Hz* in the figure represents the unconstrained assignment.

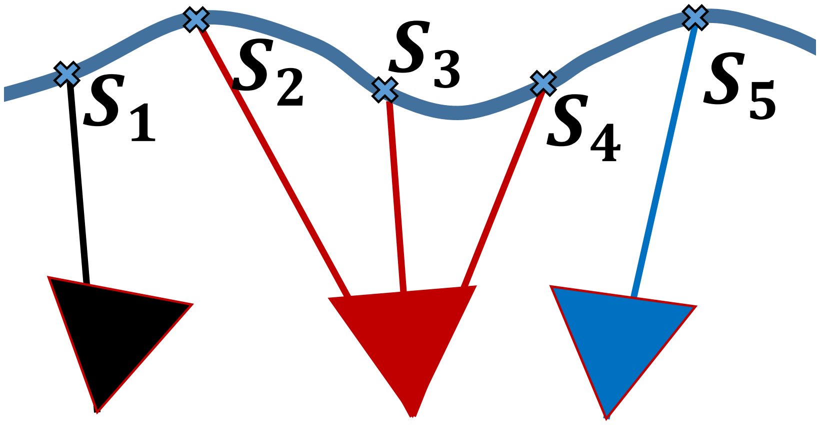


Figure 5.10: Consecutive captures are assigned to the same red camera. For easy visualizations, only one point per shape is drawn.

rotation. However, still more than 97% of 3D points have errors less than 5 cm, which is already very small considering the scale of a person and the distance range of the cameras.

Local temporal information. We also quantitatively evaluate the estimated $\mathbb{W}$. Using the same two measures described in Section 5.3, we get values of 0.9902 and 0.9923, compared to 0.9972 and 0.9994 if the 3D points are given. Therefore, our method very accurately recovers the local temporal information.

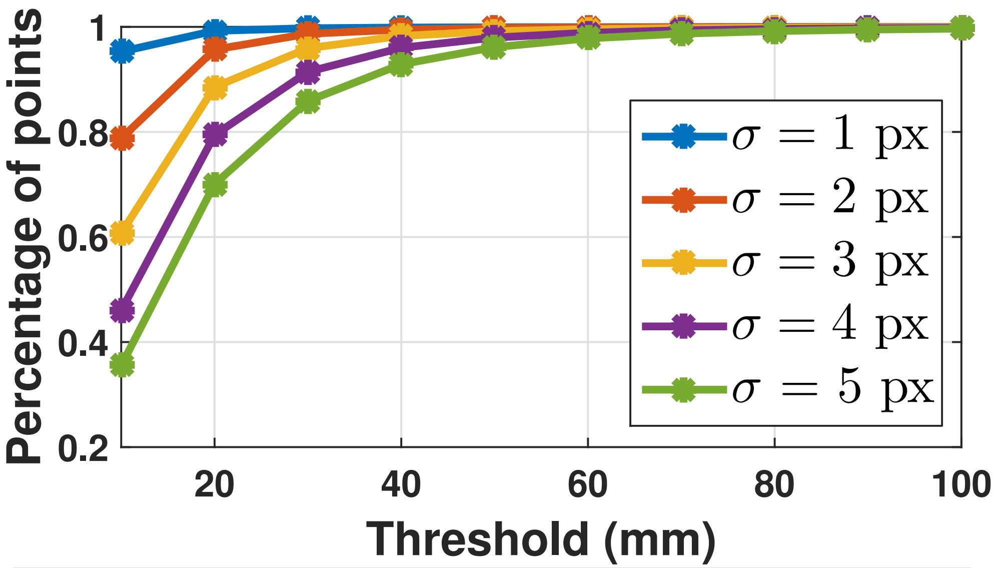
Unconstrained capture assignment. We test the case that each capture is randomly assigned to one of the four cameras so that temporally consecutive captures could have a chance to be assigned to the same camera, as is shown in Figure 5.10. In this specific case, shapes $\mathbf{S}_1$ and $\mathbf{S}_5$ are used to represent $\mathbf{S}_2$, $\mathbf{S}_3$ and $\mathbf{S}_4$. Based on the theory in Section 5.7.3, using spatially further away shapes to represent the current shape has larger residual and hence larger reconstruction errors, as is validated in Figure 5.9.

5.8.1.2 Data Robustness

To evaluate the robustness of our method, we test it in the case of noisy measurements and missing data.

Noisy measurements. We add zero-mean Gaussian noise with different standard deviations to the 2D measurements. Considering that the focal length of the image is 1000, one pixel error corresponds to one millimeter if the object is one meter away. We apply the soft constraint formulation described in Section 5.5.1 and empirically set the parameter λ_3 to 100. As depicted in Figure 5.11, the quality of reconstruction degrades as the noise level increases. As λ_3 increases, the soft constraint approximates the hard constraint. We evaluate the difference of the estimated results by the hard constraint formulation and the soft constraint formulation with different λ_3, and we show the median difference in Figure 5.12. It is apparent that as λ_3 increases, the difference of the output between the two formulations becomes smaller.

We have tested the hard constraint formulation using noisy measurements, and the overall accuracy of the output is very similar. Though the soft constraint appears more robust in the presence



Threshold / Noise	10	20	30	40	50	100
$\mathcal{N}(0,1)$	0.9529	0.9925	0.9974	0.9987	0.9992	0.9998
$\mathcal{N}(0,2)$	0.7878	0.9568	0.9869	0.9949	0.9976	0.9997
$\mathcal{N}(0,3)$	0.6074	0.8855	0.9593	0.9828	0.9917	0.9991
$\mathcal{N}(0,4)$	0.4601	0.7941	0.9144	0.9602	0.9797	0.9980
$\mathcal{N}(0,5)$	0.3551	0.7008	0.8590	0.9287	0.9615	0.9966

Figure 5.11: The reconstruction accuracy when the 2D observations are corrupted with Gaussian noise of different standard deviation (σ).

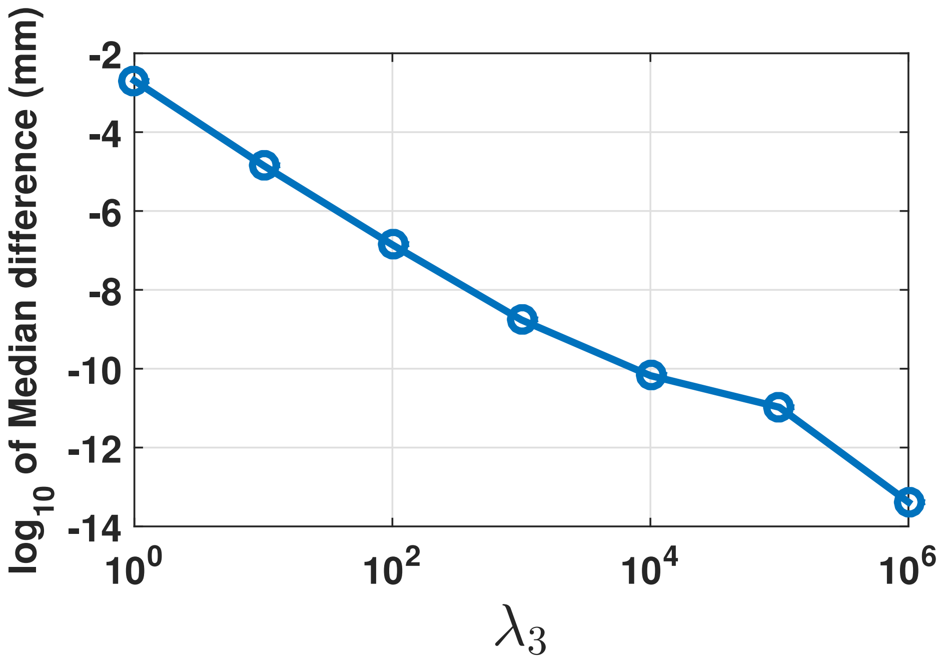
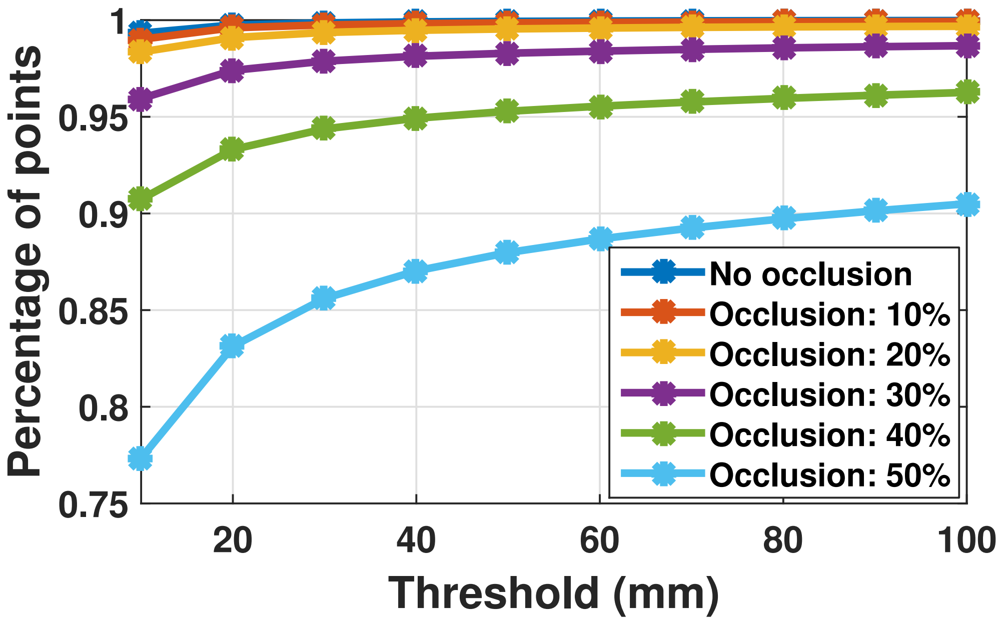


Figure 5.12: The difference of the estimated results by the hard constraint formulation in Equation (5.7) and the soft constraint formulation in Equation (5.17) with different λ_3



Threshold Miss rate	10	20	30	40	50	100
0%	0.9933	0.9975	0.9986	0.9991	0.9994	0.9998
10%	0.9901	0.9961	0.9975	0.9982	0.9986	0.9993
20%	0.9835	0.9910	0.9936	0.9948	0.9955	0.9968
30%	0.9594	0.9740	0.9788	0.9813	0.9829	0.9868
40%	0.9074	0.9331	0.9438	0.9493	0.9529	0.9626
50%	0.7734	0.8313	0.8560	0.8703	0.8798	0.9050

Figure 5.13: The reconstruction accuracy under different percentages of occluded points.

of noise as it allows the points off the viewing ray, there is no guarantee or proof this constraint will achieve more accurate results, as it depends on the exact motion of the objects.

Missing data. In our evaluation, we randomly set some 2D measures to be unavailable. Figure 5.13 depicts the accuracy under different percentages of missing data. We observe that under 20% of occlusion, there is not much difference in reconstruction accuracy. Moreover, under a large amount of 40% occlusion, our method still produces accurate results, with 94.38% of points having errors less than 30 mm.

Our method essentially linearly interpolates the 3D points along the trajectory using estimated $\mathbb{W}$. It can still produce 3D estimates in the presence of consecutive missing observations across time, but the accuracy in such scenarios depends on the object motion. Particularly, given large displacement of nonlinear motion, our method is likely to produce less accurate results.

5.8.1.3 Comparison to Other Methods

We compare our method with a non-rigid structure from motion method (Dai et al., 2014) and a trajectory triangulation method (Valmadre and Lucey, 2012). Both of these methods are state-of-the-art for dynamic object reconstruction.

NRSFM method. Non-rigid structure from motion (NRSFM) recovers both the camera motion and the dynamic structure. It is tempting to use those methods to solve our problem, since our problem with known camera poses seems to be easier. However, most NRSFM methods work on an orthographic or weak perspective camera model, and it is unclear of their applicability under the perspective model. Park et al. (2010) test the NRSFM methods by Akhter et al. (2009b); Torresani et al. (2008); Paladini et al. (2009) under a perspective camera model, but all of them fail to produce reasonably good results. In this work, we test the state-of-the-art NRSFM method by Dai et al. (2014).

The method by Dai et al. (2014) is based on the assumption that each non-rigid shape $\mathbf{X}_f$ is a linear combination of K shape bases, and hence the shape matrix (corresponding to $\mathbb{X}$ in our problem description) has low rank. After estimating the camera motion, they recover the structure by minimizing the rank of the shape matrix, which is achieved through the minimization of the matrix nuclear norm. Their method applies to an orthographic camera model, but can be easily adapted to a perspective model, as described below.

Given the camera poses, we use the block matrix method proposed in the work by Dai et al. (2014). Denoting

$$\mathbb{X}^{\#} = \begin{bmatrix} \mathtt{X}_{(1,1)} & \dots & \mathtt{X}_{(P,1)} & \mathtt{Y}_{(1,1)} & \dots & \mathtt{Y}_{(P,F)} & \mathtt{Z}_{(1,1)} & \dots & \mathtt{Z}_{(P,F)} \\ \vdots & & \vdots & \vdots & \vdots & \vdots & & & \\ \mathtt{X}_{(1,F)} & \dots & \mathtt{X}_{(P,F)} & \mathtt{Y}_{(1,F)} & \dots & \mathtt{Y}_{(P,F)} & \mathtt{Z}_{(1,1)} & \dots & \mathtt{Z}_{(P,F)} \end{bmatrix},$$

where $\mathbf{X}_{(p,f)} = (\mathtt{X}_{(p,f)}, \mathtt{Y}_{(p,f)}, \mathtt{Z}_{(p,f)})^{\mathrm{T}}$, the shape of the object can be recovered through

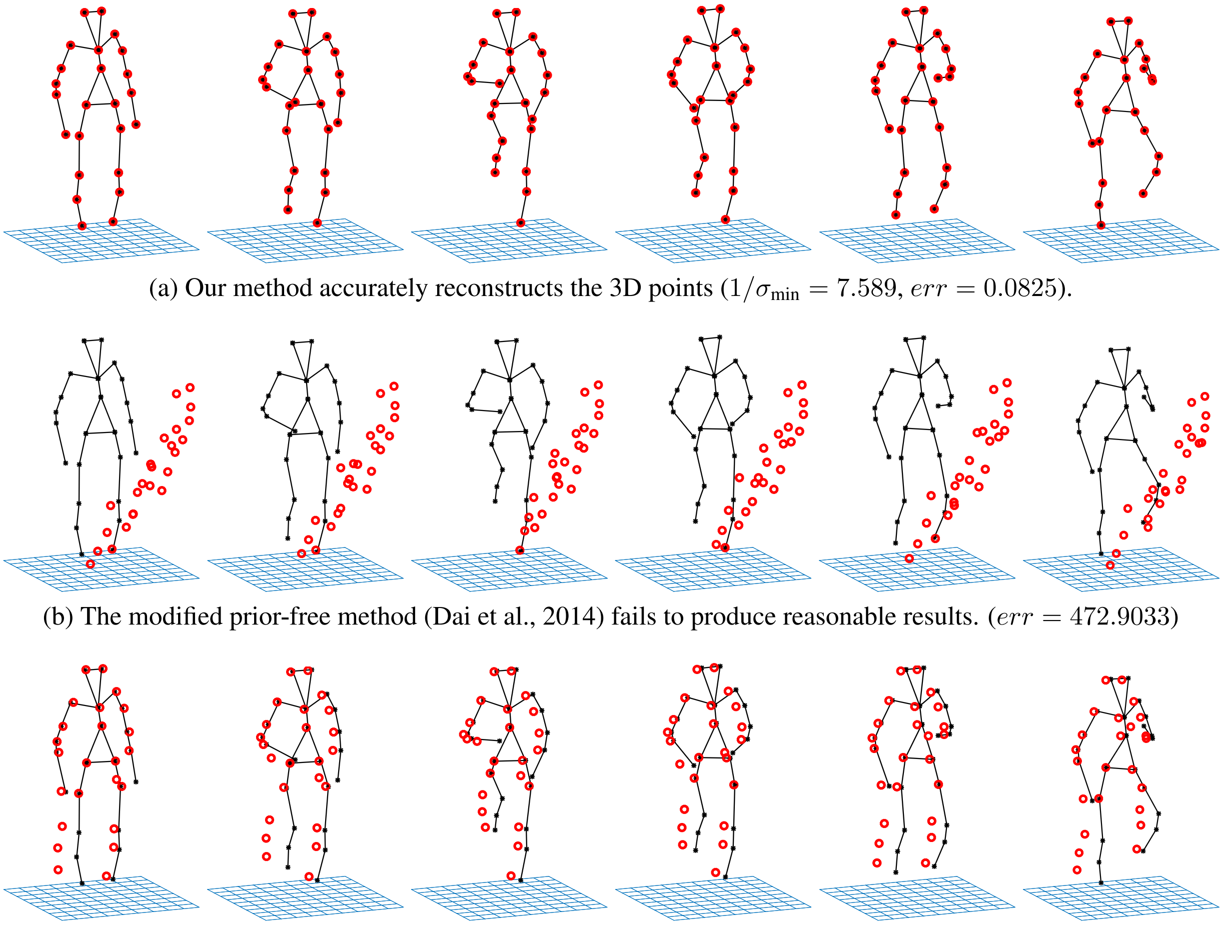
$$\begin{aligned} \underset{\mathbb{X}^{\#}, \mathbb{W}}{\text{minimize}} \quad & ||\mathbb{X}^{\#}||_{*} + \mu||\mathbf{1}_{P\text{x}1} \otimes \mathbb{C} + (\mathbb{d} \otimes \mathbf{1}_{3\text{x}1}) \odot \mathbb{r} - \mathbb{X}||_{\mathrm{F}} \\ \text{subject to} \quad & \mathbb{X}^{\#} = \mathcal{L}(\mathbb{X}), \end{aligned}$$

where $|| \cdot ||_{*}$ is the matrix nuclear norm, μ is a positive weight, and $\mathcal{L}$ is a linear operator that reshapes $\mathbb{X}$ into $\mathbb{X}^{\#}$.

This formulation seems attractive at first glance due to its convexity, in contrast to our non-convex formulation. Moreover, their method is shape-based (instead of trajectory-based), and does not require temporal information. To test the NRSFM method, we use synthetic data without noise and the random camera configuration shown in Figure 5.6c. Unfortunately, the qualitative results in Figure 5.14b show that it completely fails, as opposed to our method shown in Figure 5.14a.

Trajectory Triangulation Method. We also compare with the trajectory triangulation method by Valmadre and Lucey (2012), as is described in Section 5.7.4. Since the required sequencing information is readily available within each video stream, our test uses the simulation of one handheld camera as shown in Figure 5.6a. The camera centers are Gaussian with 20 mm standard deviation (σ_c) around a fixed point. Based on the theory in Section 5.7.2, the reconstructability increases with larger σ_c. Considering that the framerate of the motion capture dataset is 120 Hz, the camera motion with $\sigma_c = 20$mm is already very large compared to real handheld captures.

The method triangulates the trajectory of each dynamic point independently, and each trajectory has one system condition given the viewing ray directions. Since the motion of the person's head is relatively slower than that of his legs, the corresponding system condition is lower and the reconstructed points are more accurate, based on the theory in Section 5.7.2. The average system condition for all the points is 2228. Figure 5.14c shows the large system condition in this camera setup leads to significant reconstruction errors.

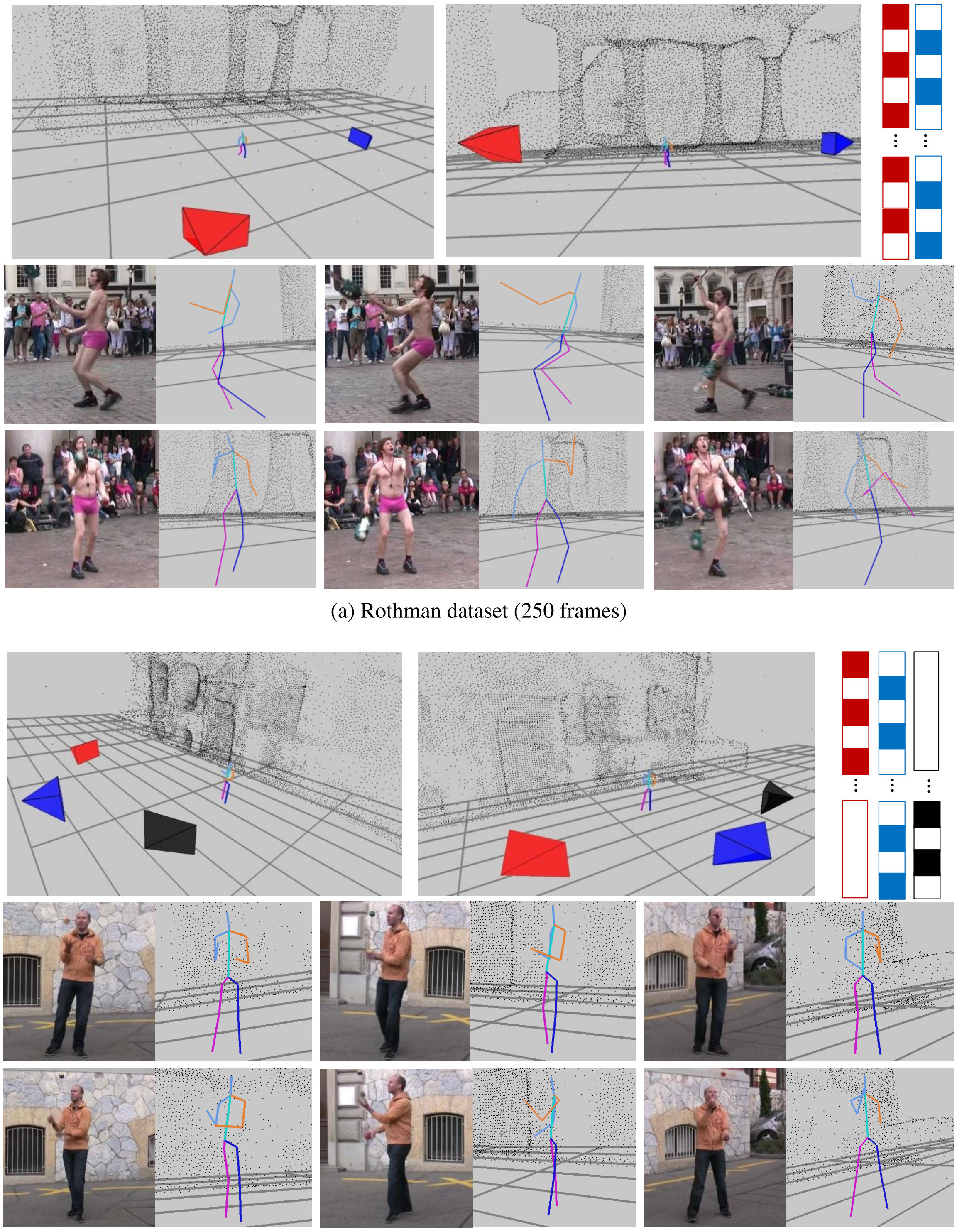


(a) Our method accurately reconstructs the 3D points ($1/\sigma_{\min} = 7.589$, $err = 0.0825$).

(b) The modified prior-free method (Dai et al., 2014) fails to produce reasonable results. ($err = 472.9033$)

(c) General trajectory prior method (Valmadre and Lucey, 2012) produces large errors due to high system condition ($1/\sigma_{\min} = 2228$, $err = 76.9700$).

Figure 5.14: Qualitative comparison of our method with (Dai et al., 2014) and (Valmadre and Lucey, 2012) on the motion capture dataset ‘jog on place’ in (Müller et al., 2007). The dataset has 214 frames, with 44 points per frame (only 24 are shown for visualization purposes). The black and red points are the ground truth and the estimated results, respectively. err is the average Euclidean error per point.



(a) Rothman dataset (250 frames)

(b) Juggler dataset (180 frames)

Figure 5.15: The datasets presented in (Ballan et al., 2010). The frame rate of each camera is 12.5 Hz. For each dataset, the top left two show the camera configuration, the top right describes the temporal distribution of each image sequence (a colored grid means the camera of the same color captures one frame at a time instance), and the bottom shows example reconstruction results.

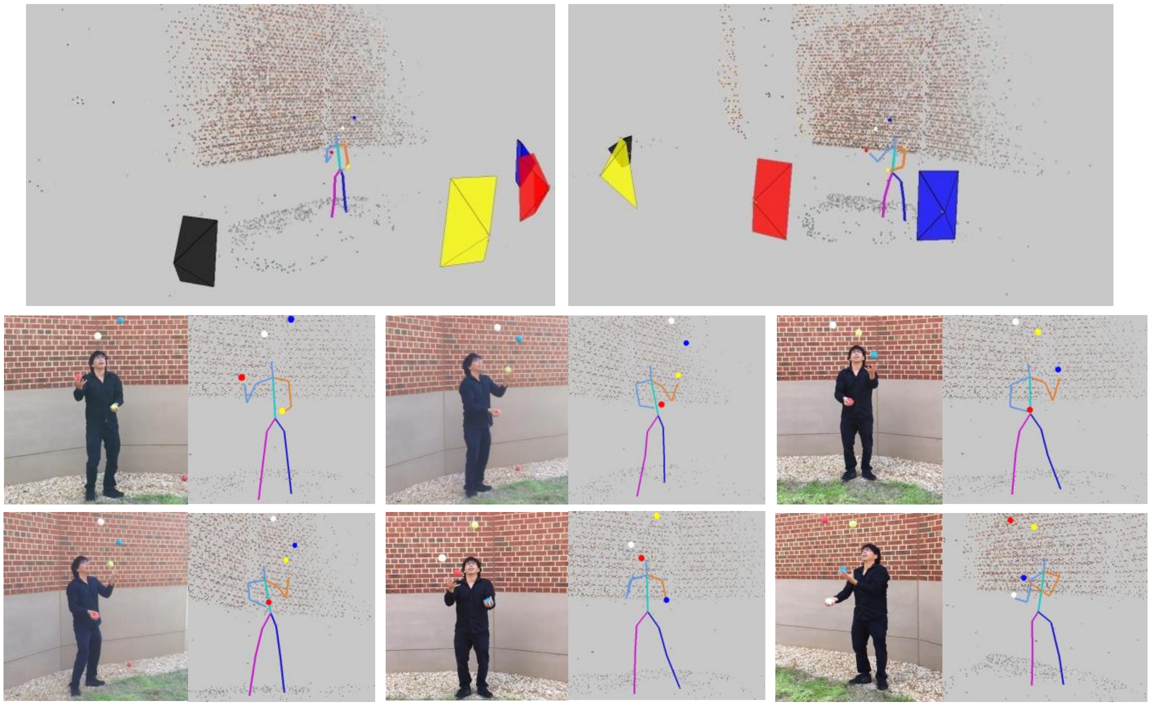


Figure 5.16: Results of a person juggling. Note we reconstruct the four juggler balls in addition to the person. The image sequence from iPhone6 and iPhone5 have frame rates of 10 Hz and 6.25 Hz respectively

5.8.2 Real Datasets

For experiments on real image capture, we use the Juggler and Rothman datasets from (Ballan et al., 2010). Given that the original datasets were synchronized, we sample the video frames to avoid concurrent captures (see Figure 5.15). We do not use the datasets in the work by Basha et al. (2012); Park et al. (2010) because they only provide images with large temporal discrepancy, and therefore the shape residual is large (*i.e.* Equation (5.8) does not hold). We also capture a new dataset of a person juggling using three iPhone6 and one iPhone5 without temporal synchronization.

We perform manual feature labeling on the input sequences and provide the obtained set of 2D measurements as input for our estimation process. For visualization purposes, Figures 5.15 and 5.16 depict the estimated 3D geometry by connecting the estimated position of the detected joint elements through 3D line segments.

5.9 Conclusion and Contributions

We have presented a method for dynamic object reconstruction from unsynchronized video streams. We demonstrated the effectiveness of our proposed method on both real and synthetic datasets. This is a first step towards dynamic 3D modeling in the wild.

The main contributions of our approach encompass:

1. **Problem Definition**. We are the first to address the problem of dynamic 3D reconstruction using unsynchronized cross-video streams.

2. **Methodology Formulation**. We pose the problem in terms of a self-expressive dictionary learning framework leveraging a novel data-adaptive local 3D interpolation model.

3. **Implementation Mechanisms**. We define and solve a biconvex optimization problem and develop an efficient ADMM-based solver amenable for parallel implementation.

To the best of our knowledge, we are the first to use the self-expression prior to solve the problem of dynamic object reconstruction. This prior has the potential to be applied in the traditional NRSFM problems.

CHAPTER 6: DISCUSSION

This dissertation presents three works for the problems in static scene reconstruction and dynamic object reconstruction. In Chapter 3, we proposed a framework of joint view selection and depthmap estimation. The experiments on large Internet collected photos demonstrates its efficiency and robustness. In Chapter 4 and Chapter 5, we solved the problems of dynamic object reconstruction from unstructured images and unsyncthronized videos, respectively. In solving these two problems, our main effort focused on 3D reconstruction without the information of spatial/temporal proximity. We showed effectiveness of the approaches by testing on synthetic and real datasets. In this section, we discuss the possible extensions of our works, as well as the potential future research directions.

6.1 Future work

6.1.1 Extensions to PatchMatch-based Joint View Selection and Depthmap Estimation

Though our method in Chapter 3 significantly outperforms existing methods on Internet collected photos (Goesele et al., 2007) and achieves the state-of-the-art accuracy on standard datasets collected under a controlled lab environment (Strecha et al., 2008). The accuracy of the method can be further improved by incorporating some standard techniques into our framework. Next, we discuss each of the techniques in detail.

In our method, we use the fronto-parallel planes to warp color patches in the reference image onto other source images to perform a color consistency check. It has been shown the plane orientation affects the reconstruction accuracy (Gallup et al., 2007; Furukawa and Ponce, 2010). Ideally, the plane orientation should be the same as the real surface normal, which is unknown before reconstruction. To address this issue, we can include the surface normals as unknown variables in our framework. Specifically, the unknown normal directions are propagated to the neighboring pixels in addition to the depths (Bleyer et al., 2011). This scheme is able to further improve reconstruction quality on the regions having large angles with the camera viewing directions (*e.g*. the ground), but at the cost of increased computational complexity.

Another issue related to color patches arises if the pixels in a patch cover scenes of significantly different depths, which typically occurs at the boundary of object surfaces. In stereo, the correspondences among multiple images are found by checking the color consistency. To improve the robustness for the color consistency measure between two pixels, current local methods (*i.e.* methods having no smoothness term between neighboring pixels in the depthmap) typically compare the two patches around the pixels. The method present in Chapter 3 applies normalized cross correlation (NCC) as a metric to measure the color consistency, where each pixel in the patch contributes equally to the measure. However, this is likely to produce swollen/fat boundary effect in the depthmap, since the use of a plane for patch warping assumes all pixels in the patch lie on the same plane, and this assumption breaks at the boundary of object surfaces. Therefore, when comparing two patches, the pixels lying on the same estimated plane as the central pixel should be given higher weight than other pixels. To achieve this, one heuristic but empirically effective solution is to use adaptive weights for each pixel within the patch, with the weights both propotional to the color similarity and the spatial proximity relative to the patch's center on the reference image (Yoon and Kweon, 2006).

Another extension to our work is to handle cameras with small baselines. In stereo methods, small baselines usually lead to unstable and inaccurate results (Hartley and Zisserman, 2004). Since the large set of Internet collected photos is typically taken at certain spots of interest, it is very likely some of the images have very small or zero baselines. Our framework in Chapter 3 selects images based on color consistency, and the images with small baselines will generally be selected because the color consistency is always high, regardless of the depth hypothesis. To address this issue, the angle of two viewing rays given a depth hypothesis should be tested to prevent invalid triangulation (Gallup et al., 2008). We can incorporate the angle value in the likelihood function, which should convey the knowledge that if the angle of two viewing rays is very small, the corresponding source image and the depth hypothesis should be deemed unreliable. In this way, the final output depth for each pixel should have appropriate triangulation angles.

Another issue related to depth estimation comes from homogeneous color regions (*i.e.* image regions lacking a textured color pattern). All existing methods based on local color consistency checks fail on these regions. To handle this problem, I believe it is necessary to incorporate the semantic knowledge of the scene rather than to just rely on low-level features such as colors. This inevitably requires introducing machine learning techniques into the stereo problem. However, incorporating camera parameters into a machine learning framework is difficult, since the testing data and training data often have different camera parameters. Although there are many single-image depth estimation approaches based on supervised machine learning (Hoiem et al., 2005; Saxena et al., 2008; Eigen et al., 2014; Liu et al., 2014; Zhuo et al., 2015), still much work needs to be done to incorporate such techniques into multiview stereo methods for more accurate depth estimation.

6.1.2 Extensions to JOST

The method presented in Chapter 4 uses object detection output as features, and the object lies along the viewing ray passing the 2D features. However, the outlier detections may prevent the algorithm from finding the correct object class trajectories. One way to manage this problem, as is done in Chapter 4, is to raise the detection threshold to suppress the false alarm rate, at the cost of increasing misdetections. Another possible way is to embed our method in a RANSAC framework (Hartley and Zisserman, 2004) to remove outliers. Specifically, a subset of randomly sampled detections is used to triangulate the trajectory, and count the number of remaining detections censuses with the trajectory as inliers. Repeating this process to yield the trajectory with the largest number of inliers. However, this scheme is computational intensive if the ratio of outliers is large, since running trajectory triangulation given a subset of detections is time-consuming.

Efficiency is another issue for our approach. In our method, the nonconvex problem is solved in a discrete-continuous scheme, and the discrete step involves solving a NP-hard GMST problem. The efficiency of solving a GMST problem can be attained by reducing the complexity of the multipartite graph. In a multipartite graph, there exists an edge between every two nodes in different independent sets (partite sets). The computational complexity of finding GMST will be lowered down if the number of edges and nodes of the graph is reduced. To achieve this, a prior knowledge, if available, can be incorporated easily. For instance, if it is known that two specific detected objects are farther away in 3D space, then all the edges connecting the associated two sets of nodes can be safely removed, since these two objects are not neighboring in the object class trajectory. Moreover, if the scene model size and the real object size is available, then the size of the detection windows can be used to roughly estimate a tight depth range of the dynamic objects, which helps reduce the number of nodes in the partite sets and hence the number of edges in the graph.

6.1.3 Extensions to Dynamic Object Reconstruction from Unsynchronized Videos

In Chapter 5, we obtain the 2D correspondences across images manually as input for our approach. This step can be automated by optical flow (Brox et al., 2004) or graph match based matching algorithms (Yan et al., 2015a,b). Moreover, optical flow can produce dense correspondences so that we can reconstruct dense 3D points for the dynamic objects.

The method presented in Chapter 4 and Chapter 5 requires a static background scene present in the image so that structure from motion can use it for camera registration. However, the crowd sourced data may have dynamic objects as the main focus and lack the content of the background scenes. This comes an open question of how to register cameras given non-current captures of dynamic objects. Considering the importance and difficulity of this problem, it is an exciting future research direction.

Moreover, to the best of our knowledge, the work in Chapter 5 is the first self-representation framework for dynamic object reconstruction. That is, each temporally varied shape can be represented by a linear combination of a few other shapes at different time instances, given the smooth motion of dynamic objects. This self-representation constraint has potential to be used to solve the NRSFM problems. Compared to most of the existing works for NRSFM, where the assumption that any shape is a linear combination of K shape bases is applied, our self-representation constraint is more intuitive and can lead to better reconstruction results.

BIBLIOGRAPHY

Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S. M., and Szeliski, R. (2011). Building rome in a day. *Communications of the ACM*.

Agarwal, S., Snavely, N., Seitz, S. M., and Szeliski, R. (2010). Bundle Adjustment in the Large. In *European Conference on Computer Vision (ECCV)*.

Aharon, M., Elad, M., and Bruckstein, A. (2006). SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*.

Akhter, I., Sheikh, Y., and Khan, S. (2009a). In defense of orthonormality constraints for nonrigid structure from motion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Akhter, I., Sheikh, Y., Khan, S., and Kanade, T. (2009b). Nonrigid structure from motion in trajectory space. In *NIPS*.

Avidan, S. and Shashua, A. (2000). Trajectory triangulation: 3d reconstruction of moving points from a monocular image sequence. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*.

Bach, F., Jenatton, R., Mairal, J., and Obozinski, G. (2012). Optimization with sparsity-inducing penalties. *Foundations and Trends® in Machine Learning*.

Bailer, C., Finckh, M., and Lensch, H. P. A. (2012). Scale robust multi view stereo. In *European Conference on Computer Vision (ECCV)*.

Ballan, L., Brostow, G., Puwein, J., and Pollefeys, M. (2010). Unstructured video-based rendering: Interactive exploration of casually captured videos. In *ACM Transactions on Graphics (TOG)*.

Barnes, C., Shechtman, E., Finkelstein, A., and Goldman, D. B. (2009). PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (TOG), Proceedings of SIGGRAPH*.

Basha, T., Moses, Y., and Avidan, S. (2012). Photo sequencing. In *European Conference on Computer Vision (ECCV)*.

Basha, T., Moses, Y., and Avidan, S. (2013). Space-time tradeoffs in photo sequencing. In *IEEE International Conference on Computer Vision (ICCV)*.

Bay, H., Ess, A., Tuytelaars, T., and Gool, L. V. (2008). Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*.

Besse, F., Rother, C., and Kautz, J. (2012). Pmbp: Patchmatch belief propagation for correspondence field estimation. In *British Machine Vision Conference (BMVC)*.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc, NJ, USA.

Blanz, V. and Vetter, T. (2003). Face recognition based on fitting a 3d morphable model. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI).*

Bleyer, M., Rhemann, C., and Rother, C. (2011). Patchmatch stereo - stereo matching with slanted support windows. In *British Machine Vision Conference (BMVC).*

Bouguet, J.-Y. (2000). Matlab camera calibration toolbox. `http://www.vision.caltech.edu/bouguetj/calib_doc/`.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122.

Boykov, Y., Veksler, O., and Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI).*

Bregler, C., Hertzmann, A., and Biermann, H. (2000). Recovering non-rigid 3d shape from image streams.

Brox, T., Bruhn, A., Papenberg, N., and Weickert, J. (2004). High accuracy optical flow estimation based on a theory for warping. In *European Conference on Computer Vision (ECCV).*

Campbell, N. D. F., Vogiatzis, G., Esteban, C. H., and Cipolla, R. (2008). Using multiple hypotheses to improve depthmaps for multi-view stereo. In *European Conference on Computer Vision (ECCV).*

Carlo, T. and Takeo, K. (1992). Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision (IJCV).*

Chen, S. E. and Williams, L. (1993). View interpolation for image synthesis. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*.

Chen, Y., Mairal, J., and Harchaoui, Z. (2014). Fast and Robust Archetypal Analysis for Representation Learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Cormen, T. H., Stein, C., Rivest, R. L., and Leiserson, C. E. (2009). *Introduction to Algorithms*. 3nd edition.

Coughlan, J. M. and Yuille, A. L. (1999). Manhattan world: Compass direction from a single image by bayesian inference. In *IEEE International Conference on Computer Vision (ICCV).*

Dai, Y., Li, H., and He, M. (2014). A simple prior-free method for non-rigid structure-from-motion factorization. *International Journal of Computer Vision (IJCV)*, 107(2):101–122.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Delage, E., Lee, H., and Ng, A. Y. (2005). Automatic single-image 3d reconstructions of indoor manhattan world scenes. In *International Symposium on Robotics Research.*

Derek Hoiem, Alexei A. Efros, M. H. (2005). Geometric context from a single image. In *IEEE International Conference on Computer Vision (ICCV).*

Dror, M., Haouari, M., and Chaouachi, J. (2000). Generalized spanning trees. *European Journal of Operational Research.*

Eigen, D., Puhrsch, C., and Fergus, R. (2014). Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems (NIPS).*

Elad, M. and Aharon, M. (2006). Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing.*

Elhamifar, E. and Vidal, R. (2009). Sparse subspace clustering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Endres, F., Hess, J., Engelhard, N., Sturm, J., Cremers, D., and Burgard, W. (2012). An evaluation of the rgb-d slam system. In *IEEE International Conference on Robotics and Automation (ICRA).*

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. (2010). Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI).*

Feremans, C., Labbe, M., and Laporte, G. (2002). A comparative analysis of several formulations for the generalized minimum spanning tree problem. *Networks.*

Ferreira, C. S., Ochi, L. S., Parada, V., and Uchoa, E. (2012). A grasp-based approach to the generalized minimum spanning tree problem. *Expert Systems with Applications.*

Frahm, J., Georgel, P., Gallup, D., Johnson, T., Raguram, R., Wu, C., Jen, Y., Dunn, E., Clipp, B., Lazebnik, S., and Pollefeys, M. (2010). Building rome on a cloudless day. *European Conference on Computer Vision (ECCV).*

Furukawa, Y., Curless, B., Seitz, S. M., and Szeliski, R. (2010). Towards Internet-scale multi-view stereo. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Furukawa, Y. and Ponce, J. (2010). Accurate, dense, and robust multiview stereopsis. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI).*

Gallup, D., Frahm, J.-M., Mordohai, P., and Pollefeys, M. (2008). Variable baseline/resolution stereo. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Gallup, D., Frahm, J.-M., Mordohai, P., Yang, Q., and Pollefeys, M. (2007). Real-time plane-sweeping stereo with multiple sweeping directions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Gallup, D., Frahm, J.-M., and Pollefeys, M. (2010a). A heightmap model for efficient 3d reconstruction from street-level video. In *Proceedings of the international symposium on 3D data processing, visualization and transmission (3DPVT).*

Gallup, D., Pollefeys, M., and Frahm, J.-M. (2010b). 3d reconstruction using an n-layer heightmap. In *German Association for Pattern Recognition (DAGM)*.

Girshick, R. B., Felzenszwalb, P. F., and McAllester, D. (2012). Discriminatively trained deformable part models, release 5. `http://people.cs.uchicago.edu/~rbg/latent-release5/`.

Goesele, M., Snavely, N., Curless, B., Hoppe, H., and Seitz, S. M. (2007). Multi-view stereo for community photo collections. In *IEEE International Conference on Computer Vision (ICCV)*.

Grant, M. and Boyd, S. (2014). CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`.

Gu, L. and Kanade, T. (2006). 3d alignment of face in a single image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Gupta, S., Arbelaez, P., and Malik, J. (2013). Perceptual organization and recognition of indoor scenes from rgb-d images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hartley, R. and Vidal, R. (2008). Perspective nonrigid shape and motion recovery. In *European Conference on Computer Vision (ECCV)*.

Hartley, R. I. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition.

Heinly, J. (2015). *Toward Efficient and Robust Large-Scale Structure-from-Motion Systems*. PhD thesis, The University of North Carolina at Chapel Hill.

Heinly, J., Dunn, E., and Frahm, J.-M. (2014). Correcting for Duplicate Scene Structure in Sparse 3D Reconstruction. In *European Conference on Computer Vision (ECCV)*.

Heinly, J., Schönberger, J., Dunn, E., and Frahm, J.-M. (2015). Reconstructing the World* in Six Days *(As Captured by the Yahoo 100 Million Image Dataset). *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hoiem, D., Efros, A. A., and Hebert, M. (2005). Automatic photo pop-up. In *ACM Transactions on Graphics (TOG), Proceedings of SIGGRAPH*.

Hu, X. and Mordohai, P. (2012). Least commitment, viewpoint-based, multi-view stereo. In *Proceedings of the international symposium on 3D data processing, visualization and transmission (3DPVT)*.

Jain, V. and Learned-Miller, E. G. (2010). Fddb: a benchmark for face detection in unconstrained settings. *UMass Amherst Technical Report*.

Jancosek, M. and Pajdla, T. (2011). Robust, accurate and weaklysupported-surfaces preserving multi-view reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jones, M. and Viola, P. (2003). Fast multi-view face detection. *Mitsubishi Electric Research Lab TR-20003-96*.

Kang, S., Szeliski, R., and Chai, J. (2001). Handling occlusions in dense multi-view stereo. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kneip, L., Scaramuzza, D., and Siegwart, R. (2011). A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Liu, M., Salzmann, M., and He, X. (2014). Discrete-continuous depth estimation from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Lowe, D. G. (2004). Distinctive Image features from Scale-Invariant Keypoints. *International Journal of Computer Vision (IJCV)*, 60(2).

Lu, J., Yang, H., Min, D., and Do, M. N. (2013). Patch match filter: Efficient edge-aware filtering meets randomized search for fast correspondence field estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Müller, M., Röder, T., Clausen, M., Eberhardt, B., Krüger, B., and Weber, A. (2007). Documentation mocap database hdm05. Technical Report CG-2007-2, Universität Bonn.

Myung, Y., Lee, C., and Tcha, D. (1995). On the generalized minimum spanning tree problem. *Networks*.

Neal, R. and Hinton, G. E. (1998). A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in Graphical Models*.

Nistér, D. (2003). An Efficient Solution to the Five-Point Relative Pose Problem. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Oliva, A. and Torralba, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision (IJCV)*.

Oncan, T., Cordeau, J., and Gilbert, L. (2008). a tabu search heuristic for the generalized minimum spanning tree problem. *European Journal of Operational Research*.

Paladini, M., Del Bue, A., Stosic, M., Dodig, M., Xavier, J., and Agapito, L. (2009). Factorization for non-rigid and articulated structure using metric projections. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Park, H. and Sheikh, Y. (2011). 3d reconstruction of a smooth articulated trajectory from a monocular image sequence. In *IEEE International Conference on Computer Vision (ICCV)*.

Park, H. S., Shiratori, T., Matthews, I., and Sheikh, Y. (2010). 3d reconstruction of a moving point from a series of 2d projections. In *European Conference on Computer Vision (ECCV)*.

Park, H. S., Shiratori, T., Matthews, I., and Sheikh, Y. (2015). 3d trajectory reconstruction under perspective projection. *International Journal of Computer Vision (IJCV)*.

Raguram, R., Chum, O., Pollefeys, M., Matas, J., and Frahm, J. (2013). Usac: a universal framework for random sample consensus. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*.

Ramalingam, S. and Brand, M. (2013). Lifting 3d manhattan lines from a single image. In *IEEE International Conference on Computer Vision (ICCV)*.

Rao, C., Gritai, A., Shah, M., and Syeda-Mahmood, T. (2003). View-invariant alignment and matching of video sequences. In *IEEE International Conference on Computer Vision (ICCV)*.

Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). ORB: An Efficient Alternative to SIFT or SURF. In *IEEE International Conference on Computer Vision (ICCV)*.

Saxena, A., Chung, S. H., and Ng, A. Y. (2008). 3d depth reconstruction from a single still image. *International Journal of Computer Vision (IJCV)*.

Scharstein, D. and Pal, C. (2007). Learning conditional random fields for stereo. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Scharstein, D. and Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision (IJCV)*.

Schönberger, J. L., Berg, A. C., and Frahm, J.-M. (2015). Paige: Pairwise image geometry encoding for improved efficiency in structure-from-motion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Shen, S. (2013). Accurate multiple view 3d reconstruction using patch-based stereo for large-scale scenes. In *IEEE Transactions on Image Processing (TIP)*.

Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A. (2011). Real-Time Human Pose Recognition in Parts from Single Depth Images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Shrestha, P., Barbieri, M., Weda, H., and Sekulovski, D. (2010). Synchronization of multiple camera videos using audio-visual features. *IEEE Transactions on Multimedia*.

Snavely, N., Seitz, S., and Szeliski, R. (2006). Photo tourism: Exploring photo collections in 3D. *ACM Transactions on Graphics*.

Snavely, N., Seitz, S. M., and Szeliski, R. (2008). Modeling the world from internet photo collections. *International Journal of Computer Vision (IJCV)*.

Strecha, C., Fransens, R., and Gool, L. V. (2004). Wide-baseline stereo from multiple views: a probabilistic account. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Strecha, C., Fransens, R., and Gool, L. V. (2006). Combined depth and outlier estimation in multi-view stereo. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Strecha, C., von Hansen, W., Gool, L. V., Fua, P., and Thoennessen, U. (2008). On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Sturm, P. F. and Maybank, S. J. (1999). On plane-based camera calibration: A general algorithm, singularities, applications. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Sun, J., Li, Y., Kang, S. B., and Shum, H.-Y. (2005). Symmetric stereo matching for occlusion handling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Sun, J., Shum, H.-Y., and Zheng, N.-N. (2002). Stereo matching using belief propagation. In *European Conference on Computer Vision (ECCV)*.

Tomasi, C. and Kanade, T. (1992). Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision (IJCV)*.

Torresani, L., Hertzmann, A., and Bregler, C. (2008). Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 30(5):878–892.

Tuytelaars, T. and Gool, L. V. (2004). Synchronizing video sequences. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tylecek, R. and Sara, R. (2010). Refinement of surface mesh for accurate multi-view reconstruction. *International Journal of VR*.

Valmadre, J. and Lucey, S. (2012). General trajectory prior for non-rigid reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Valmadre, J., Zhu, Y., Sridharan, S., and Lucey, S. (2012). Efficient articulated trajectory reconstruction using dynamic programming and filters. In *European Conference on Computer Vision (ECCV)*.

Ventura, J. and Höllerer, T. (2008). Depth compositing for augmented reality.

Vidal, R. and Abretske, D. (2006). Nonrigid shape and motion from multiple perspective views. In *European Conference on Computer Vision (ECCV)*.

Wikipedia (2014). Cayley's formula. http://en.wikipedia.org/wiki/Cayley's_formula.

Wilson, K. and Snavely, N. (2013). Network principles for sfm: Disambiguating repeated structures with local context. In *IEEE International Conference on Computer Vision (ICCV)*.

Wu, C. (2013). Towards Linear-Time Incremental Structure from Motion. In *International Conference on 3D Vision (3DV)*.

Wu, C., Agarwal, S., Curless, B., and Seitz, S. M. (2011). Multicore Bundle Adjustment. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xiao, J., Chai, J., and Kanade, T. (2004). A closed-form solution to non-rigid shape and motion recovery. In *European Conference on Computer Vision (ECCV)*.

Xiao, J., Chen, J., Yeung, D.-Y., and Quan, L. (2008). Learning two-view stereo matching. In *European Conference on Computer Vision (ECCV)*.

Yan, J., Cho, M., Zha, H., Yang, X., and Chu, S. (2015a). Multi-graph matching via affinity optimization with graduated consistency regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*.

Yan, J., Zhang, C., Zha, H., Liu, W., Yang, X., and Chu, S. M. (2015b). Discrete hyper-graph matching.

Yang, R. and Pollefeys, M. (2003). Multi-resolution real-time stereo on commodity graphics hardware. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yoon, K.-J. and Kweon, I. S. (2006). Adaptive support-weight approach for correspondence search. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*.

Zach, C. (2008). Fast and high quality fusion of depth maps. In *Proceedings of the international symposium on 3D data processing, visualization and transmission (3DPVT)*.

Zaharescu, A., Boyer, E., and Horaud, R. P. (2011). Topologyadaptive mesh deformation for surface evolution, morphing, and multi-view reconstruction. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*.

Zhang, C., Gao, J., Wang, O., Georgel, P., Yang, R., Davis, J., Frahm, J.-M., and Pollefeys, M. (2014). Personal photograph enhancement using internet photo collections. *IEEE Transactions on Visualization and Computer Graphics*.

Zhang, J., Marszalek, M., Lazebnik, S., and Schmid, C. (2006). Local features and kernels for classification of texture and object categories: A comprehensive study. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*.

Zhao, W., Chellappa, R., Phillips, P. J., and Rosenfeld, A. (2003). Face recognition: a literature survey. *Acm Computing Surveys (CSUR)*.

Zheng, E., Dunn, E., Jojic, V., and Frahm, J. (2014a). Patchmatch based joint view selection and depthmap estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zheng, E., Dunn, E., Raguram, R., and Frahm, J.-M. (2012). Efficient and scalable depthmap fusion. In *British Machine Vision Conference (BMVC)*.

Zheng, E., Ji, D., Dunn, E., and Frahm, J.-M. (2015). Sparse dynamic 3d reconstruction from unsynchronized videos. In *IEEE International Conference on Computer Vision (ICCV)*.

Zheng, E., Wang, K., Dunn, E., and Frahm, J. (2014b). Joint Object Class Sequencing and Trajectory Triangulation (JOST). In *European Conference on Computer Vision (ECCV)*.

Zheng, E. and Wu, C. (2015). Structure from Motion Using Structure-less Resection. In *IEEE International Conference on Computer Vision (ICCV)*.

Zheng, Y., Sugimoto, S., Sato, I., and Okutomi, M. (2014c). A general and simple method for camera pose and focal length determination. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhu, X. and Ramanan, D. (2012). Face detection, pose estimation, and landmark localization in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhu, Y., Cox, M., and Lucey, S. (2011). 3d motion reconstruction for real-world camera motion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhuo, W., Salzmann, M., He, X., and Liu, M. (2015). Indoor scene structure analysis for single image depth estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.