Computer Vision Techniques in Augmented Reality Systems

— 3D Understanding for Action and Placement

Wen Kou

A THESIS SUBMITTED FOR THE DEGREE OF

Master of Engineering

Department of Electrical & Computer Engineering

National University of Singapore

2016

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety.

I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Wen, Kou

April 6, 2016

Acknowledgements

I would like to thank my research supervisors, Prof. Loong-Fah Cheong and Prof. Steven Zhiying Zhou, for their guidance during my candidate years. Prof. Steven Zhiying Zhou can always give me good advice. At very beginning, I was quite confused what to do, but his advice help me find my interest. Prof. Cheong encouraged me at the beginning when I was not quite familiar with computer vision and helped build my confidence. He has also taught how to do research; especially when I proposed ideas in research, he always helped me to analyze and taught me how to analyze. Finally I could have some achievements. I could not have finished my project without his patient guidance.

I owe deep appreciation to my seniors and also my friends Zhuwen Li and Ye Luo. Zhuwen is very supportive and patient whenever I have questions about research. His suggestion always can accelerate my progress. Ye Luo gave me a lot of good advice on living in Singapore and her company made me feel not lonely far from family.

I also really appreciate my friends Fen Chen and Zhao Rui who have been my friends since undergraduate in China. I am really lucky to have them whenever I encountered any difficulties in study or life.

Special thanks to my friends and lab mates, Jiaming Guo, Tran Lam An, Shahzor Ahmad, Zhe Wu, Zhaopeng Cui, Kaimo Lin, Zhenlong Zhou, Shuaicheng Liu and Qiang Zhou for the discussions, seminars, lunches, dinners, every movie and hiking.

I also appreciate my boyfriend Qing Xu and his steadfast company. He left his hometown to Singapore in consideration of our future. I also really want to build our future together.

Last but most important, I would like to thank my parents' selfless love. Without their unconditional support, I could not have pursued my dream of studying at such an excellent university. What they want is only for me to be happy. In the future, I also want to take care of them as they do to me.

Contents

Contents v

List of Publications iii

List of Tables v

List of Figures vii

1 Introduction 1

2 Related Work 7

2.1 Structure From Motion 7

2.1.1 Factorization 7

2.1.2 Bundle Adjustment 9

2.2 Scene Understanding 10

3 Proximal Robust Factorization with Constraints from Planar Scenes 13

3.1 Technical Pre-requisites 13

3.1.1 Rank Constraint on Multiple Plane Patches across Multi-views 15

3.2 Proximal Robust Factorization with Scene and Structural Constraints 16

3.2.1 Update for $\mathbf{V}$ 18

3.2.2 Update for $\mathbf{W}$ 19

3.2.3 Update for $\mathbf{E}$ 20

3.2.4 Implementation Details 21
3.2.4.1 Scene Constraint 21
3.2.4.2 Affine Parameter Estimation 22
3.3 Experiments 25
3.3.1 Evaluation on Synthetic Data 26
3.3.1.1 Effects of Constraints on Results 26
3.3.1.2 Effect of Global Objective Function on Results 28
3.3.2 Evaluation on Real Data 31
3.3.2.1 Qualitative Evaluation 32
3.3.2.2 Quantitative Evaluation from Checkerboard Calibration 33
3.3.3 Inference of support plane from planar reconstruction . . 36

4 Conclusions and Future Works **39**

Bibliography **41**

Abstract

Augmented reality (AR) is a technique to augment virtual objects such as sound, video and graphics, etc in the real environment captured through sensors like camera and viewed from an AR device such as smart phone, glasses. Nowadays AR has been widely applied in many fields like education, art and entertainment. The development of computer vision techniques especially 3D reconstruction, object tracking are crucial for the development of AR system.

In this thesis, the aim is to obtain a dense piecewise planar reconstruction of a static scene from multiple image frames based on a factorization framework. Integrating all the relevant constraints in a global objective function, we are able to effectively leverage on the scene smoothness prior afforded by the dense formulation, as well as imposing the necessary algebraic constraints required by the shape matrix. These constraints also help to robustly decompose the measurement matrix into the underlying low-rank subspace and the sparse outlier part. Numerically, we achieve the constrained factorization and decomposition via modifying a recently proposed proximal alternating robust subspace minimization algorithm. The results show that our algorithm is effective in handling real life sequences, and outperforms other algorithms in recovering motions and dense scene estimate.

This novel planar reconstruction technique is especially beneficial for the reconstruction of indoor scenes, since artificial planes almost dominate the entire indoor scene. After we obtain the dense planar reconstruction, a simple inference based on plane geometry can be applied to infer the most likely sup-

port planes or obstacles, which is important for AR system and even for indoor navigation of robotic agent.

List of Publications

1. **Wen Kou**, Loong-Fah Cheong, and Steven Zhiying Zhou. Proximal Robust Factorization for Piecewise Planar Reconstruction. Submitted to *Computer Vision and Image Understanding (CVIU)*.

List of Tables

3.1 Quantitative Evaluation 35

List of Figures

3.1 camera model 13

3.2 Performance with and without constraints.(a) is Percentage of Correct Detection on Outlier Locations. (b) is Relative Error of the Recovered $\mathbf{W}$. (c) is Planar normal error. The blue curves(ST+SC) represent our algorithm with both structural and scene constraints, the green curves(ST) with only structural constraint, and the red curves(PARSuMi(None)) without any constraints (just PARSuMi for outlier removals). 27

3.3 Plane normal error and camera motion error under dominant lateral translation. 29

3.4 Plane normal error and camera motion error under dominant forward translation. 30

3.5 Plane normal error and camera motion error under translation and rotation comparable in magnitude. 30

3.6 Plane normal error and camera motion error under dominant rotation. 31

3.7 Visualization of reconstruction results. The various columns depict the following respectively: (a) original image, (b) superpixel segmentation overlaid on the optical flow color map, (c) and (d) depth and normal map (with missing values) and (e) and (f) depth and normal map (with missing values filled up by those of directly connected neighbors with similar intensity and texture.) 32

3.8 The respective columns depicts (a) the original image, (b) the dominant planes with each color representing one plane, and (c) the support planes with brightness indicating likelihood of being a good support. 36

Chapter 1

Introduction

AR system is nowadays widely applied in many fields, e.g entertainment, education, art, commerce, etc. The key technology fostering the AR system is computer vision technology especially 3D reconstruction, object tracking and object recognition, etc. In this thesis, we focus on 3D reconstruction and how it can be used to facilitate interacting with the scene and objects, such as placement of a virtual object and action possibilities on scene surfaces (e.g. the surface is walkable).

3D reconstruction of scenes from motion cues is a longstanding problem in computer vision. Most of the works in this area [51, 54, 5, 20, 68] are based on sparse features, and a post-processing step is required to obtain a dense reconstruction. In this thesis, we want to obtain directly a dense piecewise planar reconstruction from multiple image frames based on a factorization framework. Such a piecewise planar representation is on the one hand a more compact and efficient representation than dense 3D point cloud, and on the other hand a more informative representation than sparse 3D points, especially in man-made environment. One can for instance directly link the reconstructed planes to the notion of *occupancy* for navigation purpose [17], or more generally to the notion of *affordances* [21], a term coined by Gibson to denote properties of things that afford opportunities of interaction. A simple scheme to make these infer-

ences is to regard planes whose normal orientation is vertical and height zero as navigable, and planes which are parallel to the floor as *sittable* [29] etc.. More sophisticated schemes proposed would not only be based on the geometry of objects [6, 42, 42]; with the aids of context [8, 48], it will further encourage the interaction between objects and human [50, 35, 24, 22]. Being able to make sense of the 3D structure in this manner is evidently important for autonomous robots and augmented reality applications. For instance, in the latter, it is important to know where to place an object or even to have an avatar to manipulate or act on an object.

Despite the evident utility of such a dense piecewise planar representation, there is a paucity of works actually adopting this approach. This is despite the massive amount of works in related areas, specifically those of optical flow estimation and factorization, both with long history of research in the computer vision community. We briefly discuss some issues in these two related areas which present bottlenecks to the aforementioned approach and thus motivate our research.

Optical flow estimation is indeed still a very active area of research, in no small measure due to the release of benchmark datasets [2, 1, 3]. There has indeed been parametric model-based optical flow methods [9, 33, 45] whose underlying model is a scene with multiple planes and thus in principle could be used for recovering these planes. Yet, while optical flow has historically been understood to be a means through which eventually 3D structure and motion (a.k.a. SFM [1]) are obtained, there are nowadays not many works that utilize the flow to go on and tackle the latter part of the problem. The reasons for this state of affair are at least twofold. One of the reasons is simply the optical flows are not good enough. While the performance might look impressive according to the evaluation metrics used in the benchmarks, it is quite a different matter when being used for a geometrically exact process like SFM. For instance,

[1] Structure from motion (SFM) is often referred as estimation of 3D scene structures from 2D motion within a range of images.

[73] showed that for the discrete case of homographies, there are hidden global constraints in the form of rank of some parameter matrix. Similar constraints exist for the continuous case [72]. Since most of the existing flow methods only consider a local smoothness prior, the resulting parametric models do not necessarily obey these global constraints, and as a consequence, problems arise during the structure recovery stage. Numerically, the problem of estimating the parameters of the parametric model from the optical flows (either explicitly or implicitly) is still a significant challenge. Despite advances in optimization methods that permit discontinuity-preserving flow estimation, there are still errors remaining due to various practical reasons such as the need to perform relaxation (e.g. using L1-norm in place of L0-norm). These errors, when coupled with not knowing the number nor the boundaries of the planes, mean that there are still significant room for the parametric models to go wrong, and indeed they do go wrong.

For the second related area of factorization, the literature is also immense, though very much dominated by the discrete feature-based formulation [55, 60, 62, 15, 28, 16]. Continuous flow-based factorization works are few and far between. As a consequence, useful scene constraints such as scene smoothness and orthogonality of planes are seldom brought to bear on most factorization approaches to SFM. Indeed, as far as we can ascertain, there is no concrete practical factorization formulation for those approaches based on parametric models (be it from discrete feature or continuous flow). [73, 72] only gave theoretical formulation, whereas practical implementation is fraught with difficulties. The challenge of a practically useful formulation is manifold. Firstly, due to errors in the parametric model estimation, the input matrix to the factorization problem contains a significant number of outliers, which must be dealt with using appropriate robust factorization algorithms. Secondly, constraints of various forms should be imposed on the problems to improve the quality of the SFM solutions. These include the following: 1) the rank constraint that comes with the factor-

ization formulation, 2) what we called the structural constraints that preserve the required structures of the shape matrix (governed by the underlying physical model), and finally 3) scene constraints such as piecewise smoothness of surfaces or orthogonality between planes. Incorporating all these constraints make the factorization problem much harder. A straightforward robust implementation of the alternating least squares scheme (e.g. [47]) ignores the constraints first, and partly as a result, it does not work well (as shown in [65]); it is prohibitively costly too. Deferring the structural constraints to a post-factorization rectification step might sidestep some of the optimization difficulties but such an approach is sub-optimal as the constraint is not imposed during the minimization. In our experience, such a sub-optimal approach breaks down in the face of inevitable noise present in our problems.

Our work proposes a parametric flow-based factorization formulation that deals with all the aforementioned challenges. Optical flow is first estimated with modern optical flow technique that can handle large displacement and incorporates various best practices such as multi-scale implementation. Parameters of the affine flow (referred in Sec.3.2.4.2) that characterizes the local plane in a superpixel are then estimated. Stacking the affine parameters from all the local planes and from all views into a huge matrix, we present a robust version of factorization algorithm which factors the input matrix into a motion matrix and a shape matrix with inner dimension of six, as well as removing outliers in the form of a sparse outlier matrix. Our robust factorization is based on the proximal alternating robust subspace minimization algorithm known as PARSuMi[65] but modified to incorporate the additional constraints mentioned in the preceding paragraph. The advantage of the PARSuMi approach is that it has demonstrated significantly better performance on real practical problems with corruptions compared to other methods such as GRASTA [27], Wiberg L1 [47] and BALM [18]. It does not seek convex relaxation of any form, but rather constrains the rank and the corrupted entries' cardinality directly in their

original forms. Such faithful representation of the original problem in PARSuMi accounts for its success in solving real problems. Our modification of PARSuMi allows us to embed both the structural and scene constraints integrally into the optimization process. Firstly, we impose structural constraints on the factorized shape matrix (e.g. equality of some matrix elements) so that the shape matrix has a physical interpretation. We also enforce scene constraint such as smoothness on the resultant dense planar structure estimates. This latter constraint helps to reduce the uncertainty in decomposing the input affine parameter matrix into the low rank part and the sparse outlier matrix.

The main contributions of this thesis can be summarized as: firstly, a novel multiframe parametric model based on plane is proposed, resulting in a factorization formulation. Priors on the factors such as structural constraints and smoothness constraints (more on these in the subsequent chapter) are incorporated; secondly, the optimization derived from PARSuMi has strong capacity to deal with large amount of outliers; thirdly, the synthetic experiments and real image sequences experiments show the superiority of our algorithm to some of the other state-of-the-art algorithms; finally, a practical inference from the reconstruction result illustrates the support planes indoor.

The remainder of this thesis is organized as follows: In Chapter 2, we survey a variety of related techniques and also present some brief background on action affordance analysis coming more from the field of robotics and automation. The main technical work and the experimental results are given in Chapter 3. Chapter 4 describes the future work and presents the conclusions.

Chapter 2

Related Work

2.1 Structure From Motion

Since SFM has a long history in computer vision, with various models and attendant methods to solve this problem. Here we discuss the solution to this problem based on two categories of methods known as batch reconstruction (i.e. factorization) and incremental reconstruction (i.e. bundle adjustment).

2.1.1 Factorization

Solving SFM by factorization is a very active research field which stems from the seminal work of [57]. The simpler formulations are usually based on either orthographic or affine camera model, in which case the factorization equation is given by $\mathbf{W} = \mathbf{PX}$, where $\mathbf{W}$ is the measurement matrix of image feature positions, $\mathbf{P}$ [1]is related to the camera motions, and $\mathbf{X}$ is the 3D scene structure. Projective factorization based on perspective camera model results in a more complex factorization equation $\mathbf{\Lambda} \odot \mathbf{W} = \mathbf{PX}$ (where the left hand side now contains an unknown $\mathbf{\Lambda}$ which can be regarded as depths); this is usually solved based on iterative method [55, 60, 62, 15, 28, 16]. The most recent formulation

[1] $\mathbf{P} = \mathbf{K}\left[\mathbf{R} \mid \mathbf{t}\right]$, where $\mathbf{K} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix}$ is the camera's intrinsic parameters, $\mathbf{R}$ is the camera's rotation and $\mathbf{t}$ is the camera's translation.

[16] has an elegant treatment and can handle missing data and outliers. All the works mentioned above are based on sparse feature points and there are a few works such as [30, 44, 56] that are based on dense optical flow.

There have been quite a few factorization works that are based on a piecewise planar scene model. For the discrete formulation, [73] derived the rank constraints on homographies across multiple views based on multiple planar surfaces. Later, [14] further refined the rank four constraint over two views for practical implementation. For the continuous flow formulation, [72] showed that the parameter matrix for the planar flow is of rank six at most. There are other different formulations that are also based on planes. For instance, [40] used the area of the patch as features, in addition to the usual position features. However, the authors did not show how much the proposed area feature brings to the table. [31] developed the rank constraint in terms of the normal vector of the plane, instead of the affine parameters of its flows. This allows its algorithm to bypass the explicit estimation of optical flow, which is regarded as sensitive to noise.

Despite the vast amount of literature, including those that deal with Gaussian noise [7, 23, 34] or outliers [4, 47, 27, 59, 71], there remains a dearth of practical schemes that can handle the full set of challenges in real life SFM scenarios, chief among which are the explicit handling of outliers together with an integral handling of constraints that are applicable to the problem. Indeed, it is only recently that [65] uncovered hitherto unknown outliers inherent in the Dinosaur dataset widely used for SFM works; unfortunately, this work does not handle additional constraints too. These concerns with outliers and constraints constitute the main difference of our work with the above, and as far as we know, it is the first work that uses dense flow formulation and is capable of imposing scene and structural constraints.

2.1.2 Bundle Adjustment

Another dominant approach to solving SFM problem is known as Bundle Adjustment(BA) approach. The general BA approach can be summarized as three steps: the first step is to calculate epipolar or trifocal geometry to obtain the relative pose between camera pairs or triplets respectively; the second step is to register all camera positions and reconstruct sparse scene points in the same coordinate system and the final step is bundle adjustment to minimize the re-projection error. The first step, especially that of estimating essential matrix or fundamental matrix, has been well studied in theory and the algorithms are well established. Most estimation algorithms are based on sparse points, e.g. [53, 58, 25, 46]. There are also a few works proposed to incorporate dense flow to solve fundamental matrix e.g. [64, 63]. Several open questions remain in the second step. General approaches to register cameras are categorized into incremental method and global method. Some examples of applying incremental methods are [51, 54, 5, 20, 68]. Some of the challenges in this approach include: firstly it is inefficient to add all cameras one by one, and secondly incremental methods will suffer from drift errors as frames accumulate, due to issue with scale ambiguities. Though several compensating schemes, such as by applying a hierarchical scheme [37, 26] to merge short sequences (more than two or three views), have been proposed, global methods [32, 41, 67, 49] would be better in dealing with the drift problem. The algorithm of the final step is also well established in [61].

The work in our thesis eschews the above approach. Instead, we prefer the more elegant factorization formulation. Our chief aim is to improve the robustness of the latter so that it can be used in practical SFM scenarios.

2.2 Scene Understanding

Understanding a 3D scene also attracts a lot of attentions, since it is crucial for practical applications, e.g. AR system, navigation of robots and unmanned ground vehicle, task-completion of a robotic agent. The research in this thesis can be regarded as a kind of *affordance* analysis of scenes and objects in an environment, in the sense suggested by Gibson [21], e.g., what does a scene surface afford in terms of navigability?

Some propose to recognize or predict the *affordance* of an object by geometric shape such as the normal of the object's surface and its height w.r.t the camera coordinate [29]. Learning method becomes a powerful tool to recognize the *affordance* of different objects mostly based on RGB-D images or videos. Below are some representative works. [42] fully learns intra-class variation of objects' function; [36] applies deep learning to solve graspable objects problem; [6] argues that objects can provide more *affordances* by different configurations in an environment. For example, a bowl can contain water only when it is put upright, so [6] proposes to learn the geometry of one object and its different poses in the environment, and then it infers the function based on shape and pose at current circumstance. However, there are limitations if the method is based on object shape only, e.g., the discrimination among objects may be similar in shape but different in function, so some indicate that the action of human or robotic agents plays an important role in recognizing objects?*affordance* [50, 35], e.g., once the action of drinking is recognized, the object in hand can be container regardless of the shape. Thus, incorporation of human's or robotic agents' action on the object to infer *affordance* is proposed. [24, 22] propose to *imagine* virtual action of human or robotic agent on a specific object and [22] further proposes the inference of the object's *affordance* based on a matching procedure between object and action. Moreover, the context where actions happen [70, 19] is also suggested as the key to know the objects' *affordance*, since the context is considered as a strong priority demonstrated in some psychology

experiments [8, 48].

This thesis takes the first step towards deriving such scene and object affordances from low level input without any supervision, and our focus is on indoor scenes. We demonstrate some simple affordance analysis; one crucial task is to find the support planes from the indoor scene, i.e. where to walk, sit and put things on. For this aim, the geometry of scene and objects is adequate for inference. In particular, for walkable surface, the task after reconstruction is to find the dominant support plane from the reconstructed scene geometry.

Chapter 3

Proximal Robust Factorization with Constraints from Planar Scenes

3.1 Technical Pre-requisites

In general, a plane in the 3D space can be written as

$$Z = Z_X X + Z_Y Y + Z_0 \tag{3.1}$$

where Z_0 is the offset along the Z axis, Z_X and Z_Y are the slopes of the surface w.r.t X, Y respectively. [56] showed that the optical flow in this local planar

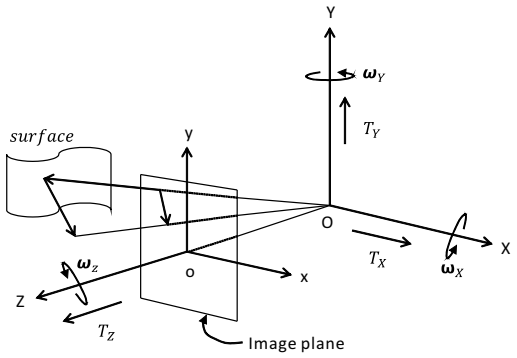


Figure 3.1: camera model

patch can be represented by a first order affine model:

$$u(x,y) = u_0 + u_x x + u_y y + O_2(x,y) \tag{3.2}$$

$$v(x,y) = v_0 + v_x x + v_y y + O_2(x,y) \tag{3.3}$$

where x, y is the image coordinate, and the six affine flow parameters are given by:

$$u_0 = -Z_3 f T_X - f\omega_Y, v_0 = -Z_3 f T_Y + f\omega_X \tag{3.4}$$

$$u_x = Z_1 T_X + Z_3 T_Z, v_x = Z_1 T_Y - \omega_Z \tag{3.5}$$

$$u_y = Z_2 T_X + \omega_Z, v_y = Z_2 T_Y + Z_3 T_Z \tag{3.6}$$

In the above, $Z_1 = \frac{Z_X}{Z_0}$ $Z_2 = \frac{Z_Y}{Z_0}$ $Z_3 = \frac{1}{Z_0}$, $\mathbf{T} = (T_X\ T_Y\ T_Z)^T$ is the camera translation velocity, $\omega = (\omega_X\ \omega_Y\ \omega_Z)^T$ is the camera angular rotation velocity, and f is the focal length of the camera. Rearranging, we obtain the following equation

$$\begin{bmatrix} u_0 \\ v_0 \\ u_x \\ v_x \\ u_y \\ v_y \end{bmatrix}^T = \begin{bmatrix} T_X\ T_Y\ T_Z\ \omega_X\ \omega_Y\ \omega_Z \end{bmatrix} \begin{bmatrix} -Z_3 f & 0 & Z_1 & 0 & Z_2 & 0 \\ 0 & -Z_3 f & 0 & Z_1 & 0 & Z_2 \\ 0 & 0 & Z_3 & 0 & 0 & Z_3 \\ 0 & f & 0 & 0 & 0 & 0 \\ -f & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \end{bmatrix} \tag{3.7}$$

3.1.1 Rank Constraint on Multiple Plane Patches across Multi-views

For the i^{th} frame and j^{th} patch, let us denote $\mathbf{W}_i^{(j)} = (u_0^{ij}\ v_0^{ij}\ u_x^{ij}\ v_x^{ij}\ u_y^{ij}\ v_y^{ij})^T$,

$\mathbf{U}_i = (T_X^i\ T_Y^i\ T_Z^i\ \omega_X^i\ \omega_Y^i\ \omega_Z^i)^T$ and $\mathbf{V}_j = \begin{bmatrix} -Z_3^{(j)}f & 0 & Z_1^{(j)} & 0 & Z_2^{(j)} & 0 \\ 0 & -Z_3^{(j)}f & 0 & Z_1^{(j)} & 0 & Z_2^{(j)} \\ 0 & 0 & Z_3^{(j)} & 0 & 0 & Z_3^{(j)} \\ 0 & f & 0 & 0 & 0 & 0 \\ -f & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \end{bmatrix}$.

Stacking them as

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_1^{(1)T} & \mathbf{W}_1^{(2)T} & \cdots & \mathbf{W}_1^{(n)T} \\ \mathbf{W}_2^{(1)T} & \mathbf{W}_2^{(2)T} & \cdots & \mathbf{W}_2^{(n)T} \\ \cdots & \cdots & \cdots & \cdots \\ \mathbf{W}_F^{(1)T} & \mathbf{W}_F^{(2)T} & \cdots & \mathbf{W}_F^{(n)T} \end{bmatrix}, \mathbf{U} = \begin{bmatrix} \mathbf{U}_1\ \mathbf{U}_2\ \mathbf{U}_3\ \cdots\ \mathbf{U}_F \end{bmatrix}^T \in R^{F\times 6}$$

and

$\mathbf{V} = \begin{bmatrix} \mathbf{V}_1\ \mathbf{V}_2\ \mathbf{V}_3\ \cdots\ \mathbf{V}_n \end{bmatrix} \in R^{6\times 6n}$, where F is number of views and n is the number of plane patches, we obtain the multi-plane multi-view formulation:

$$\mathbf{W} = \mathbf{U}\mathbf{V} \tag{3.8}$$

from which we conclude that the rank of $\mathbf{W}$ is no larger than 6. In the ensuing, we assume that the input $\mathbf{W}$ on the left hand side has been obtained; for details on how the affine parameters in $\mathbf{W}$ are computed, please refer to Sec. 3.2.4.2.

Unlike the discrete approach to SFM, the dense representation of scene in our method allows us to enforce not only a global rank constraint on $\mathbf{W}$ but also a smoothness constraint on the 3D structure matrix $\mathbf{V}$. Specifically, the factorized 3D structure matrix $\mathbf{V}$ should be such that its constituent planar patches exhibit piecewise smoothness; details of its formulation is given in Section 3.2.4.1. Exploiting this natural scene constraint helps to regularize the problem and in particular, renders the outlier detection much more robust. It is also evident

from the form of $\mathbf{V}_j$ given in the preceding paragraph that some of its elements should be subject to various equality constraints, what we referred to as structural constraint in the first section.

3.2 Proximal Robust Factorization with Scene and Structural Constraints

Various modern subspace learning techniques [65, 13, 39, 27, 47, 11] provide generic methods to recover low-rank matrices robustly in the presence of outliers, that is:

$$\begin{aligned} \min_{\mathbf{W},\mathbf{E}} \ & \|\mathbf{W} - \hat{\mathbf{W}} + \mathbf{E}\|^2 \\ s.t. \ & \text{rank}(\mathbf{W}) \leq r \\ & \|\mathbf{E}\|_0 \leq N_0 \end{aligned} \tag{3.9}$$

where $\hat{\mathbf{W}}$ is the observed data, $\mathbf{E}$ is the sparse corruption data r is upper bounded rank constraint and N_0 [1] is upper bounded number of outliers. All the $\|\cdot\|$s here are the Frobenius norm. However, these generic methods cannot incorporate the additional constraints present in our problem, namely, the constraints on the factor $\mathbf{V}$ of $\mathbf{W}$. As we will show later, imposing these constraints directly into the optimization process itself (rather than as a post processing rectification) is important in obtaining meaningful and well-posed solution. In particular, the algorithm is more likely to correctly detect the outliers, and thereby correctly decompose the observed $\hat{\mathbf{W}}$ into the noise-free, low rank $\mathbf{W}$ and the outliers $\mathbf{E}$.

To incorporate these constraints, we extend the PARSuMi [65] method. The choice of PARSuMi is based on its superior performance compared to other competing methods. We incorporate the constraints as follows:

[1] There is no conclusive number of N_0 for all problems, but we would like to choose 50% number of all entries in matrix as the upper bound since the reconstruction would fail if the number of outliers exceeds 50% in synthetic experiments.

$$
\begin{aligned}
\min_{\mathbf{W},\mathbf{E},\mathbf{V}} \ & \|\mathbf{W}-\hat{\mathbf{W}}+\mathbf{E}\|^2+\beta_1\|\mathbf{W}-\mathbf{U}\mathbf{V}\|^2 \\
& +\beta_2\sum_{j=1}^{n}\sum_{t\in N_j}\|\Omega\circ(\mathbf{V}_j-\mathbf{V}_t)\|^2 \qquad (3.10)\\
s.t. \ & \text{rank}(\mathbf{W})\leq r \\
& \|\mathbf{E}\|_0\leq N_0 \\
& \mathbf{B}\text{Vec}(\mathbf{V})=\mathbf{d}
\end{aligned}
$$

where N_j is the local neighbourhood of plane patch j. Compared with PARSuMi, the additional term $\beta_2\sum_{j=1}^{n}\sum_{t\in N_i}\|\Omega^{jt}\circ(\mathbf{V}_j-\mathbf{V}_t)\|^2$ in the objective stems from the scene smoothness constraint. Ω^{jt} is a shorthand for extracting the relevant entries of $\mathbf{V}$ to impose the following smoothness term $S^{jt}([Z_1^j,Z_2^j,Z_3^j]-[Z_1^t,Z_2^t,Z_3^t])$ and S^{jt} is a weight factor defined in Section 3.2.4.1. The last line in the constraint terms arises from the structural constraint associated with $\mathbf{V}$ ($\text{Vec}(\cdot)$ vectorizes a matrix to a column vector). We want this constraint on $\mathbf{V}$ to in turn influence the solution of $\mathbf{W}$ via $\mathbf{W}=\mathbf{U}\mathbf{V}$, thus the term $\beta_1\|\mathbf{W}-\mathbf{U}\mathbf{V}\|^2$ in the objective. In Sec3.3.1.2, we will demonstrate that it is essential both to seek a low rank subspace $\mathbf{N}$ and to satisfy the property of matrix $\mathbf{V}$

The algorithm proceeds by the minimization of the low-rank matrix $\mathbf{W}$, the structure matrix $\mathbf{V}$, and the sparse matrix $\mathbf{E}$ alternatingly until convergence. The efficiency of our method depends on the fact that the inner minimizations of $\mathbf{W}$. $\mathbf{V}$ and $\mathbf{E}$ admit efficient solutions. Basically, at step k, the three subproblems update the respective variable as follows:

$$\min_{\mathbf{V}} \ \beta_1 \|\mathbf{W}^k - \mathbf{U}^k\mathbf{V}\|^2 + \beta_2 \sum_{j=1}^{n} \sum_{t \in N_j} \|\Omega^{jt} \circ (\mathbf{V}_j - \mathbf{V}_t)\|^2 \tag{3.11}$$
$$s.t. \ \mathbf{B}\mathrm{Vec}(\mathbf{V}) = \mathbf{d}$$

$$\min_{\mathbf{W}} \|\mathbf{W} - \hat{\mathbf{W}} + \mathbf{E}^k\|^2 + \beta_1 \|\mathbf{W} - \mathbf{U}^k\mathbf{V}^{k+1}\|^2 + \beta_3 \|\mathbf{W} - \mathbf{W}^k\|^2 \tag{3.12}$$
$$s.t. \ \mathrm{rank}(\mathbf{W}) \leq r$$

$$\min_{E} \|\mathbf{W}^{k+1} - \hat{\mathbf{W}} + \mathbf{E}\|^2 + \beta_4 \|\mathbf{E} - \mathbf{E}^k\|^2 \tag{3.13}$$
$$s.t. \ \|\mathbf{E}\|_0 \leq N_0$$

Note that the above iteration is different from applying a direct alternating minimization of (3.10). We have added the proximal regularization terms $\beta_3\|\mathbf{W} - \mathbf{W}^k\|^2$ and $\beta_4\|\mathbf{E} - \mathbf{E}^k\|^2$ to make the objective functions in the subproblems coercive and hence ensuring that $\mathbf{W}^{k+1}$ and $\mathbf{E}^{k+1}$ are well defined. Empirically they are important for the critical point convergence of the sequence. We did not add a similar proximal term in (3.11), since $\|\mathbf{W}^k - \mathbf{U}^k\mathbf{V}\|^2$ can be readily shown to be equivalent to $\|\mathbf{V} - \mathbf{V}^k\|^2$ up to a multiplicative factor (using the fact $\mathbf{U}^k = \mathbf{W}^k\mathbf{V}^{kT}(\mathbf{V}^k\mathbf{V}^{kT})^{-1}$).

3.2.1 Update for V

Writing $\mathbf{v} = \mathrm{Vec}(\mathbf{V})$, (3.11) can be written as

$$\min_{\mathbf{v}} \ \frac{1}{2}\mathbf{v}^T\mathbf{Q}\mathbf{v} + \mathbf{c}^T\mathbf{v} \tag{3.14}$$
$$s.t. \ \mathbf{B}\mathbf{v} = \mathbf{d}$$

This is a standard quadratic programming (QP) problem with a large number of variables and linear constraints. We use a standard QP solver based on interior-point technique to solve for $\mathbf{v}$, from which we obtain the updated $\mathbf{V}^{k+1}$.

3.2.2 Update for W

Following [65], we do not directly solve for $\mathbf{W}$ in the subproblem; instead, we seek a low-rank $\mathbf{N} \in R^{F\times 6}$ whose column space is the underlying subspace of $\mathbf{W}$. This is a more parsimonious representation and is thus numerically more advantageous to optimize.

For $\min_{\mathbf{W}} \|\mathbf{W} - \hat{\mathbf{W}} + \mathbf{E}^k\|^2 + \beta_1 \|\mathbf{W} - \mathbf{U}^k\mathbf{V}^{k+1}\|^2 + \beta_3 \|\mathbf{W} - \mathbf{W}^k\|^2$ there exists a closed-form solution which we denote as $\mathbf{G}$:

$$\mathbf{G} = \frac{\hat{\mathbf{W}} + \beta_3 \mathbf{W}^k + \beta_1 \mathbf{U}^k \mathbf{V}^{k+1} - \mathbf{E}^k}{1 + \beta_3 + \beta_1} \tag{3.15}$$

Then the objective function to recover the subspace $\mathbf{N}$ becomes

$$\begin{aligned} &\min_{\mathbf{N},\mathbf{C}} \ \|\mathbf{G} - \mathbf{N}\mathbf{C}\|^2 \\ &s.t. \ \mathbf{N}^T\mathbf{N} = \mathbf{I}, \ \mathbf{N} \in \mathbf{R}^{F\times 6} \end{aligned} \tag{3.16}$$

Of which the optimal solutions for $\mathbf{N}$ and $\mathbf{C}$ are given by the following:

$$
\begin{aligned}
&[\mathbf{U}\ \mathbf{S}\ \mathbf{V}] = SVD(\mathbf{G}) \\
&\mathbf{U} = \begin{bmatrix} \mathbf{U}' & \mathbf{U}'' \end{bmatrix} \\
&\mathbf{S} = \begin{bmatrix} \mathbf{S}' & \mathbf{0} \\ \mathbf{0}' & \mathbf{S}'' \end{bmatrix} \\
&\mathbf{V} = \begin{bmatrix} \mathbf{V}' & \mathbf{V}'' \end{bmatrix} \qquad (3.17) \\
&\mathbf{N} = \mathbf{U}' \\
&\mathbf{C} = \mathbf{S}'\mathbf{V}'^{T} \qquad (3.18)
\end{aligned}
$$

where SVD stands for singular value decomposition, $\mathbf{U}' \in R^{F\times 6}$,$\mathbf{U}'' \in R^{F\times(F-6)}$,$\mathbf{S}' \in R^{6\times 6}$,$\mathbf{S}'' \in R^{(F-6)\times(6n-6)}$ and $\mathbf{V}' \in R^{6n\times 6}$,$\mathbf{V}'' \in R^{6n\times(6n-6)}$.

The updated $\mathbf{W}^{k+1}$ is

$$\mathbf{W}^{k+1} = \mathbf{N}\mathbf{C} \qquad (3.19)$$

$\mathbf{U}$ is also updated in this step, with $\mathbf{U}^{k+1}$ given by:

$$\mathbf{U}^{k+1} = \mathbf{W}^{k+1}\mathbf{V}^{k+1^T}(\mathbf{V}^{k+1}\mathbf{V}^{k+1^T})^{-1} \qquad (3.20)$$

3.2.3 Update for E

While a least squares minimization problem constrained by L_0-norm term is in general combinatorial in nature, the subproblem in (3.13) has a closed-form solution, as was shown by section 4.3 of [65]. The closed-form solution $\mathbf{E}'$ for $\min_E \|\mathbf{W}^{k+1} - \hat{\mathbf{W}} + \mathbf{E}\|^2 + \beta_4\|\mathbf{E} - \mathbf{E}^k\|^2$ is

$$\mathbf{E}' = \frac{\hat{\mathbf{W}} - \mathbf{W}^{k+1} + \beta_4\mathbf{E}^k}{1 + \beta_4} \qquad (3.21)$$

We first find the N_0^{th} largest element in magnitude in $\mathbf{E}'$ and denote it as E_{N_0}. Then the element (i, j) of the updated $\mathbf{E}_{ij}$ is given by:

$$\mathbf{E}_{ij}^{k+1} = \begin{cases} \mathbf{E}'_{ij} & if \ \|\mathbf{E}'_{ij}\| > \|E_{N_0}\| \\ 0 & \text{else} \end{cases} \tag{3.22}$$

Readers are referred to [65] for details.

The overall algorithm is summarized in **Algorithm** (1).

Algorithm 1 ParSuMi Factorization with Constraints

Input: Observed matrix $\hat{\mathbf{W}}$, parameter r, $\mathbf{N}_0$, Initialization $\mathbf{W}^0$, $\mathbf{U}^0$, $\mathbf{E}^0$,$k = 0$.
Repeat
1: Solve $\mathbf{v}$ (3.14) in the form of quadratic programming by interior point method and update $\mathbf{V}^{k+1}$
2: Compute $\mathbf{N}^{k+1}$ and $\mathbf{W}^{k+1}$ by (3.15), (3.19) and (3.17) respectively
3: Update $\mathbf{E}^{k+1}$ by (3.21) and (3.22)
Until $\frac{\|\mathbf{W}^{k+1}-\mathbf{W}^k\|^2}{\|\mathbf{W}^{k+1}\|^2} \leq \eta_1$ && $\frac{\|\mathbf{V}^{k+1}-\mathbf{V}^k\|^2}{\|\mathbf{V}^{k+1}\|^2} \leq \eta_2$ && $\frac{\|\mathbf{U}^{k+1}-\mathbf{U}^k\|^2}{\|\mathbf{U}^{k+1}\|^2} \leq \eta_3$ && $\frac{\|\mathbf{W}^{k+1}-\mathbf{U}^{k+1}\mathbf{V}^{k+1}\|^2}{\|\mathbf{U}^{k+1}\mathbf{V}^{k+1}\|} \leq \eta_4$

3.2.4 Implementation Details

3.2.4.1 Scene Constraint

In implementing the smoothness constraint, we use bilateral filtering to prevent smoothing across plane boundaries. In particular, we use colour intensity $\mathbf{I}_j = (I_R^j, I_G^j, I_B^j)$ and texture $\mathbf{T}_j$ to describe each local plane patch P_j. (I_R^j, I_G^j, I_B^j) is the mean intensity of the three colour channels in the patch, and the local texture $\mathbf{T}_j$ is obtained by entropy filter analysis [43] . Let patch t, $t \in N_j$ be the neighbour of patch j; the neighbourhood N_j is defined as all those plane patches t sharing part of the boundaries with patch j. The weight factor S^{jt} that weight the neighbors t adaptively in Ω^{jt} of (3.10) is then defined as

$$S^{jt} = \exp\left(-\frac{\|\mathbf{I}_j - \mathbf{I}_t\|^2}{\sigma_1^2} - \frac{\|\mathbf{T}_j - \mathbf{T}_t\|^2}{\sigma_2^2}\right)\delta(\rho_{jt} > \xi) \tag{3.23}$$

The weight factor is also determined by the occlusion states of the patches j and t. The last term in the above expression is an indicator function, which returns 0 if the mean occlusion state (ρ_{jt} of all the pixels lying on the common boundary of patches j and t) exceeds the threshold ξ, and 1 otherwise. Occlusion state of individual pixel is computed by bidirectional (forward/backward) consistency check of flow[52].

3.2.4.2 Affine Parameter Estimation

Define $\mathbf{x}_j = (u_0^j \; u_x^j \; u_y^j \; v_0^j \; v_x^j \; v_y^j)^T$ as a vector of the unknown affine flow parameters in the local patch P_j and $\mathbf{u}_j$ as the known optical flow of all the pixels in the patch. Using equations (3.2) and (3.3), and collecting all unknown affine parameter $\mathbf{x}_j$ from all patches, we can form the following linear system of equations:

$$\mathbf{A}\mathbf{x} = \mathbf{u} \tag{3.24}$$

where $\mathbf{x} = (\mathbf{x}_1^T \; \mathbf{x}_2^T \; ... \; \mathbf{x}_n^T)^T$ is the stacked affine parameters and $\mathbf{u} = (\mathbf{u}_1^T \; \mathbf{u}_2^T \; ... \; \mathbf{u}_n^T)^T$ is the stacked optical flow vectors from the n local patches.

To regularize the affine parameters, our objective function incorporates a smoothness term between neighboring patches similar to that used in the subproblem for $\mathbf{V}$ (3.11) and also applies a global rank constraint explained below:

$$\begin{aligned} &\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{u}\|^2 + \mu_1 \sum_{j=1}^{n} \sum_{t \in N_j} S_{jt} \|\mathbf{x}_j - \mathbf{x}_t\|^2 \\ &s.t \; \text{rank}(\text{reshape}(\mathbf{x})) \leq 4 \end{aligned} \tag{3.25}$$

The affine parameters $\mathbf{x} = (\mathbf{x}_1^T \; \mathbf{x}_2^T \; ... \; \mathbf{x}_n^T)^T$ are restacked by the reshape operator

to the following form $(\mathbf{x}_1\ \mathbf{x}_2\ ...\ \mathbf{x}_n)^T$, which can be written as:

$$(\mathbf{x}_1\ \mathbf{x}_2\ ...\ \mathbf{x}_n)^T = \begin{bmatrix} u_0^{(1)} & v_0^{(1)} & u_x^{(1)} & v_x^{(1)} & u_y^{(1)} & v_y^{(1)} \\ u_0^{(2)} & v_0^{(2)} & u_x^{(2)} & v_x^{(2)} & u_y^{(2)} & v_y^{(2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_0^{(n)} & v_0^{(n)} & u_x^{(n)} & v_x^{(n)} & u_y^{(n)} & v_y^{(n)} \end{bmatrix} \tag{3.26}$$

$$= \begin{bmatrix} Z_1^{(1)} & Z_2^{(1)} & Z_3^{(1)} & 1 \\ Z_1^{(2)} & Z_2^{(2)} & Z_3^{(2)} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ Z_1^{(n)} & Z_2^{(n)} & Z_3^{(n)} & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & T_x & T_y & 0 & 0 \\ 0 & 0 & 0 & 0 & T_x & T_y \\ -fT_x & -fT_y & T_z & 0 & 0 & T_z \\ -f\omega_y & f\omega_x & 0 & -\omega_z & \omega_z & 0 \end{bmatrix}$$

and thus its rank is at most 4. Empirically, we found that in real images, despite that there is already a rank 6 constraint on $\mathbf{W}$, imposing the above rank 4 constraint still brings about small improvement, so it is always recommended. To simplify the computation, we first minimize (3.25) without the rank constraint, and then do a post-hoc correction using truncated SVD.

To further improve robustness against outliers, the random sample consensus (RANSAC) is applied to each local patch. Those points deemed as outliers with regards to equation (3.24) are removed from the optimization in (3.25).

Lastly, as real optical flow is not equally reliable everywhere in the 2d image, we associate to each affine parameter vector estimate a confidence factor, which is a product of three factors. They are:

1. a factor related to the image gradient strength in the patch (specifically the mean of the top 15% largest gradient values $\sqrt{I_x^2 + I_y^2}$ in the patch)

2. a bidirectional consistency factor which is inversely proportional to the

mean difference between the forward and backward optical flow in the patch (discarding the top 20% largest flow differences as these might be genuine occlusions, and small amount of occlusions can be handled by RANSAC)

3. a factor related to the distance d_j from the center of patch j to the image center, as there is an increasing modelling error incurred by the affine camera assumption as we move towards the image periphery. Specifically, the factor is given by:

$$\rho(d_j) = \begin{cases} 1 & \mathrm{d}_j < d \\ \exp(-\alpha d_j) & \text{else} \end{cases} \tag{3.27}$$

The final confidence c^j for patch j is the mean value of the confidence factors across all views. Attaching this confidence factor to the j^{th} column of $\hat{\mathbf{W}}$ and $\mathbf{W}$ ($\hat{\mathbf{W}}^{(j)}$ and $\mathbf{W}^{(j)}$ respectively), it yields the following weighted form of our original optimization problem (3.28):

$$\begin{aligned} \min_{\mathbf{V}} \ & \sum_{j=1}^{n} \|c^j(\mathbf{W}^{(j)} - \hat{\mathbf{W}}^{(j)} + \mathbf{E}^{(j)})\|^2 + \beta_1 \sum_{j=1}^{n} \|c^j(\mathbf{W}^{(j)} - \mathbf{U}\mathbf{V}^{(j)})\|^2 \\ & + \beta_2 \sum_{j=1}^{n} \sum_{t \in N_j} \|\Omega^{jt} \circ (\mathbf{V}_j - \mathbf{V}_t)\|^2 \\ s.t. \ & \text{rank}(\mathbf{W}) \leq r \\ & \|\mathbf{E}\|_0 \leq N_0 \\ & \mathbf{B}\text{Vec}(\mathbf{V}) = \mathbf{d} \end{aligned} \tag{3.28}$$

The subproblems(3.11),(3.12),(3.13) are modified as

$$\min_{\mathbf{V}} \ \beta_1 \sum_{j=1}^{n} \|c^j(\mathbf{W}^k - \mathbf{U}^k\mathbf{V})\|^2 + \beta_2 \sum_{j=1}^{n} \sum_{t \in N_j} \|\Omega^{jt} \circ (\mathbf{V}_j - \mathbf{V}_t)\|^2 \tag{3.29}$$

$$s.t. \ \ \mathbf{B}\mathrm{Vec}(\mathbf{V}) = \mathbf{d}$$

$$\min_{\mathbf{W}} \ \sum_{j=1}^{n} \|c^j(\mathbf{W}^{(j)} - \hat{\mathbf{W}}^{(j)} + \mathbf{E}^{k(j)})\|^2 + \beta_1 \|\mathbf{W} - \mathbf{U}^k\mathbf{V}^{k+1}\|^2$$

$$+ \beta_3 \|\mathbf{W} - \mathbf{W}^k\|^2 \tag{3.30}$$

$$s.t. \ \mathrm{rank}(\mathbf{W}) \leq r$$

$$\min_{E} \ \sum_{j=1}^{n} \|c^j(\mathbf{W}^{k+1(j)} - \hat{\mathbf{W}}^{(j)} + \mathbf{E}^{(j)})\|^2 + \beta_4 \|\mathbf{E} - \mathbf{E}^k\|^2 \tag{3.31}$$

$$s.t. \ \|\mathbf{E}\|_0 \leq N_0$$

The column weighting in the second term of (3.28)warrants some explanation. While it might be argued there is no need to perform column-weighting other than that arising from the source of uncertainties (i.e. the input $\hat{\mathbf{W}}$ in the first term), doing the column weighting in the second term helps the optimization to converge to the correct solution more frequently. This is because the uncertainties in $\hat{\mathbf{W}}$ will spread to the estimate $\mathbf{W}^k$ and could lead the optimization astray; explicitly coding this uncertainties in $\mathbf{W}^k$ too will allow the inner optimization for $\mathbf{V}$ to place more trust on the prior term rather than the data term.

3.3 Experiments

In our experiments, we evaluate our algorithm using both synthetic data and real image sequences.

3.3.1 Evaluation on Synthetic Data

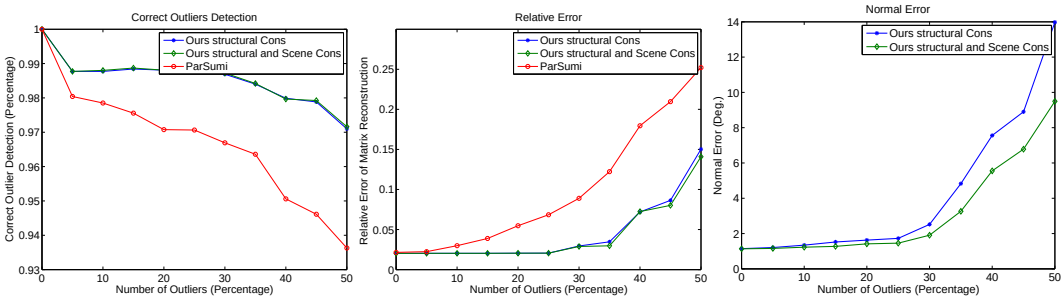
For input, we generate a sequence of motion matrices $\mathbf{U} \in R^{F\times 6}$ and n different planes in random. Each plane is segmented into p patches. Both the structure matrix $\mathbf{V} \in R^{6\times 6np}$ and the matrix $\mathbf{W} \in R^{m\times 6np}$ can then be obtained. In the computation of $\mathbf{V}$ for this synthetic case, instead of performing bilateral filtering as in (3.11), we assume we have perfect knowledge on the distribution of the scene discontinuities so that we are able to impose smoothness constraint only between patches lying in the same plane. Henceforth, we denote the ground truth version of the respective matrices with the subscript GT.

So we can have the groundtruth of $\mathbf{W}$, $\mathbf{U}$, $\mathbf{V}$ and perfect scene constraint.

3.3.1.1 Effects of Constraints on Results

In our first experiment on synthetic data, we evaluate the effectiveness of our algorithm on sparse outlier detection vis-à-vis the scene constraint and the structural constraint. A series of frames ($F = 19$) with smooth motions is generated; $n = 10$ planes are created and each single plane is segmented into $p = 6$ patches. We add outliers to $0-50\%$ of the entries of $\mathbf{W}$. The magnitudes of these outliers are set to the magnitude of the largest entries in $\mathbf{W}$. In addition, dense Gaussian noise $N(0, \sigma)$ is also added to each column of $\mathbf{W}$, with σ set to 5% of the mean magnitude of the entries in that column. This serves as the noisy input $\hat{\mathbf{W}}$ to the factorization algorithm.

We then apply our algorithm (**Algorithm** 1) to $\hat{\mathbf{W}}$ to obtain $\mathbf{W}$, $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{E}$. To demonstrate the crucial role played by the scene constraint and the structural constraint, we also apply two variants of **Algorithm** 1: one where we only apply the scene constraint, and one where we just use the original PARSuMi algorithm (i.e. without any constraints). Three measures are used to demonstrate the difference brought about by these constraints. First, we measure the amount of outliers that are correctly detected. This is given by the percentage of correct



(a) (b) (c)

Figure 3.2: Performance with and without constraints.(a) is Percentage of Correct Detection on Outlier Locations. (b) is Relative Error of the Recovered $\mathbf{W}$. (c) is Planar normal error. The blue curves(ST+SC) represent our algorithm with both structural and scene constraints, the green curves(ST) with only structural constraint, and the red curves(PARSuMi(None)) without any constraints (just PARSuMi for outlier removals).

detection of the outlier locations:

$$\frac{N_{correct}}{\max(N_{GT}, N_{recovered})} \tag{3.32}$$

where $N_{correct}$ is the number of outliers whose supports are correctly identified, N_{GT} is the actual number of outliers and $N_{recovered}$ is the number of recovered outliers. We also measure the extent to which the correct $\mathbf{W}$ is recovered:

$$\frac{\|\mathbf{W} - \mathbf{W}_{GT}\|^2}{\|\mathbf{W}_{GT}\|^2} \tag{3.33}$$

The above measure may not tell the full story as far as estimation of scene structure is concerned. Thus we also measure the average directional error in the recovered plane normal. The results are shown in Fig. 3.2. It is clear from Fig. 3.2 (a) that leveraging on the scene and structural constraints, our algorithm is better able to recover the support of $\mathbf{E}$ under increasing amount of outliers. Without any unexpected in Fig.3.2 (b) the accuracy of the recovered $\mathbf{W}$ depending on the aforementioned constraints is much higher than on PARSuMi with no constraints and just outlier removals. Finally, Fig. 3.2 (c) shows that even when the recovered $\mathbf{W}$s seem to exhibit only small differences (between our algorithm and one where only structural constraint is used), these differences

could be important as far as scene recovery is concerned.

3.3.1.2 Effect of Global Objective Function on Results

Having seen the crucial role played by the scene and structural constraints, we want to evaluate in this subsection the importance of incorporating these constraints integrally into a global objective function, like what we did in **Algorithm** 1. This is contrasted against the following schemes which sequentially remove outliers (via ParSuMi) and then enforce constraints separately via either alternating least squares (ALS) or rectification:

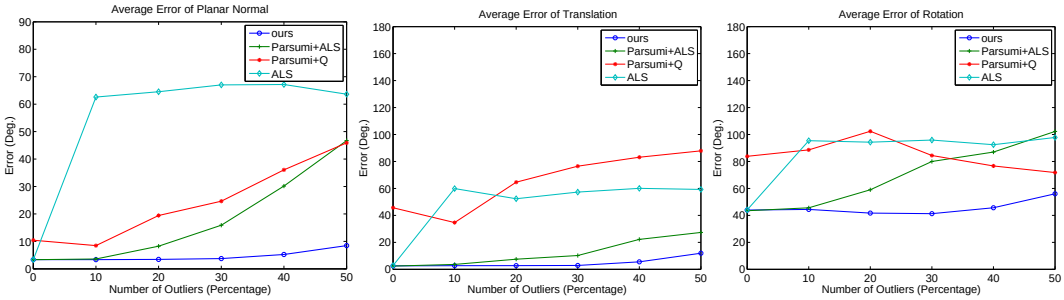
1. ParSuMi (for outlier removal) + ALS (to incorporate constraints)

2. ParSuMi (for outlier removal) + rectification matrix $\mathbf{Q}$ (to enforce constraints)

Note that the original ParSuMi algorithm performs outlier removal and factorization but does not handle constraints. Thus, in the first method above, the ALS further improves the factorization by incorporating the constraints and then solves for $\mathbf{U}$ and $\mathbf{V}$ alternatingly until convergence. The objective function for the ALS is as follows:

$$\min_{\mathbf{U},\mathbf{V}} \ \|\mathbf{W}-\mathbf{U}\mathbf{V}\|^2 + \beta \sum_{j=1}^{n} \sum_{t\in N_j} \|\Omega^{jt} \circ (\mathbf{V}_j - \mathbf{V}_t)\|^2 \qquad (3.34)$$

$$s.t. \ \mathbf{B}\mathbf{Vec}(\mathbf{V}) = \mathbf{d}$$



Plane Normal Error Camera Translation Error Camera Rotation Error

Figure 3.3: Plane normal error and camera motion error under dominant lateral translation.

where $\mathbf{W}$ is the denoised output by ParSuMi. In the k^{th} iteration, we update $\mathbf{U}^{k+1}$ and $\mathbf{V}^{k+1}$ as follows:

$$\mathbf{V}^{k+1} = \arg\min_{\mathbf{V}} \|\mathbf{W} - \mathbf{U}^k\mathbf{V}\|^2 + \beta \sum_{j=1}^{n} \sum_{t \in N_j} \|\Omega^{jt} \circ (\mathbf{V}_j - \mathbf{V}_t)\|^2 \tag{3.35}$$

$$s.t. \mathbf{BVec}(\mathbf{V}) = \mathbf{d}$$

$$\mathbf{U}^{k+1} = \arg\min_{\mathbf{U}} \|\mathbf{W} - \mathbf{U}\mathbf{V}^{k+1}\|^2 \tag{3.36}$$

In the second method, we seek a matrix $\mathbf{Q}$ that will rectify the $\mathbf{U}$ and $\mathbf{V}$ output by PARSuMi, so that the conditions required by the constraints are fulfilled:

$$\min_{\mathbf{Q}} \sum_{j=1}^{n} \sum_{t \in N_j} \|\Omega^{jt} \circ ((\mathbf{QV})_j - (\mathbf{QV})_t)\|^2 \tag{3.37}$$

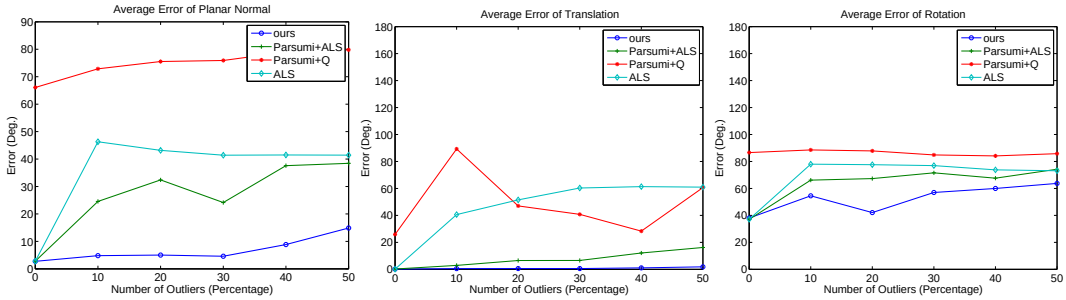
$$s.t. \mathbf{BVec}(\mathbf{QV}) = \mathbf{d}$$

The rectified solutions $\hat{\mathbf{U}}$ and $\hat{\mathbf{V}}$ are obtained as follows:

$$\hat{\mathbf{U}} = \mathbf{U}\mathbf{Q}^{-1} \tag{3.38}$$

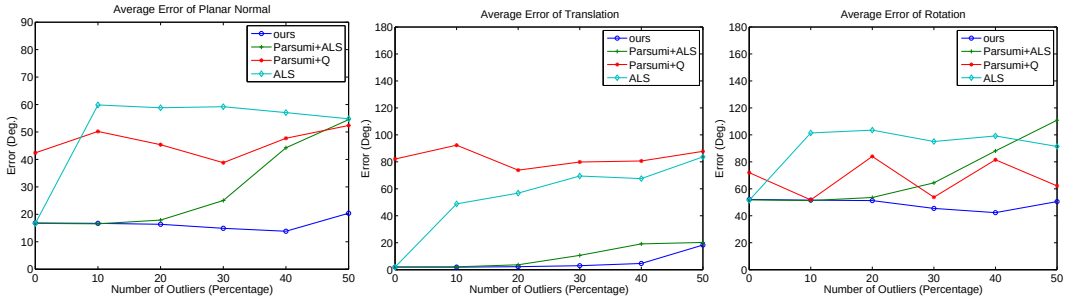
$$\hat{\mathbf{V}} = \mathbf{Q}\mathbf{V} \tag{3.39}$$

Lastly, we also tried out a variant with just ALS to enforce the constraints, without any outlier removal.



Plane Normal Error Camera Translation Error Camera Rotation Error

Figure 3.4: Plane normal error and camera motion error under dominant forward translation.

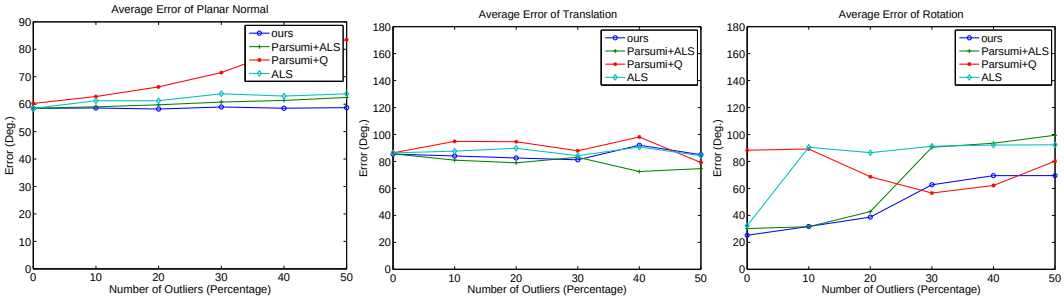


Plane Normal Error Camera Translation Error Camera Rotation Error

Figure 3.5: Plane normal error and camera motion error under translation and rotation comparable in magnitude.

Four types of camera motions are evaluated: translation dominant, with direction either forward or lateral, translation (lateral) and rotation comparable in magnitude, rotation dominant. The remaining parameters such as F, n, p are the same as in the preceding section, so are the amount of outliers and the type of Gaussian noise added. Three measures are used to evaluate the performance of our algorithm and the various alternatives with non-global objective functions: average errors in the plane normals recovered, and directional errors in the camera translation and camera rotation.

Fig. 3.3 and Fig. 3.4 show the error performance under a translation dominant camera motion. Even though the accuracy of the recovered camera motion under the lateral (Fig. 3.3) and forward camera motion (Fig. 3.4) is very similar, the recovered 3D structure under lateral translation is much better than that under a forward translation. This is not surprising given previous findings [69] about how lateral translation yields more reliable depth cues. In all these set-



Plane Normal Error Camera Translation Error Camera Rotation Error

Figure 3.6: Plane normal error and camera motion error under dominant rotation.

tings, our algorithm achieved the best performance. Not only it retained the strong outlier removal capability of the original PARSuMi, all aspects of performance are improved by the addition of relevant constraints.

Fig. 3.5 shows the error performance under the case of translational and rotational flows having comparable magnitudes. Our algorithm and the "PARSuMi + ALS" variant achieved very similar results as in Fig. 3.3, whereas other variants experienced some fluctuations in performance.

Fig. 3.6 shows the error performance under a rotation dominant camera motion. Clearly, the strong rotation makes the recovery of translation and scene structure much more problematic. In this case, there is little difference between our algorithm and the "PARSuMi + ALS" variant.

3.3.2 Evaluation on Real Data

For the real image sequences, different scenes are captured by a calibrated handheld camera. We next obtain the 2D local patches by the temporal superpixel (TSP) algorithm [12] and dense optical flow by the DeepFlow algorithm [66]. The TSP algorithm is chosen because it yields temporally consistent superpixel labels throughout the video sequence, with the label terminated if the superpixel is occluded in a particular frame. This greatly faciliates our factorization setup (we drop patches which are not visible in every frame). The method described in Section 3.2.4.2 is then used to estimate the affine parameters and construct the

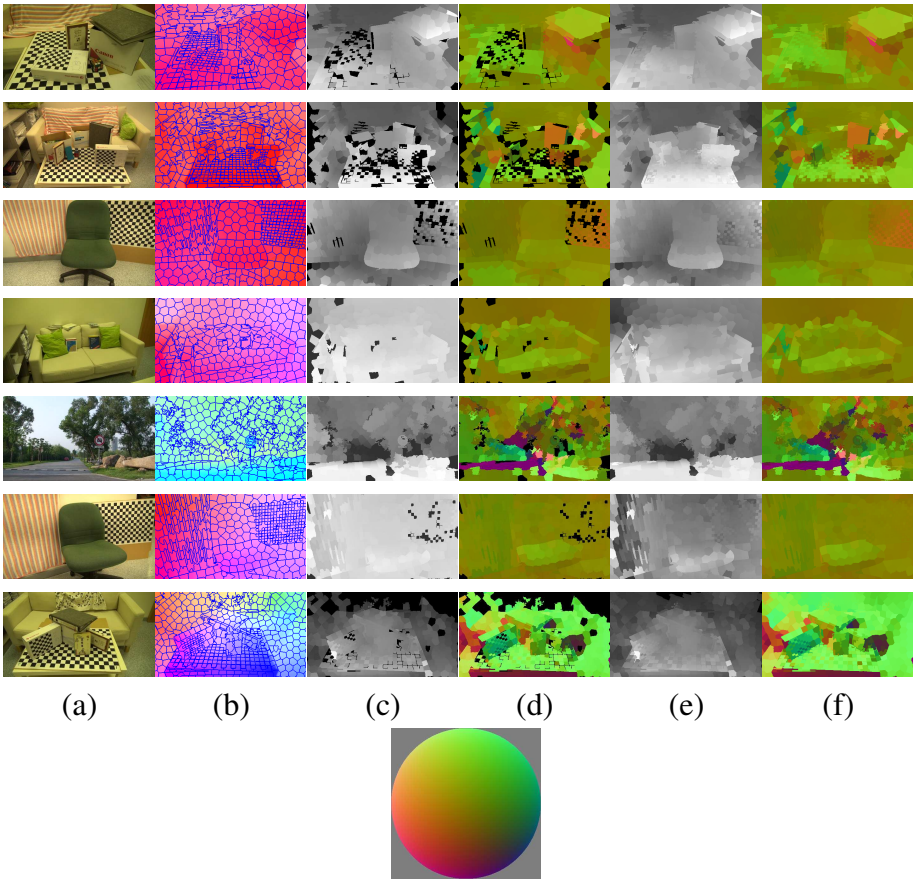


Figure 3.7: Visualization of reconstruction results. The various columns depict the following respectively: (a) original image, (b) superpixel segmentation overlaid on the optical flow color map, (c) and (d) depth and normal map (with missing values) and (e) and (f) depth and normal map (with missing values filled up by those of directly connected neighbors with similar intensity and texture.)

input matrix $\hat{\mathbf{W}}$ to the factorization process. We choose $\beta_1 = \frac{10^{-6}}{\max(F,6n)} * 1.02^k$ (where k is the iteration number), $\frac{\beta_2}{\beta_1} = 2.5$ and the proximal regularization term $\beta_3 = \beta_4 = \frac{10^{-3}}{\max(F,6n)}$. For the distance threshold d, we choose $d = \text{atan}\,(18^\circ) * f$. In the ensuing subsections, we will present both qualitative and quantitative evaluations.

3.3.2.1 Qualitative Evaluation

We present a visualization of the depth and normal map of the real image sequences to demonstrate our result qualitatively. For the depth map, the objects nearer to the camera will be brighter than those further away from the camera (with the nearest depth set to 255 and the furthest set to 50). For the normal map,

we use the colors on a normal sphere to indicate the normal directions (last row in Fig. 3.7).

Fig.3.7 shows the results from seven different sequences from both indoor and outdoor scenes and with different motion types. The first five sequences are captured by camera moving with predominantly lateral motion, with the rest predominantly forward motion. The first four image sequences are indoor scenes and the fifth sequence is the *road* sequence from [74]. As can be seen, for the indoor scenes, our algorithm successfully extracted different planes under a predominantly lateral motion. For the outdoor scene, most of the surfaces are not strictly planar, but sufficiently planar when viewed from afar and thus our algorithm also successfully reconstructed the scene in sequence five. In general, the structure recovery is much better by camera with lateral motion than that with forward motion (this is demonstrated in synthetic experiment too), not only because of the reasons [69] mentioned in the preceding synthetic experiments, but also because for optical flow generated from real image sequence, a forward motion suffers greater modelling errors than that of a lateral motion. Nevertheless, our algorithm successfully reconstructs parts of the scenes. In sequence 6, other than the noisy depths on the left side of the image, the rest of the depths and the various plane normals are correctly recovered. In particular, the horizontal support plane of the chair is successfully recovered. In sequence 7, the objects on the table are reconstructed well, but not so for most parts of the couch. Together with sequence 6, the results for these two sequences demonstrate the sensitivity of depth reconstruction under forward motion.

3.3.2.2 Quantitative Evaluation from Checkerboard Calibration

We use checkerboard calibration [10] to obtain planar orientation for the checkerboard with respect to the first camera, as well as the camera rotation and absolute translation for each subsequent frame. Specifically, from checkerboard calibration, each camera frame rotation R_i $i \in \{1, 2, 3, ..., F\}$ and absolute transla-

tion T_i $i \in \{1, 2, 3, ..., F\}$ w.r.t the checkerboard coordinate frame can be obtained. We set the first frame as the reference frame. By equation (3.40),(3.41), the 'groundtruth' rotation R_{1m} $m \in \{2, 3, 4..., F\}$ and translation T_{1j} $j \in \{2, 3, 4..., n\}$ w.r.t the first camera's coordinate frame can be calculated.

$$R_{1m} = R_m / R_1 \tag{3.40}$$

$$T_{1m} = R_{1m} T_1 - T_m \tag{3.41}$$

T_{1m} is the translation velocity of camera motion. By equation (3.42) we can obtain the camera angular velocity $\omega = [\omega_x(t), \omega_y(t), \omega_z(t)]$.

$$R_{1m} = I + W_{1m} dt \tag{3.42}$$

where $W = \begin{bmatrix} 0 & -\omega_Z(t) & \omega_Y(t) \\ \omega_Z(t) & 0 & -\omega_X(t) \\ -\omega_Y(t) & \omega_X(t) & 0 \end{bmatrix}$ is the angular velocity tensor. The normal direction of the checkerboard w.r.t. the checkerboard coordinate frame is $n_c = (0, 0, 1)$ and it is transferred onto the first frame coordinate by $R_1 n_c$.

We then compare our results with those of the multiview dense reconstruction by G. Zhang et al [74] and the sparse reconstruction algorithm by Y. Dai et al [16], for both motion and structure (represented by only the normal of the checkerboard). The algorithm by [74] is representative of those dense reconstruction algorithms using traditional Bundle Adjustment approach, whereas that of [16] is representative of state-of-the-art reconstruction algorithms based on factorization. The sequences compared are the first, second, and seventh in Fig.3.7, containing 10, 17, and 11 images respectively [1]. Since in our method the checkerboard is oversegmented into multiple superpixel patches by TSP, we

[1]While sequence 3 and 6 also contain checkerboard for ground truth calibration, the checkerboard is far from the camera and with an almost fronto-parallel orientation, configuration that is not conducive for accurate calibration results.

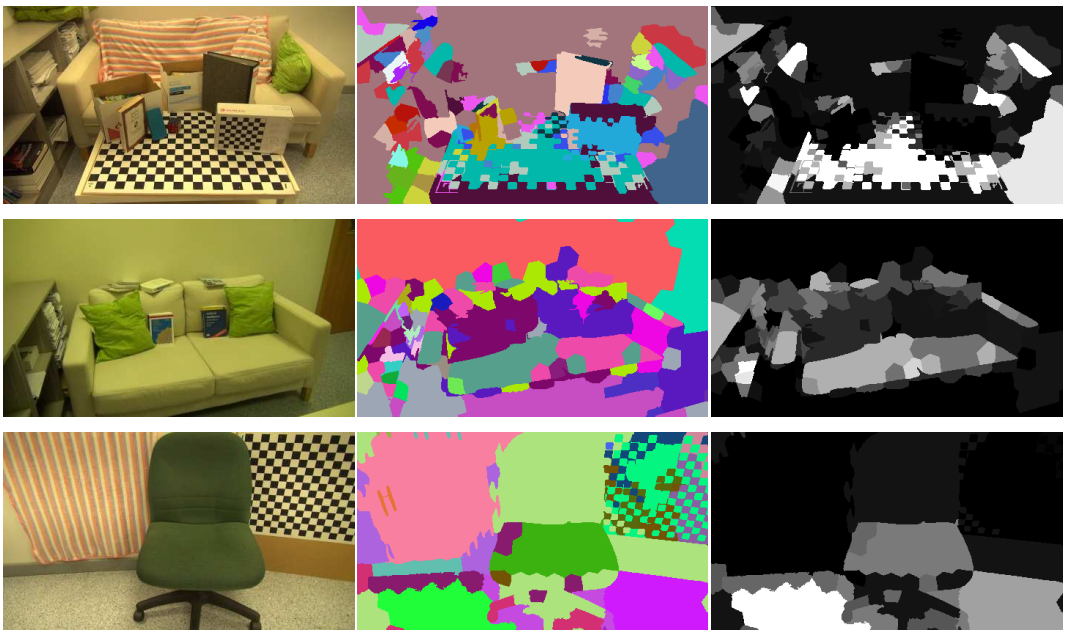
Table 3.1: Quantitative Evaluation

Sequence Name	Normal Err. (Deg.)			Translation Err. (Deg.)			Rotation Err. (Deg.)		
	ours	G. Zhang[74]	Y. Dai [16]	ours	G. Zhang [74]	Y. Dai[16]	ours	G. Zhang [74]	Y. Dai[16]
SQ. 1	**7.4770**	34.7304	51.8554	**8.1465**	13.8377	53.3743	73.5671	**19.4221**	116.5598
SQ. 2	**8.9901**	20.9990	20.0714	**2.7288**	3.2133	5.2512	97.1477	102.8634	58.2363
SQ. 7	49.6045	**19.2127**	122.8575	**9.6247**	17.4474	113.2668	90.4913	*33.1826*	78.9828

use the mean reconstructed normal of these patches to represent the orientation estimate of the checkerboard. For [74] and [16], the normal for each patch is obtained from the reconstruction points on the checkerboard by fitting a planar equation $n_X X + n_Y Y + n_Z Z + c = 0$ to the points (X, Y, Z), from which (n_X, n_Y, n_Z) is recovered as the orientation of the planar patch. For the motion estimates, we compare the errors in the translational and rotational directions. The final error is the average error across all views

The results are shown in TABLE 3.1. Note that the three sequences are captured by camera where the translation significantly dominates the rotation, so the figures on rotation accuracy are not very meaningful. As can be seen, our algorithm achieves the best performance in translation accuracy for all the three sequences, and best performance in the planar orientation for two of the sequences under lateral motion. For sequence 7 captured under forward camera motion, the recovery of planar orientation is much more sensitive to noise. The reason for this sensitivity is clear if we look at equations (3.4): the plane gradients ($Z_X = -\frac{n_X}{n_Z}$,$Z_Y = -\frac{n_Y}{n_Z}$) are coupled to and thus carried by the lateral translational terms T_X and T_Y. If there is little or no lateral translation, it would be very difficult to recover a good estimate of the plane gradients in the X- and Y-directions.

As an aside, we note that if the camera motion is known to be predominantly lateral (under which the convergence is much less prone to errors and initialization), a larger β_1 and distance threshold d can be used for a faster convergence rate.



(a) (b) (c)

Figure 3.8: The respective columns depicts (a) the original image, (b) the dominant planes with each color representing one plane, and (c) the support planes with brightness indicating likelihood of being a good support.

3.3.3 Inference of support plane from planar reconstruction

One direct application from the planar reconstruction is to infer the affordance of the scene. In this experiment, the dominant support planes upon which one can perform actions like sitting, walking and placement of objects, can be inferred from the orientation of the recovered planes. Assume there is a robotic agent standing upright such that its visual axis is more or less parallel to the ground plane. Those planes with normals near to the vertical y-axis of the robot's 'eye' (Fig.3.1) would be more capable of supporting things. Our scheme is that after clustering over-segmented superpixel patches into several dominant planes using [38], a ranking of support likelihood based on the affinity between the plane's normal and the y-axis will be performed. For the clustering, the affinity between 2 patches is based on the similarity of their normal $\overrightarrow{n} = (n_X, n_Y, n_Z)$ and offset c , specifically, $A_{ij} = \exp\left(-\frac{\arccos(\overrightarrow{n}^i \cdot \overrightarrow{n}^j) + \alpha(c^i - c^j)^2}{\sigma^2}\right)$, where $\cdot$ is the dot product of 2 vectors. The result is shown in Fig.3.8. The brighter area indicates planes with higher support likelihood. As we can see, despite a few

clustering mistakes, (e.g. in the first row, the boxes and part of the book shelf are merged into the background), the dominant support planes are still found, e.g. the table and the floor in the first row, the support planes of the couch and shelf in the second row and the support plane of the chair and floor in the last row. The information of support planes can be very important for robotic agents to decide an action or for virtual object placements in some AR applications.

Chapter 4

Conclusions and Future Works

In this thesis, a robust and practical factorization algorithm to recover 3D motion and dense planar scene is proposed, given noisy optical flow as input. To achieve this, we first developed the numerical machinery that can effectively integrate all the relevant constraints in a global objective function, including scene smoothness and algebraic constraint associated with the shape matrix. As a consequence, our algorithm can robustly decompose the measurement matrix into the underlying low-rank subspace and the inevitable outliers that are present in real life SFM scenarios. We tested the algorithm under a wide variety of motion-scene configurations. The results show that integrating all these relevant constraints is crucial for reliable outlier removal, and for yielding a shape matrix that is meaningful and interpretable. Furthermore a simple and effective inference of support planes in the 3D indoor scene is proposed for the possible actions on and placement of objects, which is important for indoor robotic agent navigation and Augmented Reality System e.g indoor entertainment application.

For future works, firstly, a more global constraint on the estimation of optical flow can be investigated so as to obtain more robust results based only on a local smoothness prior. These constraints include rigidity, or more generally, multiple rigid motions. The challenge lies in how to incorporate them in an integral manner into our factorization formulation (as opposed to a preprocessing step

in our current formulation) which is a non-trivial problem.

For a second related future work, updating the changes in a scene is a frequent need.The update can be done with more semantics than what has been addressed in this thesis. The work of my thesis can be the first step to obtain a dense reconstruction result in a scene. Object recognition could then be involved using the planar surfaces and appearance information. By combining the two steps, a reliable dictionary for the scene can be built. Subsequent update can be performed on the dictionary.

Bibliography

[1] Hci robust vision challenge. http://hci.iwr.uni-heidelberg.de/Static/challenge2012/. 2

[2] Kitti vision benchmark. http://www.cvlibs.net/datasets/kitti/. 2

[3] Mpi sintel benchmark. http://sintel.is.tue.mpg.de/. 2

[4] H. Aanaes, R. Fisker, K. Åström, and J. M. Carstensen. Robust factorization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(9):1215–1225, 2002. 8

[5] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Building rome in a day. In *ICCV*, 2009. 1, 9

[6] A. Aldoma, F. Tombari, and M. Vincze. Supervised learning of hidden and non-hidden 0-order affordances and detection in real scenes. In *ICRA*, 2012. 2, 10

[7] P. Anandan and M. Irani. Factorization with uncertainty. *International Journal of Computer Vision*, 49(2-3):101–116, 2002. 8

[8] I. Biederman, R. J. Mezzanotte, and J. C. Rabinowitz. Scene perception: Detecting and judging objects undergoing relational violations. *Cognitive psychology*, 14(2):143–177, 1982. 2, 11

[9] M. J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, 1996. 2

[10] Caltech. Camera Calibration Toolbox for Matlab. http://http://www.vision.caltech.edu/bouguetj/calib_doc/. 33

[11] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11, 2011. 16

[12] J. Chang, D. Wei, and J. W. F. III. A video representation using temporal superpixels. In *2013 CVPR*. 31

[13] P. Chen. Optimization algorithms on subspaces: Revisiting missing data problem in low-rank matrix. *International Journal of Computer Vision*, 80(1):125–142, 2008. 16

[14] P. Chen and D. Suter. Rank constraints for homographies over two views: Revisiting the rank four constraint. *International Journal of Computer Vision*, 81(2):205–225, 2009. 8

[15] Q. Chen and G. G. Medioni. Efficient iterative solution to m-view projective reconstruction problem. In *CVPR*, 1999. 3, 7

[16] Y. Dai, H. Li, and M. He. Projective multiview structure and motion from element-wise factorization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(9):2238–2251, 2013. 3, 7, 8, 34, 35

[17] R. Danescu, F. Oniga, and S. Nedevschi. Modeling and tracking the driving environment with a particle-based occupancy grid. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1331–1342, 2011. 1

[18] A. Del Bue, J. M. F. Xavier, L. de Agapito, and M. Paladini. Bilinear modeling via augmented lagrange multipliers (BALM). *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(8):1496–1508, 2012. 4

[19] S. K. Divvala, D. Hoiem, J. H. Hays, A. Efros, M. Hebert, et al. An empirical study of context in object detection. In *CVPR*, 2009. 10

[20] J. Frahm, P. F. Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y. Jen, E. Dunn, B. Clipp, and S. Lazebnik. Building rome on a cloudless day. In *ECCV*, 2010. 1, 9

[21] J. J. Gibson. *The ecological approach to visual perception: classic edition*. Psychology Press, 2014. 1, 10

[22] H. Grabner, J. Gall, and L. J. V. Gool. What makes a chair a chair? In *CVPR*, 2011. 2, 10

[23] A. Gruber and Y. Weiss. Factorization with uncertainty and missing data: Exploiting temporal coherence. In *NIPS*, 2003. 8

[24] A. Gupta and L. S. Davis. Objects in action: An approach for combining action understanding and object perception. In *CVPR*, 2007. 2, 10

[25] R. I. Hartley. In defense of the eight-point algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(6):580–593, 1997. 9

[26] M. Havlena, A. Torii, J. Knopp, and T. Pajdla. Randomized structure from motion based on atomic 3d models from camera triplets. In *CVPR*, 2009. 9

[27] J. He, L. Balzano, and J. C. S. Lui. Online robust subspace tracking from partial information. *CoRR*, 2011. 4, 8, 16

[28] A. Heyden, R. Berthilsson, and G. Sparr. An iterative factorization method for projective structure and motion from image sequences. *Image Vision Comput.*, 17(13):981–991, 1999. 3, 7

[29] D. Holz, S. Holzer, R. B. Rusu, and S. Behnke. Real-time plane segmentation using RGB-D cameras. In *RoboCup International Symposium, 2011]*, pages 306–317, 2011. 2, 10

[30] M. Irani. Multi-frame correspondence estimation using subspace constraints. *International Journal of Computer Vision*, 48(3):173–194, 2002. 8

[31] H. Ji and C. Fermüller. A 3d shape constraint on video. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(6):1018–1023, 2006. 8

[32] N. Jiang, Z. Cui, and P. Tan. A global linear method for camera pose registration. In *ICCV*, 2013. 9

[33] S. X. Ju, M. J. Black, and A. D. Jepson. Skin and bones: Multi-layer, locally affine, optical flow and regularization with transparency. In *CVPR96*. 2

[34] T. Kanade and D. D. Morris. Factorization methods for structure from motion. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 356(1740):1153–1173, 1998. 8

[35] H. Kjellström, J. Romero, and D. Kragic. Visual object-action recognition: Inferring object affordances from human demonstration. *Computer Vision and Image Understanding*, 115(1):81–90, 2011. 2, 10

[36] I. Lenz, H. Lee, and A. Saxena. Deep learning for detecting robotic grasps. *I. J. Robotic Res.*, 34(4-5):705–724, 2015. 10

[37] M. Lhuillier and L. Quan. A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(3):418–433, 2005. 9

[38] Z. Li, L.-F. Cheong, and S. Z. Zhou. Scams: Simultaneous clustering and model selection. In *CVPR*, 2014. 36

[39] Z. Lin, M. Chen, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*, 2010. 16

[40] J. Ma and N. Ahuja. Dense shape and motion from region correspondences by factorization. In *CVPR*, 1998. 8

[41] P. Moulon, P. Monasse, and R. Marlet. Global fusion of relative motions for robust, accurate and scalable structure from motion. In *ICCV*, 2013. 9

[42] A. Myers, C. L. Teo, C. Fermüller, and Y. Aloimonos. Affordance detection of tool parts from geometric features. In *ICRA*, pages 1374–1381, 2015. 2, 10

[43] M. Nabi. Digital image processing using matlab. 21

[44] S. Negahdaripour and S. Lee. Motion recovery from image sequences using only first order optical flow information. *International Journal of Computer Vision*, 9(3):163–184, 1992. 8

[45] T. Nir, A. M. Bruckstein, and R. Kimmel. Over-parameterized variational optical flow. *International Journal of Computer Vision*, 76(2):205–216, 2008. 2

[46] D. Nistér. An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(6):756–777, 2004. 9

[47] T. Okatani and K. Deguchi. On the wiberg algorithm for matrix factorization in the presence of missing components. *International Journal of Computer Vision*, 72(3):329–337, 2007. 4, 8, 16

[48] A. Oliva and A. Torralba. The role of context in object recognition. *Trends in cognitive sciences*, 11(12):520–527, 2007. 2, 11

[49] O. Özyesil and A. Singer. Robust camera location estimation by convex programming. In *CVPR*, 2015. 9

[50] P. Peursum, G. A. W. West, and S. Venkatesh. Combining image regions and human activity for indirect object recognition in indoor wide-angle views. In *ICCV*, 2005. 2, 10

[51] M. Pollefeys, L. J. V. Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a hand-held camera. *International Journal of Computer Vision*, 59(3):207–232, 2004. 1, 9

[52] M. Proesmans, L. J. V. Gool, E. J. Pauwels, and A. Oosterlinck. Determination of optical flow and its discontinuities using non-linear diffusion. In *ECCV*, pages 295–304, 1994. 22

[53] L. Quan. Invariants of six points and projective reconstruction from three uncalibrated images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(1):34–46, 1995. 9

[54] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. *ACM Trans. Graph.*, 25(3):835–846, 2006. 1, 9

[55] P. F. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *ECCV*, 1996. 3, 7

[56] M. Subbarao. Interpretation of image flow: A spatio-temporal approach. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(3):266–278, 1989. 8, 13

[57] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992. 7

[58] P. H. S. Torr and D. W. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *International Journal of Computer Vision*, 24(3):271–300, 1997. 9

[59] M. Trajkovic and M. Hedley. Robust recursive structure and motion recovery under affine projection. In *BMVC*, 1997. 8

[60] B. Triggs. Factorization methods for projective structure and motion. In *CVPR*, 1996. 3, 7

[61] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment - A modern synthesis. In *Vision Algorithms: Theory and Practice, International Workshop on Vision Algorithms, held during ICCV '99, Corfu, Greece, September 21-22, 1999, Proceedings*, pages 298–372, 1999. 9

[62] T. Ueshiba and F. Tomita. A factorization method for projective and euclidean reconstruction from multiple perspective views via iterative depth estimation. In *ECCV*, 1998. 3, 7

[63] L. Valgaerts, A. Bruhn, M. Mainberger, and J. Weickert. Dense versus sparse approaches for estimating the fundamental matrix. *International Journal of Computer Vision*, 96(2):212–234, 2012. 9

[64] L. Valgaerts, A. Bruhn, and J. Weickert. A variational model for the joint recovery of the fundamental matrix and the optical flow. In *Pattern Recognition, 30th DAGM Symposium, Munich, Germany, June 10-13, 2008, Proceedings*, pages 314–324, 2008. 9

[65] Y. Wang, C. M. Lee, L. Cheong, and K. Toh. Practical matrix completion and corruption recovery using proximal alternating robust subspace minimization. *International Journal of Computer Vision*, 111(3):315–344, 2015. 4, 8, 16, 19, 20, 21

[66] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. Deepflow: Large displacement optical flow with deep matching. In *2013 ICCV*. 31

[67] K. Wilson and N. Snavely. Robust global translations with 1dsfm. In *ECCV*, 2014. 9

[68] C. Wu. Towards linear-time incremental structure from motion. In *3DV*, pages 127–134, 2013. 1, 9

[69] T. Xiang and L. F. Cheong. Understanding the behavior of SFM algorithms: A geometric approach. *International Journal of Computer Vision*, 51(2):111–137, 2003. 30, 33

[70] B. Yao and F. Li. Modeling mutual context of object and human pose in human-object interaction activities. In *CVPR*, 2010. 10

[71] A. Zaharescu and R. Horaud. Robust factorization methods using a gaussian/uniform mixture model. *International Journal of Computer Vision*, 81(3):240–258, 2009. 8

[72] L. Zelnik-Manor and M. Irani. Multi-frame estimation of planar motion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(10):1105–1116, 2000. 3, 8

[73] L. Zelnik-Manor and M. Irani. Multiview constraints on homographies. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(2):214–223, 2002. 3, 8

[74] G. Zhang, J. Jia, T. Wong, and H. Bao. Consistent depth maps recovery from a video sequence. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(6):974–988, 2009. 33, 34, 35