NATIONAL UNIVERSITY OF SINGAPORE

Exploiting Structural Constraints in Image Pairs

by

Lin Wen Yan, Daniel

A thesis submitted in partial fulfillment for
a PhD dgree in Engineering

in the
Faculty of Engineering
Department of Electrical and Computer Engineering

August 2011

NATIONAL UNIVERSITY OF SINGAPORE

Summary

Two images of a scene can provide the 3-dimensional structural information that is absent in a single 2-D image. This is because, provided correspondence can be established across the two views, the variations between the two images provide cues related to the depth ordering of objects in the scene. These cues can be exploited for applications such as 3-D reconstruction, mosaicing and computation of relative camera positions. While these applications are dependent upon the quality of the inter-image correspondence, with the anticipated correspondence noise having a significant impact on the problem formulation, many of these applications can also facilitate the correspondence computation. In this thesis, we explore the interlocking relationship between image correspondence and computation and utilization of structural cues using a series of case studies. In chapter 2, we show how studying the small motion problem with an explicit focus on the types of correspondence noise anticipated, allows for a theoretical fusion of the discrete and differential algorithms. In chapter 3, we consider how to design a structure from motion algorithm which can utilize edge information. In contrast with most existing algorithms, we do not simply use corner or line features. Rather, we incorporate edge (without making a straight line assumption) information with a smoothing term to enable computation of structure from motion from scenes which are dominated by strong edge information but lacking in corner features. Finally, in chapter 4, we use an algorithm similar to that in chapter 3, to enable

the computation of inter-image mosaicing on image pairs with parallax, without the need to explicitly compute structure from motion.

Acknowledgements

I would like to take this opportunity to thank the many people who have worked with me and helped in the formulation and shaping of the ideas presented here. First in line is my supervisor Dr Cheong Loong Fah and his wife Dr Tan Geok Choo. I must also thank our DSO collaborates Dr Guo Dong and Dr Yan Chye Hwang. I am also grateful to my lab mates Liu Sying and Hiew Litt Teen for sharing their knowledge freely as well as our superb lab officer Francis Hoon. Special thanks must go to Dr Tan Ping for freely rendering his invaluable advice.

Contents

Summary **i**

Acknowledgements **iii**

1 Introduction **1**

1.1 Structure from Motion 2

1.2 Mosaicing 3

1.3 Other issues 4

2 Discrete meets Differential in SfM **5**

2.1 Motivation 5

2.1.1 The Differential Formulation 6

2.1.2 Noise and Perturbation Analysis 8

2.1.3 Findings and Organization 11

2.1.4 Mathematical Notations 13

2.1.5 Mathematical Expressions 16

2.2 A Single Moving Camera Viewing a Stationary Scene 17

2.2.1 Epipolar Constraint with Normalization 19

2.3 The Degeneracy Affecting the Discrete Algorithm 23

2.3.1 The Null Space of $A_R^T A_R$ 24

2.4 On the Noiseless Case $A(\epsilon)^T A(\epsilon)$ 28

2.4.1 How the Eigenvectors of $A^T(\epsilon)A(\epsilon)$ Vary with ϵ 29

2.4.2 How the Eigenvalues of $A^T(\epsilon)A(\epsilon)$ Vary with ϵ 31

2.5 Eigenvalues of $A^T(\epsilon)A(\epsilon)$ under Noise 34

2.5.1 Eigenvalue $\widetilde{\lambda}_9(\epsilon)$ 35

2.6 Projection of $\widetilde{\mathbf{q}}_9(\epsilon)$ along $\mathbf{q}_k(\epsilon)$ 42

2.7 Obtaining the Rotation and Translation Parameters 51

2.7.1 Some Preliminaries 52

2.7.2 Splitting the Fundamental Matrix 54

2.7.3 Errors in the Motion Estimates 56

2.8 Simulation Results 58

2.8.1 Decreasing Baseline 58

2.8.2 Increasing Noise 60

2.8.3 Observations 60

2.9 Results on Real Image Sequences 63

2.10 Concluding remarks 66

3 Simultaneous Camera Pose and Correspondence Estimation with Motion Coherence 68

3.1 Introduction 69

3.1.1 Related works 72

3.2 Formulation 75

3.2.1 Definitions 75

3.2.2 Problem formulation 76

3.2.3 Coherence term 78

3.2.4 Epipolar term 81

3.2.5 Registration term and overall cost function 82

3.3 Joint estimation of correspondence and pose 83

3.3.1 Updating registration, $\mathfrak{B}$ 84

3.3.2 Updating camera pose, $\mathbf{F}$ 87

3.3.3 Initialization and iteration 88

3.4 System implementation 88

3.5 Experiments and Evaluation 91

3.5.1 Evaluation 93

3.5.2 Performance with increasing baseline 98

3.5.3 Unresolved issues and Discussion 100

3.6 Concluding remarks 101

4 Mosaicing **103**

4.1 Motivation 103

4.1.1 Related Work 108

4.2 Our Approach 111

4.2.1 Minimization 115

4.3 Implementation 117

4.4 Analysis 118

4.5 Applications 119

4.5.1 Re-shoot 120

4.5.2 Panoramic stitching 123

4.5.3 Matching 124

4.6 Concluding remarks 124

5 Conclusions and Future Work **128**

A Proofs related to Chapter 2 **131**

A.1 Perturbation of Eigenvalues and Eigenvectors 131

A.2 Errors in the Translation Vector and Rotation Matrix 135

B Proofs related to Chapter 3 **141**

C Proofs related to Chapter 4 **145**

C.1 Minimization of Smoothly varying Affine field 145

C.2 Affine Smoothness 148

Bibliography **152**

Chapter 1

Introduction

An image is a 2-D projection of a 3-D world. The loss of one dimension means that the appearance of images of the same scene change with view point, a reflection of the scenes depth variation, a phenomenon known as parallax. It is possible to utilize these differences to recover 3-D structure and relative camera orientation. One can also take the opposite approach and compensate for the differences caused by variation in view point and structure to integrate the image pair into a mosaic.

Utilizing two views of a scene requires the establishment of accurate correspondence across the image pairs, a non-trivial problem. The anticipated correspondence noise has a significant impact on the way applications utilizing image pairs are formulated. This relationship is made more complex because many of the applications, such as camera pose recovery, can also facilitate correspondence computation. In this thesis, we investigate the interlocking relationship between correspondence computation and high level image pair applications.

1.1 Structure from Motion

Structure from Motion or SfM is the process of obtaining of 3-D structure from multiple images of the same scene and has a long and rich history in computer vision. While there are many different SfM algorithms, they all share some common modules. Typically, correspondence is first established across images. This is followed a computation of relative camera orientation and finally a dense reconstruction to recover the full 3-D model.

As a means of recovering 3-D models, SfM's key advantage lies in it adaptability. Since it requires only image data as an input, it is significantly more flexible than alternative techniques such as 3-D laser scanning, which need bulky and expensive equipment. In addition, SfM techniques are readily scalable and the same algorithm used to reconstruct a city can be applied without modification to reconstruct a small toy. This degree of flexibility makes SfM important for many other vision based applications such as navigation, recognition, 3-D movies etc. Further, SfM also acts as a form of data compression, in which the information in a large collection of images is summarized within a single compact model, thus summarizing the information contained in multiple images into a form that is easily accessible to the viewer. The primary drawback of SfM is that the algorithm remains fragile and more work is needed to increase the quality of its results. This desire for increased stability is a major theme in this thesis.

Typically, SfM algorithms are divided into large motion and small motion algorithms. This is because structure from motion as its name implies, is dependent

upon motion and non-motion is a degenerate case. This makes the SfM problem very ill conditioned if the motion was small, possibly infinitesimally small. To overcome this problem, researchers have reformulated the small motion problem as a structure from velocity problem. In this thesis, we show that for the two view scenario, if one considers the relationship between noise and displacement, the linear structure from velocity problem is the same as the discrete structure from motion problem. This work was published in [57].

While SfM involves computing relative camera position (pose), known camera pose can also facilitate correspondence. This is because given camera pose, we can define an epipolar line which narrows the correspondence search space from a 2-dimensions into a single dimension line search problem. In this thesis, we show that by jointly estimating both correspondence and camera pose, we can utilize non-unique features like edges to facilitate camera pose recovery. These edge features are difficult to correspond in a point to point fashion and are usually not incorporated into traditional camera pose recovery modules. This work was published in [58].

1.2 Mosaicing

Mosaicing is the process of integrating multiple images into a single, novel picture. This is allows us to fuse aspects from different images and is frequently used to create large field of view mosaics. Traditionally, mosaicing is performed between

parallax free images (such as images of a planar scene or images taken from a camera executing pure rotations). In this thesis, we formulate a mosaicing algorithm which can handle parallax.

Unlike in SfM, our mosaicing algorithm does not complete a full structure recovery process to utilize depth information, thus avoiding some of SfM algorithms fragility in common mosaicing scenarios. Rather, our formulation uses a smoothly varying affine field to make implicit to achieve mosaicing by making implicit use of the underlying structure.

While this application differs somewhat from the previous two, the underlying design considerations are similar, with our designing a joint mosaicing and correspondence computation algorithm so as to leverage on the interlocking nature of both problems. This helps reduce the problem of outlier matches and permits more and better correspondence, which in turn improves the mosaic.

1.3 Other issues

The interlocking issues of correspondence noise, correspondence quality and global parameter estimation involve a large number of different fields and applications. In this thesis, we concentrate primarily on static scenes, though our mosaicing work may also facilitate independent motion detection. We feel that this is a promising research area and we look forward to greater improvements.

Chapter 2

Discrete meets Differential in SfM

Differential Structure from Motion problems are velocity based formulations. In this chapter, we note that a velocity based formulation is similar to a discrete formulation, assuming a proportional noise model (the noise incurred in the correspondence is proportional to the amount of motion). If one makes this assumption explicit and investigate the discrete Structure from Motion formulation through the lenes of matrix perturbation theory, it appears that discrete SfM can handle small motion in a manner similar to differential formulations.

2.1 Motivation

Differential algorithms have been employed in SfM for many years. They are formulated for situations in which the motion is very small, such that the said motion can be approximated by velocity. To date, for nearly every discrete SfM algorithm, such as the seminal eight point algorithm by Longuet-Higgins [61], there exists a

differential counterpart (such as [62]). Although there is work reporting simulation results which indicate that some discrete SfM algorithms appear capable of handling very small motions [6, 99], the stability of the discrete formulation under small motion has largely been viewed with suspicion. Despite the many error analyses conducted on discrete SfM [18, 22, 66, 68, 72, 109], there is no work that specifically looks at the behaviour of these algorithms under increasingly smaller motions, and it is not clear what exactly is gained by resorting to a differential formulation. As such, the primary question that we seek to answer is whether differential algorithms are merely a simplification of the SfM problem, made possible by making a small motion approximation, or if the formulations address fundamental degeneracies in the SfM problem caused by small motion which cannot be handled by discrete algorithms.

If the answer is the former, it calls into question the motivation for a large volume of SfM literature which by and large treat the differential problem as something distinct from the discrete one. Examples of differential SfM formulations include [11, 30, 41, 44, 49, 62, 67, 105]. If the answer is the latter, it gives rise to the question of whether a proper understanding of the role of differential algorithms will allow us to design better discrete algorithms which can perform the same task without resorting to a differential approximation.

2.1.1 The Differential Formulation

Let us begin by considering the motivation underlying differential algorithms. As the name structure from motion suggests, one degenerate scenario common to all

SfM algorithms is that of a stationary camera. This degeneracy is intrinsically insurmountable (if there is no motion, there will surely be no structure from motion). However, it brings to mind a set of very interesting questions. How large must the motion be before we can recover structure? Is it possible to recover structure from an infinitesimally small motion? If so, what are the conditions required for a reasonable structure recovery?

Differential SfM algorithms provide a very elegant answer to all of the above questions. They assert that when the motion is small, the movement of the individual feature points on the image plane can be approximated as 2D image velocity (which is in turn approximated by optical flow). After estimating the 2D optical flow, the differential algorithms seeks to compute the differential quantities defining the cameras motion (angular velocity and translation direction) and following that, the scene structure. As these algorithms are formulated in terms of the instantaneous motion, a quantity that is independent of the amount moved, it is clear that provided the image feature velocity (or optical flow) can be extracted reasonably well, the stability of the algorithm is not affected by issues of whether or not a motion is "too small".

The need to extract a reasonable estimate of the instantaneous feature velocity for arbitrarily small motion in turn requires that the ratio of noise to optical flow magnitude (percentage noise) must be sufficiently small. In essence, the underlying premises of the differential formulation is that one can recover structure and motion from a sufficiently small motion, provided one has a reasonable bound on the percentage noise in the optical flow.

In seeking to ascertain if the differential formulation avoids an intrinsic degeneracy present in the discrete formulation we need to consider whether the associated discrete algorithm will yield a reasonable estimate for structure and motion given a sufficiently small motion and a reasonable bound on the percentage noise. Henceforth, we denote algorithms that demonstrate such behavior as being able to handle "differential conditions". We would also like to distinguish between the inherent sensitivity of the underlying problem and the error properties of a particular algorithm for solving that problem. For instance, trying to solve the SfM problem for a configuration near to the critical surface [72, 79] is an inherently sensitive problem. No algorithms (discrete or differential) working with finite arithmetic precision can be expected to obtain a solution that is not contaminated with large errors. In this chapter, we are primarily interested in the stability of the discrete SfM algorithms under small motion, in the sense that it does not produce any more sensitivity to perturbation than is inherent in the underlying problem. Thus we would only deal with general scenes not close to an inherently ambiguous configuration.

2.1.2 Noise and Perturbation Analysis

We feel that a major reason for the persistent division of the two view problem into the differential and discrete domain is because it is very difficult to systematically analyze the performance of discrete algorithms when the motion is small.

Some intuition into this problem can be obtained by looking at the classical discrete eight point algorithm, where the essential matrix is obtained as the solution to the least squares problem $\min\|Ax\|^2$. Since the solution is in the null space of the

symmetric matrix A^TA, the sensitivity of the problem can be characterized by how the eigenvalues and eigenvectors of the data matrix A^TA is influenced by the amount of motion and noise. As we show later, under small motion, the data matrix can be written as:

$$A(\epsilon) \approx A_R + \epsilon A_T$$

where the data matrix $A(\epsilon)$ is now written as a function of ϵ. $A(\epsilon)$ is split into two terms: the residue term A_R when there is no motion, and the motion term ϵA_T, with $\epsilon \to 0$ as the amount of motion becomes progressively smaller. As we will show later, the rank of the matrix A_R is at most 6, and in fact, for a general scene, the rank of A_R is exactly 6. Since A_R has right nullity greater than 1, as ϵ approaches 0 and $A(\epsilon)$ approaches A_R, the problem of finding a unique solution to the right null space of $A^T(\epsilon)A(\epsilon)$ becomes increasingly ill-conditioned as the gaps between the eigenvalues become smaller. In particular, if we assume that a small fixed noise N exists in the estimation process (e.g. noise arising from finite arithmetic precision, which is 16 decimal digits for double precision):

$$\tilde{A}(\epsilon) \approx A_R + \epsilon A_T + N$$

then at a small enough motion, the noise N becomes comparable or even exceeds the motion term ϵA_T such that the legitimate solution is no longer associated with the smallest eigenvalue of $\tilde{A}^T(\epsilon)\tilde{A}(\epsilon)$. This sudden appearance of a second solution has been termed as the second eigenmotion in [68]. This ill-conditioning is the primary reason why vision researchers have reservations over applying the discrete formulation when faced with the problem of small motion.

However, before reaching the limit of arithmetic precision, the noise is likely to be dictated by measurement noise in the feature correspondence or the optical flow, and this noise is likely to obey a proportional model. In small motion, the correspondence problem is much simplified by the fact that the two views of the scene do not differ greatly from each other. There will be less hidden surfaces, smaller difference in radiometry, and less geometrical deformation. Hence, although the motion of individual feature points is small, the absolute error incurred in the matching process is also small. For really small motion, differential optical flow algorithms [43, 65] would be better placed to yield the desired measurement accuracy, especially with some of the more sophisticated recent implementations [13, 42, 55, 80, 83, 91, 89]. The error in estimating image velocity through the Brightness Constancy Equation (BCE) has been analyzed by [104] from which it is clear that the noise is also likely to be proportional to the magnitude of the motion. It was shown that error stems from various sources, such as changes in the lighting arising from non-uniform illumination or different point of view, or abrupt changes in the reflectance properties of the moving surfaces at the corresponding location in space, all of which are proportional to the magnitude of the motion. Ohta's analysis [81] approached from the perspective of the electronic noise in the imaging devices and also showed the same dependence of the measurement error in the optical flow on the amount of image motion. This is a consequence of the finite receptive field in real cameras, whereby the sampling function is not a Dirac's delta function but rather depends on both the image gradient and the image motion.

In this connection, it is also well to note that many algorithms for finding optical flow make errors not only due to the aforementioned sources, but also due to violation of the flow distribution model that is assumed (such as the smoothness assumption). This latter source of error might give rise to non-proportional noise and thus prevent us from obtaining structure from truly infinitesimally small motions, even if we have succeeded in proving the stability of the discrete eight point algorithm under small motion with proportional noise. However, we envisage that these algorithm-specific errors arising from flow distribution model would become smaller and smaller, especially with the recent spate of optical flow algorithms [13, 42, 55, 80, 83, 91, 89] and together with the publication of a database for optical flow evaluation [3]. Indeed, with better flows computed from these algorithms in regular usage, there is greater motivation for using flows to recover scene structure since it avoids having to solve the tricky problem of feature correspondence. It then begs the question whether we should recover structure from flow using one of the differential SfM formulations, or if inputting flows to some of the discrete normalized variants offer a better alternative.

2.1.3 Findings and Organization

In this chapter, we carry out perturbation analysis to study the numerical stability of the discrete eight point algorithm and its variants [19, 39, 61, 74, 97] under small motion. The noise regime that we have adopted is such that the data matrix $\tilde{A}(\epsilon)$ is given by

$$\tilde{A}(\epsilon) \approx A_R + \epsilon A_T + M(\epsilon)$$

where $M(\epsilon)$ represents the inherent measurement errors arising from various sources such as the BCE constraint and the electronic noise, both of which are proportional to the amount of motion "ϵ". We show that given a sufficiently small proportional noise $M(\epsilon)$, the discrete eight point algorithm and its variants are all capable of handling "differential conditions". For researchers who view the differential/discrete dichotomy as inviolate, this result is significant because much effort has been spent in refining the discrete eight point algorithm. It permits us to use the more intensively researched discrete algorithms without first reformulating the problem as a differential one; this can result in very large improvements over the current state-of-the-art differential algorithms. As we show later in the experimental section, the normalized discrete algorithms appear to give considerably better performance than its differential counterparts even when the motion is extremely small. For researchers who believe that discrete algorithms can be readily applied to the small motion problem, this chapter provides some explanation for their empirical results and illustrates the limits within which such an attitude may be adopted.

The theoretical portion of the chapter is primarily divided into three large portions. The first third of the chapter (Sections 2.2 to 2.4) involves introducing the eight point formulation, with some minor reformulations to allow rigorous analysis of its supposed ill-conditioning in the context of small motion. The second third (Sections 2.5 to 2.6) is primarily an adaptation of traditional perturbation theory to our problem of relating baseline to noise, one of the differences being that our data matrix $A(\epsilon)$ is also a function of ϵ. Finally, in the last third of the chapter

(Section 2.7), we complete the stability analysis by tracking how the errors in the fundamental matrix estimate are propagated to the rotation and translation estimates, from which structure of the scene is finally recovered. The theoretical analysis is then followed by the experiments and the conclusion. Lastly, we also record in the appendix some theorems and results required for the perturbation analysis carried out in the chapter proper.

2.1.4 Mathematical Notations

In this section, we explain some of the mathematical notations that a reader will frequently encounter when reading the thesis.

1. A^S **symbol**

   Let $A = \begin{bmatrix} a & d & g \\ b & e & h \\ c & f & i \end{bmatrix}$. The symbol A^S denotes the vector obtained by stacking the columns of A, i.e.,

   $$A^S = \begin{bmatrix} a & b & c & d & e & f & g & h & i \end{bmatrix}^T.$$

2. $\overbrace{\mathbf{w}}$ **symbol**

Let $\mathbf{w} = \begin{bmatrix} w_1 & w_2 & w_3 & \dots & w_9 \end{bmatrix}^T \in \mathbb{R}^9$. We denote by $\widehat{\mathbf{w}}$ the following 3×3 matrix

$$\widehat{\mathbf{w}} = \begin{bmatrix} w_1 & w_4 & w_7 \\ w_2 & w_5 & w_8 \\ w_3 & w_6 & w_9 \end{bmatrix}.$$

Clearly, we have $\left(\widehat{\mathbf{w}}\right)^S = \mathbf{w}$ and $\widehat{A^S} = A$.

3. **$\widehat{\mathbf{u}}$ symbol**

For each $\mathbf{u} = \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix}^T \in \mathbb{R}^3$, we form the 3×3 skew-symmetric matrix $\widehat{\mathbf{u}} = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix}$.

(a) For $\mathbf{v} \in \mathbb{R}^3$, we have

$$\widehat{\mathbf{u}}\mathbf{v} = \mathbf{u} \times \mathbf{v}, \tag{2.1}$$

where $\mathbf{u} \times \mathbf{v}$ is the vector product of $\mathbf{u}$ and $\mathbf{v}$.

(b) For a 3×3 invertible matrix A, with $\det(A) \neq 0$, we have the following result from page 456 of [69].

$$(A^{-1})^T \widehat{\mathbf{u}} A^{-1} = \frac{1}{\det(A)} (\widehat{A\mathbf{u}}) \tag{2.2}$$

4. Throughout this thesis, we work on the Frobenius norm of a matrix (say C) which is defined and denoted as follows:

$$\|C\| = \sqrt{\sum_{i,j} c_{ij}^2}.$$

It generalizes the definition of the usual norm on vectors.

5. Δx **symbol**

Suppose the function $x(\epsilon)$ is defined for $\epsilon \geq 0$. We shall use the usual notation Δx to denote the change in x:

$$\Delta x = x(\epsilon) - x(0).$$

Likewise, we have ΔY_i, etc. Sometimes, to avoid cumbersome notation, we denote a function $x(\epsilon)$ at $\epsilon = 0$ by just x.

6. For the ease of reading this thesis, we gather in the following a table of symbols for the eigenvalues and eigenvectors of the matrices $A_R^T A_R$, $A^T(\epsilon)A(\epsilon)$ and $\tilde{A}(\epsilon)^T\tilde{A}(\epsilon)$ (to be introduced in subsequent sections).

Matrix	Eigenvalues	Eigenvectors
$A_R^T A_R$	λ_i	$\mathbf{r}_i$, unit vector
$A^T(\epsilon)A(\epsilon)$	$\lambda_i(\epsilon)$	$\mathbf{q}_i(\epsilon)$, unit vector
$\tilde{A}(\epsilon)^T\tilde{A}(\epsilon)$	$\tilde{\lambda}_i(\epsilon)$	$\tilde{\mathbf{q}}_i(\epsilon)$

2.1.5 Mathematical Expressions

The following phrases will be frequently encountered in this thesis.

1. **For a sufficiently small ϵ:** If we say that a condition (or a statement) X is satisfied for a sufficiently small ϵ, it means that there exists a positive $\epsilon_0 > 0$, such that the condition (or statement) X is satisfied for all ϵ where $0 \leq \epsilon < \epsilon_0$.

2. **Order ϵ^n or $O(\epsilon^n)$:** For an integer n, a function $f(\epsilon)$ is said to be of order ϵ^n if $|f(\epsilon)| \leq K\epsilon^n$ for some $K > 0$ as $\epsilon \to 0$. That is, for a sufficiently small $\epsilon > 0$, $\left|\frac{f(\epsilon)}{\epsilon^n}\right|$ is uniformly bounded. In symbol, we write $f(\epsilon) = O(\epsilon^n)$. When $n = 0$, we write $O(\epsilon^0)$.

   Some special cases/notes:

   (a) For a function $f(\epsilon)$, we note that

   $$f(\epsilon) = O(\epsilon^{n+1}) \Rightarrow f(\epsilon) = O(\epsilon^n),$$

   but the converse is not true in general. In other words, $O(\cdot)$ is not an asymptotically tight bound.

   (b) If $F(\epsilon)$ is a matrix (or a vector) , then saying it is of order ϵ^n means that each of its individual entries is of order ϵ^n. This is equivalent to saying that the norm of $F(\epsilon)$ (which is a real valued function) is of order ϵ^n.

(c) Let k be a rational number. For a sufficiently small η, we have

$$(1+\eta)^k = 1 + k\eta + O(\eta^2).$$

This follows from the first order term of the Taylor expansion. In particular, for non-negative real numbers n and l, and sufficiently small ϵ and m, we have

$$(1 + O(\epsilon^n)m^l)^k = 1 + O(\epsilon^n)m^l, \tag{2.3}$$

where the constant k has been absorbed in the O-notation.

2.2 A Single Moving Camera Viewing a Stationary Scene

Let us assume that there is a single moving camera viewing a stationary scene consisting of N feature points $\mathbf{P}_i$, where $1 \leq i \leq N$.

Let $\epsilon \geq 0$ be a non-negative real number representing the elapsed time. Our goal in this section is to formulate the eight point algorithm in the form of a data matrix and a solution vector, both of which can be expressed as a series in ϵ. Subsequently, we will use matrix perturbation theory to analyze their properties when the elapsed time ϵ (and hence the motion) is small.

At time instance $\epsilon \geq 0$, a point $\mathbf{P}_i$ has its coordinates with respect to the camera reference frame given by

$$\mathbf{P}_i(\epsilon) = \begin{bmatrix} X_i(\epsilon) & Y_i(\epsilon) & Z_i(\epsilon) \end{bmatrix}^T.$$

Let us assume that the motion is smooth, with the camera positions being related to each other by the translation vector $\epsilon\mathbf{T}_c$ and a smoothly changing rotation $R(\epsilon)$. The 3×1 vector $\mathbf{T}_c$ is a constant vector representing the translational velocity, whereas the 3×3 matrix $R(\epsilon)$ is a rotation matrix which changes smoothly with ϵ and $R(0) = I$, where I is the 3×3 identity matrix.

The rotation matrix $R(\epsilon)$ can be expressed as the exponential of some skew-symmetric matrix $\widehat{\boldsymbol{\omega}}$, that is, a series of the form (Theorem 2.8, [69])

$$R(\epsilon) = I + \epsilon\widehat{\boldsymbol{\omega}} + O(\epsilon^2), \tag{2.4}$$

where $\boldsymbol{\omega}$ is the angular velocity.

As a result of the motion, we have

$$\mathbf{P}_i(\epsilon) = R(\epsilon)(\mathbf{P}_i - \epsilon\mathbf{T}_c). \tag{2.5}$$

Recall from the preceding section that sometimes we shall denote $\mathbf{P}_i(0) = \mathbf{P}_i$. When projected onto the image plane of the camera, the points $\mathbf{P}_i$ and $\mathbf{P}_i(\epsilon)$ will

have image coordinates $\mathbf{p}_i$ and $\mathbf{p}_i(\epsilon)$ respectively where

$$\begin{aligned} \mathbf{p}_i &= \frac{1}{Z_i}\mathbf{P}_i = \begin{bmatrix} x_i & y_i & 1 \end{bmatrix}^T, \\ \mathbf{p}_i(\epsilon) &= \frac{1}{Z_i(\epsilon)}\mathbf{P}_i(\epsilon) = \begin{bmatrix} x_i(\epsilon) & y_i(\epsilon) & 1 \end{bmatrix}^T. \end{aligned} \tag{2.6}$$

Using Equations (3.1) and (2.6), we have

$$\mathbf{p}_i(\epsilon) = \mathbf{p}_i + \epsilon[\ \Delta_t x \qquad \Delta_t y \qquad 0\ ]^T + O(\epsilon^2) \tag{2.7}$$

where $\Delta_t x$, $\Delta_t y$ are the x and y components of the image feature velocity respectively.

2.2.1 Epipolar Constraint with Normalization

Given two camera images, one at time 0 and the other at time ϵ, the epipolar constraint is

$$\mathbf{p}_i^T E(\epsilon)\mathbf{p}_i(\epsilon) = 0, \tag{2.8}$$

where $E(\epsilon) = \widehat{\mathbf{T}_c} R^T(\epsilon)$.

Given eight or more point matches, the above epipolar constraint is sufficient for us to determine the essential matrix $E(\epsilon)$ up to a scale factor, by solving a set of linear equations.

This is the famous eight point algorithm of [61]. However, it is important to note that the epipolar constraint is seldom used in its raw form. Rather, for the sake

of numerical stability in the presence of noise, a normalization procedure is often employed.

Let Θ be an 3×3 invertible matrix introduced for this purpose. For example, it can be of the form $\begin{bmatrix} a & 0 & c \\ 0 & b & d \\ 0 & 0 & 1 \end{bmatrix}$, with $ab \neq 0$. Examples of normalization matrices taking such form are the normalization matrix in Hartley normalization [39], or in the context of uncalibrated motion analysis, Θ would be the camera's intrinsic matrix.

For a sufficiently small $\epsilon \geq 0$, suppose

$$\Theta(\epsilon) = \Theta + O(\epsilon) \tag{2.9}$$

is invertible. Then its inverse $(\Theta(\epsilon))^{-1}$ takes the form

$$(\Theta(\epsilon))^{-1} = \Theta^{-1} + O(\epsilon). \tag{2.10}$$

We denote the normalized (or uncalibrated) version of the essential matrix as $F(\epsilon)$ where

$$F(\epsilon) = (\Theta^T)^{-1} E(\epsilon) (\Theta(\epsilon))^{-1}. \tag{2.11}$$

We will sometimes call $F(\epsilon)$ the fundamental matrix where appropriate.

Using Equations (2.9) and (2.7), we can write

$$\Theta(\epsilon)\mathbf{p}_i(\epsilon) = [\ \underline{x_i} \qquad \underline{y_i} \qquad 1\ ]^T + \epsilon[\ \underline{\Delta_t x} \qquad \underline{\Delta_t y} \qquad 0]\ + O(\epsilon)^2, \tag{2.12}$$

where

$$\Theta\mathbf{p}_i = [\ \underline{x_i} \qquad \underline{y_i} \qquad 1\ ]^T,$$

$$\Theta(\epsilon)\mathbf{p}_i(\epsilon) = [\ \underline{x_i}(\epsilon) \qquad \underline{y_i}(\epsilon) \qquad 1\ ]^T$$

and $(\underline{\Delta_t x}, \underline{\Delta_t y})$ is the image feature velocity in the normalized system. In this normalized system, the corresponding epipolar constraint (3.6) becomes

$$[(\Theta\mathbf{p}_i)]^T F(\epsilon)[(\Theta(\epsilon))\mathbf{p}_i(\epsilon)] = 0. \tag{2.13}$$

Collecting N such constraints for $i = 1, \ldots, N$, we form a system of linear equations:

$$A(\epsilon)\,(F(\epsilon))^S = 0, \tag{2.14}$$

where

$$A(\epsilon) = \begin{bmatrix} \underline{x_1(\epsilon)x_1} & \underline{x_1(\epsilon)y_1} & \underline{x_1(\epsilon)} & \underline{y_1(\epsilon)x_1} & \underline{y_1(\epsilon)y_1} & \underline{y_1(\epsilon)} & \underline{x_1} & \underline{y_1} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \underline{x_N(\epsilon)x_N} & \underline{x_N(\epsilon)y_N} & \underline{x_N(\epsilon)} & \underline{y_N(\epsilon)x_N} & \underline{y_N(\epsilon)y_N} & \underline{y_N(\epsilon)} & \underline{x_N} & \underline{y_N} & 1 \end{bmatrix} \tag{2.15}$$

and $(F(\epsilon))^S$ is the column vector defined in Section 2.1.4. Thus, an estimate of the matrix $F(\epsilon)$ can be obtained via the null space of $A(\epsilon)$.

Finally, we rewrite the matrix $F(\epsilon)$ in Equation (2.11) into a form more amenable to analysis:

$$\begin{aligned} F(\epsilon) &= (\Theta^T)^{-1} E(\epsilon)(\Theta(\epsilon))^{-1} \\ &= (\Theta^{-1})^T \widehat{\mathbf{T}_c} \Theta^{-1} \Theta R^T(\epsilon)(\Theta(\epsilon))^{-1} \\ &= \frac{1}{\det(\Theta)} [\widehat{(\Theta \mathbf{T}_c)}][\Theta R^T(\epsilon)(\Theta(\epsilon))^{-1}] \end{aligned} \tag{2.16}$$

where the last step has been obtained by using Equation (2.2). By Equations (2.10) and (2.4), we have $\Theta R^T(\epsilon)(\Theta(\epsilon))^{-1} = I + O(\epsilon)$ which gives

$$F(\epsilon) = \frac{1}{\det(\Theta)} [\widehat{(\Theta \mathbf{T}_c)}][I + O(\epsilon)]. \tag{2.17}$$

Since $F(\epsilon)$ is defined up to a scale factor, we can write

$$F(\epsilon) = \widehat{\mathbf{T}} + O(\epsilon) \tag{2.18}$$

where

$$\mathbf{T} = \frac{\Theta \mathbf{T}_c}{\sqrt{2} \|\Theta \mathbf{T}_c\|}. \tag{2.19}$$

Here, we have set $\|\mathbf{T}\| = \frac{1}{\sqrt{2}}$ so that $(\widehat{\mathbf{T}})^S$ has unit norm. From Equations (2.16) and (2.1), we note that $\mathbf{T}$ is in the left null space of $F(\epsilon)$.

2.3 The Degeneracy Affecting the Discrete Algorithm

Using Equation (2.12), we rewrite the data matrix $A(\epsilon)$ as a series expansion in ϵ,

$$A(\epsilon) = A_R + \epsilon A_T + O(\epsilon^2) \tag{2.20}$$

where

$$A_R = \begin{bmatrix} \underline{x_1^2} & \underline{x_1 y_1} & \underline{x_1} & \underline{y_1 x_1} & \underline{y_1^2} & \underline{y_1} & \underline{x_1} & \underline{y_1} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \underline{x_N^2} & \underline{x_N y_N} & \underline{x_N} & \underline{y_N x_N} & \underline{y_N^2} & \underline{y_N} & \underline{x_N} & \underline{y_N} & 1 \end{bmatrix},$$

$$A_T = \begin{bmatrix} \underline{x_1 \Delta_t x_1} & \underline{y_1 \Delta_t x_1} & \underline{\Delta_t x_1} & \underline{x_1 \Delta_t y_1} & \underline{y_1 \Delta_t y_1} & \underline{\Delta_t y_1} & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \underline{x_N \Delta_t x_N} & \underline{y_N \Delta_t x_N} & \underline{\Delta_t x_N} & \underline{x_N \Delta_t y_N} & \underline{y_N \Delta_t y_N} & \underline{\Delta_t y_N} & 0 & 0 & 0 \end{bmatrix}. \tag{2.21}$$

Recall that when the motion (i.e., ϵ) is small, the discrete eight point algorithm is regarded as increasingly ill conditioned. In this section, we revisit the explanation in terms of the data matrix $A(\epsilon)$.

As ϵ tends to zero, using Equation (2.20), we know that $A(\epsilon)$ tends to A_R. Let F_0 be a 3×3 matrix satisfying

$$(\Theta \mathbf{p}_i)^T F_0 \left(\Theta(0)\mathbf{p}_i(0)\right) = 0$$

$$\text{i.e., } (\Theta \mathbf{p}_i)^T F_0 (\Theta \mathbf{p}_i) = 0, \tag{2.22}$$

which is the constraint given in Equation (2.13) when $\epsilon = 0$. Solving F_0 from Equation (2.22) is equivalent to solving the following linear least squares system

$$A_R (F_0)^s = 0.$$

whose solution space we will analyze now.

2.3.1 The Null Space of $A_R^T A_R$

In this subsection, we prove that for a general scene, the nullity of the 9×9 matrix $A_R^T A_R$ is 3 and we also determine the null space of $A_R^T A_R$.

Assume that the feature points on the image plane are well distributed such that we cannot fit a conic section that passes through all of them (this condition is easily satisfied, especially under small motion where the number of features which can be matched is very dense). We then have the following result.

Proposition 2.1. *Assume that all the feature points on the image plane do not lie on any conic section. The nullity of* $A_R^T A_R$ *is* 3.

Proof. Since

$$A_R^T A_R \mathbf{u} = \mathbf{0} \Leftrightarrow A_R \mathbf{u} = \mathbf{0},$$

we shall determine the nullity of A_R.

Note that the matrix A_R in Equation (2.21) contains 3 pairs of identical columns, namely columns 2 and 4, columns 3 and 7, and columns 6 and 8. Hence, the rank of A_R is at most 6.

Consider the submatrix A'_R formed from A_R by removing one copy of each repeating column pair:

$$A'_R = \begin{bmatrix} \underline{x_1^2} & \underline{x_1 y_1} & \underline{x_1} & \underline{y_1^2} & \underline{y_1} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \underline{x_N^2} & \underline{x_N y_N} & \underline{x_N} & \underline{y_N^2} & \underline{y_N} & 1 \end{bmatrix}.$$

We shall show that the 6 columns of A'_R are linearly independent so that the rank of A_R is at least 6. Suppose $A'_R\mathbf{v} = 0$, where $\mathbf{v} = \begin{bmatrix} a & b & c & d & e & f \end{bmatrix}^T \neq \mathbf{0}$. This gives,

$$a\underline{x_i^2} + b\underline{x_i y_i} + c\underline{x_i} + d\underline{y_i^2} + e\underline{y_i} + f = 0, 1 \leq i \leq N,$$

which means that all the feature points lie on the conic defined by $ax^2 + bxy + cx + dy^2 + ey + f = 0$. This violates our assumption. So, we must have $\mathbf{v} = \mathbf{0}$, which implies that the rank of A'_R (and hence A_R) is at least 6. Therefore, the rank of A_R is 6. By the rank-nullity formula, the nullity of A_R is given by $(9 - (\text{ rank of } A_R))$. Hence the nullity of A_R is 3, and so is that of $A_R^T A_R$. □

Proposition 2.2. *The null space of $A_R^T A_R$ is the following set*

$$S = \left\{ (\widehat{\mathbf{u}})^S \mid \mathbf{u} \in \mathbb{R}^3 \right\} = \left\{ \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix}^S \mid u_1, u_2, u_3 \in \mathbb{R} \right\}.$$

Proof. By Equation (2.1), every skew-symmetric matrix $\widehat{\mathbf{u}}$ formed from $\mathbf{u} \in \mathbb{R}^3$ will satisfy Equation (2.22). Thus, the set $S \subseteq$ null space of A_R.

However, the set S is a subspace of $\mathbb{R}^9$ and its dimension is 3. Since the nullity of A_R is also 3, the set S is indeed the null space of A_R. □

It is a well known fact that a real symmetric matrix of the form $Z^T Z$ has non-negative eigenvalues. Thus, we can arrange the 9 non-negative eigenvalues λ_i of the matrix $A_R^T A_R$ in a non-increasing order:

$$\lambda_1 \geq \lambda_2 \geq \cdots > \lambda_7 \geq \lambda_8 \geq \lambda_9 \geq 0.$$

Proposition 2.3. *Consider the real symmetric matrix $A_R^T A_R$, and let λ_i be its eigenvalue, with corresponding unit eigenvector $\mathbf{r}_i$, for $1 \leq i \leq 9$. Then we have*

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_6 > \lambda_7 = \lambda_8 = \lambda_9 = 0.$$

Moreover, we may choose $\mathbf{r}_7, \mathbf{r}_8$ and $\mathbf{r}_9$ such that

$$\mathbf{r}_7 = \widehat{\mathbf{T}_7}^S, \mathbf{r}_8 = \widehat{\mathbf{T}_8}^S, \mathbf{r}_9 = \widehat{\mathbf{T}}^S$$

where $\mathbf{T}$ is defined in (2.19)*, and $\mathbf{T}_7$, $\mathbf{T}_8$ and $\mathbf{T}$ are mutually orthogonal vectors of norm $\frac{1}{\sqrt{2}}$.*

Proof. It follows from the nullity of A_R (and hence $A_R^T A_R$) being 3 that the real symmetric matrix $A_R^T A_R$ has a zero eigenvalue, with multiplicity 3. Thus, $\lambda_7 =$

$\lambda_8 = \lambda_9 = 0$.

The eigen-space corresponding to the zero eigenvalue is indeed the null space of $A_R^T A_R$. Since the null space of $A_R^T A_R$ is spanned by its three eigenvectors, we are free to choose $\mathbf{r}_7$, $\mathbf{r}_8$ and $\mathbf{r}_9$, as long as they belong to the subspace S in Proposition 2.2, and are orthonormal to each other. Therefore we choose to set

$$\mathbf{r}_9 = \widehat{\mathbf{T}}^S, \tag{2.23}$$

where $\mathbf{T}$ is defined in Equation (2.19).

By Proposition 2.2, the other two eigenvectors $\mathbf{r}_7$ and $\mathbf{r}_8$ can also be written in the form $\mathbf{r}_7 = \widehat{\mathbf{T}_7}^S$, $\mathbf{r}_8 = \widehat{\mathbf{T}_8}^S$, where $\mathbf{T}_7$, $\mathbf{T}_8$ and $\mathbf{T}$ must be mutually orthogonal vectors of norm $\frac{1}{\sqrt{2}}$ to ensure the orthonormality of $\mathbf{r}_7$, $\mathbf{r}_8$ and $\mathbf{r}_9$. □

Since A_R has right nullity greater than 1, as ϵ approaches 0 and $A(\epsilon)$ approaches A_R, the problem of finding a unique solution to the right null space of $A^T(\epsilon)A(\epsilon)$ (recall that camera pose is estimated from the right null space of $A(\epsilon)$) becomes increasingly ill-conditioned. This ill-conditioning is the primary reason why vision researchers have reservations over applying the discrete formulation when faced with the problem of small motion. However, as we have argued in Section 1, if the noise in the flow estimation is proportional to ϵ, the question then becomes whether the noise declining proportionally to ϵ is sufficient to compensate for the increased instability due to the last three eigenvalues of $A^T(\epsilon)A(\epsilon)$ getting closer together. In the next section, we explain how this problem can be analyzed using matrix perturbation theory.

2.4 On the Noiseless Case $A(\epsilon)^T A(\epsilon)$

The least squares solution to Equation (2.14) is given by the right null space of the 9×9 symmetric matrix $A^T(\epsilon)A(\epsilon)$. As such, the subsequent analysis is conducted on $A^T(\epsilon)A(\epsilon)$ rather than $A(\epsilon)$.

If one thinks of the eigenvectors $\mathbf{q}_i(\epsilon)$ of $A^T(\epsilon)A(\epsilon)$ as possible solutions to Equation (2.14), then their corresponding eigenvalues $\lambda_i(\epsilon)$ are the residue (sum of squared error) related to these solutions. That is, we have

$$\mathbf{q}_i(\epsilon)^T A(\epsilon)^T A(\epsilon)\mathbf{q}_i(\epsilon) = \mathbf{q}_i(\epsilon)^T \lambda_i(\epsilon)\mathbf{q}_i(\epsilon) = \lambda_i(\epsilon).$$

Thus, each $\lambda_i(\epsilon)$ represents the residue of $A(\epsilon)$ associated with $\mathbf{q}_i(\epsilon)$. The larger the value of $\lambda_i(\epsilon)$, for $1 \leq i \leq 8$, the more stable is the solution as the "wrong" solution is less likely to be confused with the correct one.

In the absence of noise, using Equation (2.20) , the matrix $A^T(\epsilon)A(\epsilon)$ can be expressed as the following series expansion,

$$A^T(\epsilon)A(\epsilon) = A_R^T A_R + \epsilon(A_T^T A_R + A_R^T A_T) + O(\epsilon^2). \tag{2.24}$$

This says that the matrix $A^T(\epsilon)A(\epsilon)$ can be thought of as the "perturbation" of the matrix $A_R^T A_R$ by the matrix $\epsilon(A_T^T A_R + A_R^T A_T)$ for sufficiently small ϵ.

We shall use matrix perturbation theory to discuss the eigenvectors and eigenvalues of this "perturbed" matrix.

2.4.1 How the Eigenvectors of $A^T(\epsilon)A(\epsilon)$ Vary with ϵ

Let us denote the eigenvalues of the matrix $A^T(\epsilon)A(\epsilon)$ by $\lambda_i(\epsilon)$, $i = 1, 2, \ldots, 9$, where

$$\lambda_1(\epsilon) \geq \lambda_2(\epsilon) \geq \cdots \geq \lambda_9(\epsilon) \geq 0.$$

We shall now choose corresponding unit eigenvectors $\mathbf{q}_i(\epsilon)$, for $1 \leq i \leq 9$, such that $\{\mathbf{q}_1(\epsilon), \mathbf{q}_2(\epsilon), \ldots, \mathbf{q}_9(\epsilon)\}$ is an orthonormal basis of $\mathbb{R}^9$.

It is clear from Equation (2.16) and the definition of $A(\epsilon)$ in Equation (2.15) that the actual camera motion satisfies Equation (2.14). Thus $A(\epsilon)$ and hence $A(\epsilon)^T A(\epsilon)$ have a nullity of at least one. In other words, we have $\lambda_9(\epsilon) = 0$ and $(F(\epsilon))^S$ is the corresponding eigenvector.

It follows from both Equation (2.18) and our choice of $\mathbf{r}_9$ in Equation (2.23) that

$$(F(\epsilon))^S = \mathbf{r}_9 + O(\epsilon).$$

Normalizing $(F(\epsilon))^S$, we obtain the the unit eigenvector $\mathbf{q}_9(\epsilon) = \frac{(F(\epsilon))^S}{\|(F(\epsilon))^S\|}$ corresponding to the eigenvalue $\lambda_9(\epsilon) = 0$. By Lemma A.4 (in Appendix A.1), we have

$$\mathbf{q}_9(\epsilon) = \mathbf{r}_9 + \mathbf{z}_9(\epsilon) \text{ where } \mathbf{z}_9(\epsilon) = O(\epsilon). \tag{2.25}$$

Treating the matrix $A^T(\epsilon)A(\epsilon)$ as the "perturbation" of the matrix $A_R^T A_R$ by

$\epsilon(A_T^T A_R + A_R^T A_T)$ for sufficiently small ϵ, we apply perturbation theory (in particular, Theorem A.7 in Appendix A.1) to obtain the following result for the remaining unit eigenvectors of $A^T(\epsilon)A(\epsilon)$.

Theorem 2.4. *The set of unit eigenvectors of* $A^T(\epsilon)A(\epsilon)$ *given by*

$$\{\mathbf{q}_1(\epsilon), \mathbf{q}_2(\epsilon), \ldots, \mathbf{q}_9(\epsilon)\}$$

can be chosen such that

$$\mathbf{q}_i(\epsilon) = \mathbf{r}'_i(\epsilon) + \mathbf{z}_i(\epsilon),$$

where $\|\mathbf{r}'_i(\epsilon)\| = 1$ *and* $\mathbf{z}_i(\epsilon) = O(\epsilon)$. *Moreover, the vectors* $\mathbf{r}'_i(\epsilon)$*'s have the following properties:*

1. $\mathbf{r}'_9(\epsilon) = \mathbf{r}_9 = \widehat{\mathbf{T}}^S$ *(from Equation* (2.25)*).*
2. $\mathbf{r}'_i(\epsilon)$ *is a linear combination of all eigenvectors* $\mathbf{r}_j$ *of* $A_R^T A_R$*, whose associated eigenvalue* λ_j *is identical to* λ_i*, for* $i \leq 9$.
3. $\mathbf{r}'_i(\epsilon)$ *is orthogonal to* $\mathbf{r}_j$ *if* $\lambda_i \neq \lambda_j$.
4. $\mathbf{r}'_i(\epsilon)$, $1 \leq i \leq 8$ *is orthogonal to* $\mathbf{r}_9$.

Remark 2.5. For $7 \leq i \leq 9$, each vector $\mathbf{r}'_i(\epsilon)$ is a linear combination of $\mathbf{r}_7$, $\mathbf{r}_8$ and $\mathbf{r}_9$, and hence it is a vector in the right null space of A_R. From Proposition 2.2, we have

$$\widehat{\mathbf{r}'_7(\epsilon)} = \widehat{\mathbf{T}'_7(\epsilon)}, \text{ and } \widehat{\mathbf{r}'_8(\epsilon)} = \widehat{\mathbf{T}'_8(\epsilon)}$$

for some orthogonal vectors $\mathbf{T}'_7(\epsilon)$ and $\mathbf{T}'_8(\epsilon)$ in $\mathbb{R}^3$ where

$$\|\mathbf{T}'_7(\epsilon)\| = \|\mathbf{T}'_8(\epsilon)\| = \|\mathbf{T}\| = \frac{1}{\sqrt{2}}.$$

2.4.2 How the Eigenvalues of $A^T(\epsilon)A(\epsilon)$ Vary with ϵ

As discussed in the preceding section, $\lambda_9(\epsilon) = 0$. For the remaining eigenvalues $\lambda_i(\epsilon)$, applying perturbation theory (Theorem 6 in Appendix A.1) to the expression for $A^T(\epsilon)A(\epsilon)$ in Equation (2.24) yields the following result.

Proposition 2.6. *For* $1 \leq i \leq 8$,

$$\lambda_i(\epsilon) = \lambda_i + O(\epsilon).$$

From Proposition 2.3 we know that $\lambda_i > 0$ for $1 \leq i \leq 6$. As such, when ϵ is sufficiently small, using Proposition 2.6, we know that eigenvalues $\lambda_i(\epsilon)$ remains positive and hence their corresponding eigenvectors are distinct from the true solution.

However, for $7 \leq i \leq 8$, we note that Proposition 2.3 indicates that $\lambda_i(\epsilon)$ may be zero. From the point of view of stability, this is worrying and we must seek a more explicit expression than that offered by standard matrix perturbation theory.

Lemma 2.7. *For* $i = 7$ *or* 8, *if the hypothesis* $\|A(\epsilon)\mathbf{q}_i(\epsilon)\| = \gamma_i\epsilon + O(\epsilon^2)$ *where* $\gamma_i > 0$ *is true, then* $\lambda_i(\epsilon) = O(\epsilon^2)$. *In particular,*

$$\lambda_i(\epsilon) = \Lambda_i\epsilon^2 + O(\epsilon^3), \text{ where } \Lambda_i = \gamma_i^2 > 0.$$

Proof. This follows readily from the hypothesis since

$$\begin{aligned}\lambda_i(\epsilon) =& \mathbf{q}_i^T(\epsilon)A^T(\epsilon)A(\epsilon)\mathbf{q}_i(\epsilon)\\ =& \|A(\epsilon)\mathbf{q}_i(\epsilon)\|^2\\ =& \Lambda_i\epsilon^2 + O(\epsilon^3),\end{aligned}$$

where $\Lambda_i = \gamma_i^2 > 0$. □

We shall explain why the hypothesis imposed on $A(\epsilon)\mathbf{q}_i(\epsilon)$ is meaningful. Note that for $7 \leq i \leq 8$,

$$\begin{aligned}A(\epsilon)\mathbf{q}_i(\epsilon) &= (A_R + \epsilon A_T)\left(\mathbf{r}_i'(\epsilon) + \mathbf{z}_i(\epsilon)\right)\\ &= A_R\mathbf{z}_i(\epsilon) + \epsilon A_T\mathbf{r}_i'(\epsilon) + O(\epsilon^2)\\ &= \epsilon\left(\frac{1}{\epsilon}A_R\mathbf{z}_i(\epsilon) + A_T\mathbf{r}_i'(\epsilon)\right) + O(\epsilon^2),\end{aligned}$$

where $A_R\mathbf{z}_i(\epsilon) + \epsilon A_T\mathbf{r}_i'(\epsilon) = O(\epsilon)$, which is the first order approximation of the residue of $A(\epsilon)$ associated with the solution $\mathbf{q}_i(\epsilon)$. The hypothesis that $\gamma_i > 0$ is intimately related to the assumption that we are dealing with a non-degenerate scene configuration in a differential setting. The reason can be seen by looking at the square of the coefficient of the first order term in the preceding equation and substituting the expressions for A_R and A_T from Equation (2.21):

$$\|\frac{1}{\epsilon}A_R\mathbf{z}_i(\epsilon) + A_T\mathbf{r}_i'(\epsilon)\|^2 = \sum_{j=1}^{N}((\underline{\Delta_t\mathbf{p}_j})^T\widetilde{\mathbf{r}_i'(\epsilon)}\,\underline{\mathbf{p}_j} + \underline{\mathbf{p}_j}^T\widetilde{\mathbf{z}_i'(\epsilon)}\,\underline{\mathbf{p}_j})^2 \tag{2.26}$$

where

$$\mathbf{z}_i'(\epsilon) = \frac{1}{\epsilon}\mathbf{z}_i(\epsilon) = O(\epsilon^0),$$

$$\underline{\mathbf{p}_j} = [\ \underline{x_j}\ \ \underline{y_j}\ \ 1\ ]^T,\ \underline{\Delta_t \mathbf{p}_j} = [\ \underline{\Delta_t x_j}\ \ \underline{\Delta_t y_j}\ \ 0\ ]^T$$

Equation (2.26) should be familiar to most vision researchers: it is the sum squared error of the differential fundamental matrix [69, 105] associated with the "solution" $\mathbf{r}'_i(\epsilon)$ and $\mathbf{z}'_i(\epsilon)$, with $\mathbf{r}'_i(\epsilon)$ representing the translational velocity and $\mathbf{z}'_i(\epsilon)$ related to the angular velocity.

From Theorem 2.4, for $i = 7, 8$, $\widehat{\mathbf{r}'_i(\epsilon)} = \widehat{\mathbf{T}'_i(\epsilon)}$ where $\mathbf{T}'_i(\epsilon)$ is orthogonal to the true translation $\mathbf{T}$ and

$$\|\mathbf{T}'_i(\epsilon)\| = \frac{1}{\sqrt{2}}.$$

Thus, the positivity hypothesis made by Lemma 2.7 on the first order term γ_i amounts to saying that when we substitute with a translation vector orthogonal to the true translation, the sum squared error must be greater than zero. This hypothesis must hold, otherwise the scene in view would be degenerate to the differential fundamental matrix. Therefore, using Lemma 2.7, we can say that

$$\lambda_i(\epsilon) = \epsilon^2 \Lambda_i + O(\epsilon^3), \tag{2.27}$$

$$\text{where } \Lambda_i > 0, 7 \leq i \leq 8.$$

Hence, in principle, under noiseless condition, there is no degeneracy in the solution to the eight point algorithm even under infinitesimal motion. The question of whether this non-degeneracy is sufficient to ensure stability under the proportional noise model is one which will investigate in the subsequent sections.

2.5 Eigenvalues of $A^T(\epsilon)A(\epsilon)$ under Noise

Having determined how the eigenvectors and eigenvalues of $A^T(\epsilon)A(\epsilon)$ vary with ϵ, we are now in a position to determine how they are affected by noise.

Let the corrupted data matrix $\widetilde{A}(\epsilon)$ be of the form

$$\widetilde{A}(\epsilon) = A(\epsilon) + M(\epsilon),$$

where $A(\epsilon)$ is defined in Equation (2.15) and $\|M(\epsilon)\| \leq \epsilon m$ for sufficiently small ϵ. The matrix $M(\epsilon)$ represents the proportional noise model, and m is some proportionality factor which is a function of the percentage noise in the optical flow.

Now, the matrix $\widetilde{A}^T(\epsilon)\widetilde{A}(\epsilon)$ is a perturbed version of $A^T(\epsilon)A(\epsilon)$ given by

$$\widetilde{A}^T(\epsilon)\widetilde{A}(\epsilon) = A^T(\epsilon)A(\epsilon) + B(\epsilon, M) \tag{2.28}$$

where

$$B(\epsilon, M) = A^T(\epsilon)M(\epsilon) + M^T(\epsilon)A(\epsilon) + M^T(\epsilon)M(\epsilon) = O(\epsilon)m. \tag{2.29}$$

The estimated solution is obtained by finding an eigenvector $\widetilde{\mathbf{q}}_9(\epsilon, M)$ of $\widetilde{A}^T(\epsilon)\widetilde{A}(\epsilon)$ that corresponds to its smallest eigenvalue $\widetilde{\lambda}_9(\epsilon, M)$. Thus, we have

$$\left(A^T(\epsilon)A(\epsilon) + B(\epsilon, M)\right)\widetilde{\mathbf{q}}_9(\epsilon, M) = \widetilde{\lambda}_9(\epsilon, M)\widetilde{\mathbf{q}}_9(\epsilon, M). \tag{2.30}$$

Note that both the eigenvalues $\widetilde{\lambda}_i(\epsilon, M)$ and eigenvectors $\widetilde{\mathbf{q}}_i(\epsilon, M)$ of $\widetilde{A}^T(\epsilon)\widetilde{A}(\epsilon)$ are functions of ϵ and M. Henceforth, we rely on the $\widetilde{\ }$ notation to remind the reader of the dependence on M, suppressing M in these cases to keep our notation simple. However, in cases where the dependence on M is not clear, we will explicitly write down the dependence.

2.5.1 Eigenvalue $\widetilde{\lambda}_9(\epsilon)$

In this subsection, we shall determine the order of the eigenvalue $\widetilde{\lambda}_9(\epsilon)$ via the error $|\widetilde{\lambda}_9(\epsilon) - \lambda_9(\epsilon)|$, where $\lambda_9(\epsilon) = 0$. Specifically, we shall prove that $\widetilde{\lambda}_9(\epsilon) = O(\epsilon^2)m$ for a sufficiently small m.

Unfortunately, the standard results in perturbation theory only lead to $|\widetilde{\lambda}_i(\epsilon) - \lambda_i(\epsilon)| = O(\epsilon)m$ for each i, from which we are not able to deduce that $\widetilde{\lambda}_9(\epsilon)$ is simple since the three Gerschgorin's discs might overlap. To overcome this difficulty, we apply the techniques developed in [106] and prove a modified result of Gerschgorin's Theorems, namely Proposition A.3 in Appendix A.1. For readers not familiar with Gerschgorin's Theorems and the notion of Gerschgorin's discs, please refer to Theorems A.1 and A.2 in Appendix A.1.

Before we apply Proposition A.3, let us record the following simple result which plays an important role in providing the order of eigenvalues and is also crucial for obtaining the projection $\alpha_i(\epsilon, M)$ of $\widetilde{\mathbf{q}}_9(\epsilon)$ along $\mathbf{q}_i(\epsilon)$ in Section 2.6.

Lemma 2.8.

(a) If either i or k is in the set $\{1,2,3,4,5,6\}$, then

$$\|\mathbf{q}_k^T(\epsilon)B(\epsilon,M)\mathbf{q}_i(\epsilon)\| = O(\epsilon)m$$

(b) If both i and k are in the set $\{7,8,9\}$, then

$$\|\mathbf{q}_k^T(\epsilon)B(\epsilon,M)\mathbf{q}_i(\epsilon)\| = O(\epsilon^2)m$$

Proof. Part (a) is straightforward from Equation (2.29). For part (b), we use $\mathbf{q}_i(\epsilon) = \mathbf{r}'_i(\epsilon) + \mathbf{z}_i(\epsilon)$ in Theorem 2.4, and the data matrix expression in Equation (2.20) to obtain

$$\mathbf{q}_k^T(\epsilon)B(\epsilon,M)\mathbf{q}_i(\epsilon) = \left(\mathbf{r}'_k(\epsilon) + \mathbf{z}_k(\epsilon)\right)^T \left(A_R^T M(\epsilon) + M^T(\epsilon)A_R\right)\left(\mathbf{r}'_i(\epsilon) + \mathbf{z}_i(\epsilon)\right) + O(\epsilon^2)m^2$$

When $7 \le k \le 9$ and $7 \le i \le 9$, we have $\lambda_k = 0$ and $\lambda_i = 0$. Using Theorem 2.4, we have $A_R\mathbf{r}'_k(\epsilon) = \mathbf{0}$ and $A_R\mathbf{r}'_i(\epsilon) = \mathbf{0}$. Hence, we have

$$\begin{aligned}
& \mathbf{q}_k^T(\epsilon)B(\epsilon,M)\mathbf{q}_i(\epsilon) \\
= \; & \left(\mathbf{r}'_k(\epsilon) + \mathbf{z}_k(\epsilon)\right)^T \left(A_R^T M(\epsilon) + M^T(\epsilon)A_R\right)\left(\mathbf{r}'_i(\epsilon) + \mathbf{z}_i(\epsilon)\right) + O(\epsilon^2)m^2 \\
= \; & \mathbf{z}_k^T(\epsilon)A_R^T M(\epsilon)\mathbf{r}'_i(\epsilon) + \mathbf{r}'^T_k(\epsilon)M^T(\epsilon)A_R\mathbf{z}_i(\epsilon) + \mathbf{z}_k^T(\epsilon)A_R^T M(\epsilon)\mathbf{z}_i(\epsilon) + O(\epsilon^2)m^2 \\
= \; & O(\epsilon^2)m
\end{aligned}$$

□

If we can now specially construct an invertible matrix K such that the matrix $K^{-1}A^T(\epsilon)A(\epsilon)K$ becomes a diagonal matrix $\text{Diag}(\lambda_i(\epsilon))$, then Proposition A.3 provides an upper bound for $|\widetilde{\lambda}_i(\epsilon) - \lambda_i(\epsilon)|$. Our aim is to have upper bounds on $|\widetilde{\lambda}_i(\epsilon) - \lambda_i(\epsilon)|$ which enable us to isolate the 9th circular disc $\widetilde{G}_9$ from the other $\widetilde{G}_i$'s.

As mentioned above, the standard result from perturbation theory ([106]) is not adequate as there remains a possibility that the 9th circular disc $\widetilde{G}_9$ defined in Proposition A.3 might overlap with other disc $\widetilde{G}_i$, in which case $\widetilde{\lambda}_9(\epsilon)$ would lie in the union of the discs. We need to choose K properly so that $\widetilde{G}_9$ is isolated from the other $\widetilde{G}_i$. We shall now work towards a suitable choice of an invertible matrix K.

First consider the matrix $Q^T(\epsilon)\widetilde{A}^T(\epsilon)\widetilde{A}(\epsilon)Q(\epsilon)$ where $Q(\epsilon)$ is the matrix whose ith column is the unit eigenvector $\mathbf{q}_i(\epsilon)$ of the real symmetric matrix $A^T(\epsilon)A(\epsilon)$. Clearly, $Q^{-1}(\epsilon) = Q^T(\epsilon)$. From Equation (2.28), we have

$$Q^T(\epsilon)\widetilde{A}^T(\epsilon)\widetilde{A}(\epsilon)Q(\epsilon) = \text{Diag}(\lambda_i(\epsilon)) + Q^T(\epsilon)B(\epsilon, M)Q(\epsilon) \tag{2.31}$$

where $\text{Diag}(\lambda_i(\epsilon))$ is a diagonal matrix whose ith diagonal entry is $\lambda_i(\epsilon)$.

By Proposition A.3, every eigenvalue $\widetilde{\lambda}_j(\epsilon)$ of $\widetilde{A}^T(\epsilon)\widetilde{A}(\epsilon)$ lies in at least one of the circular discs with center $\lambda_k(\epsilon)$ and radius

$$\sum_{i=1}^{9} |\mathbf{q}_k^T(\epsilon)B(\epsilon, M)\mathbf{q}_i(\epsilon)| = O(\epsilon)m.$$

(Note that Proposition A.3 does not imply that j is necessarily equal to k.) Now, the center of the 9th circular disc is 0 while those of the 7th and 8th circular discs are $0 + O(\epsilon^2)$ (from Equation (2.27)). However, all three circular discs have radii of order $O(\epsilon)$ by the preceding equation. Consequently, for a sufficiently small ϵ, these three discs may overlap with each other, and $\widetilde{\lambda}_9(\epsilon)$ lies in their union. As such, for this naive choice of $K = Q(\epsilon)$, we are not able to ascertain a good upper bound for $|\widetilde{\lambda}_9(\epsilon) - \lambda_9(\epsilon)|$.

Fortunately, by inspecting the entries in the matrix
$Q(\epsilon)^T B(\epsilon, M)Q(\epsilon)$, we find that the upper bound on $|\widetilde{\lambda}_i(\epsilon) - \lambda_i(\epsilon)|$ can be improved by pre- and post-multiplying the matrix in Equation (2.31) with the respective matrices $S^{-1}(\epsilon)$ and $S(\epsilon)$, where

$$S(\epsilon) = \begin{bmatrix} \epsilon I_6 & 0 \\ 0 & I_3 \end{bmatrix}.$$

Here, I_n denotes the $n \times n$ identity matrix, and 0 is a zero matrix. The effect of post-multiplying a matrix by S is the same as multiplying its first six columns by ϵ while pre-multiplying a matrix by S^{-1} is the same as multiplying its first six rows by $\frac{1}{\epsilon}$. So, we have

$$S^{-1}(\epsilon)Q^T(\epsilon)B(\epsilon, M)Q(\epsilon)S(\epsilon) =$$

$$\left[\begin{array}{ccc|c} \mathbf{q}_1^T(\epsilon)B\mathbf{q}_1(\epsilon) & \cdots & \mathbf{q}_1^T(\epsilon)B\mathbf{q}_6(\epsilon) & \frac{1}{\epsilon}\mathbf{q}_1^T(\epsilon)B\mathbf{q}_7(\epsilon) \ \cdot \ \frac{1}{\epsilon}\mathbf{q}_1^T(\epsilon)B\mathbf{q}_9(\epsilon) \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{q}_6^T(\epsilon)B\mathbf{q}_1(\epsilon) & \cdots & \mathbf{q}_6^T(\epsilon)B\mathbf{q}_6(\epsilon) & \frac{1}{\epsilon}\mathbf{q}_6^T(\epsilon)B\mathbf{q}_7(\epsilon) \ \cdot \ \frac{1}{\epsilon}\mathbf{q}_6^T(\epsilon)B\mathbf{q}_9(\epsilon) \\ \hline \epsilon\mathbf{q}_7^T(\epsilon)B\mathbf{q}_1(\epsilon) & \cdots & \epsilon\mathbf{q}_7^T(\epsilon)B\mathbf{q}_6(\epsilon) & \mathbf{q}_7^T(\epsilon)B\mathbf{q}_7(\epsilon) \ \cdot \ \mathbf{q}_7^T(\epsilon)B\mathbf{q}_9(\epsilon) \\ \cdots & \cdots & \cdots & \cdots \\ \epsilon\mathbf{q}_9^T(\epsilon)B\mathbf{q}_1(\epsilon) & \cdots & \epsilon\mathbf{q}_9^T(\epsilon)B\mathbf{q}_6(\epsilon) & \mathbf{q}_9^T(\epsilon)B\mathbf{q}_7(\epsilon) \ \cdot \ \mathbf{q}_9^T(\epsilon)B\mathbf{q}_9(\epsilon) \end{array}\right]$$

in which the diagonal sub-matrices of the above matrix

$S^{-1}(\epsilon)Q^T(\epsilon)B(\epsilon, M)Q(\epsilon)S(\epsilon)$ remain the same as those of $Q^T(\epsilon)B(\epsilon, M)Q(\epsilon)$.

Note that the above transformation does not affect the eigenvalues. Thus, we have

$$S^{-1}(\epsilon)Q^T(\epsilon)\widetilde{A}^T(\epsilon)\widetilde{A}(\epsilon)Q(\epsilon)S(\epsilon)$$

$$= \text{Diag}(\lambda_i(\epsilon)) + S^{-1}(\epsilon)Q^T(\epsilon)B(\epsilon, M)Q(\epsilon)S(\epsilon).$$

By Proposition A.3, where $K = Q(\epsilon)S(\epsilon)$, every eigenvalue of $\widetilde{A}^T(\epsilon)\widetilde{A}(\epsilon)$ lies in at least one of the circular discs $\widetilde{G}_k$ with center $\lambda_k(\epsilon)$ and radius

$$d_k(\epsilon, M) = \begin{cases} \sum_{i=1}^{6} |\mathbf{q}_k^T(\epsilon)B(\epsilon, M)\mathbf{q}_i(\epsilon)| + \frac{1}{\epsilon}\sum_{i=7}^{9} |\mathbf{q}_k^T(\epsilon)B(\epsilon, M)\mathbf{q}_i(\epsilon)| \\ = O(\epsilon)m, 1 \le k \le 6 \\ \epsilon\sum_{i=1}^{6} |\mathbf{q}_k^T(\epsilon)B(\epsilon, M)\mathbf{q}_i(\epsilon)| + \sum_{i=7}^{9} |\mathbf{q}_k^T(\epsilon)B(\epsilon, M)\mathbf{q}_i(\epsilon)| \\ = O(\epsilon^2)m, 7 \le k \le 9 \end{cases} \tag{2.32}$$

Using the above, we may now prove that the desired property that the 9th circular disc $\widetilde{G}_9$ is disjoint from the rest.

Proposition 2.9. *For a sufficiently small ϵ and m, the 9th circular disc $\widetilde{G}_9$ is*

disjoint from the union $\widetilde{G}_7 \cup \widetilde{G}_8$ *of the* 7*th and* 8*th circular discs, which in turn is disjoint from the union* $\cup_{i=1}^{6} \widetilde{G}_i$ *of the first six circular discs.*

Proof. First, the 9th circular disc $\widetilde{G}_9$ is disjoint from the union $\widetilde{G}_7 \cup \widetilde{G}_8$ of the 7th and 8th circular discs if the gap $(\Lambda_8 \epsilon^2 + O(\epsilon^3))$ between the disc centers $\lambda_8(\epsilon)$ and $\lambda_9(\epsilon)$ satisfies the following:

$$\Lambda_8 \epsilon^2 + O(\epsilon^3) > d_9(\epsilon, M) + \max(d_7(\epsilon, M), d_8(\epsilon, M)) \tag{2.33}$$

For a sufficiently small ϵ such that the left hand side is more than $\frac{1}{2}\Lambda_8 \epsilon^2$, and the right hand side is $O(\epsilon^2)m$, by Equation (2.32), we have a unform bound on m (independent of ϵ) such that for a sufficiently small m, the above condition (2.33) is satisfied.

Next, the union $\cup_{i=1}^{6} \widetilde{G}_i$ of the first six circular discs is disjoint from the the union $\widetilde{G}_7 \cup \widetilde{G}_8$ of the 7th and 8th circular discs, if the gap between the two nearest disc centers $\lambda_6(\epsilon)$ and $\lambda_7(\epsilon)$ satisfies the following:

$$\lambda_6(\epsilon) - \lambda_7(\epsilon) > \max_{j=7,8}(d_j(\epsilon, M)) + \max_{1 \le k \le 6} d_k(\epsilon, M).$$

From Proposition 2.6, we note that

$$\lambda_k(\epsilon) = \lambda_k + O(\epsilon) \text{ where } \lambda_k > 0, \text{ for } 1 \le k \le 6,$$

while under a non-degenerate scene, Equation (2.27) holds:

$$\lambda_j(\epsilon) = \Lambda_j \epsilon^2 + O(\epsilon^3) \text{ where } \Lambda_j > 0 \text{ for } 7 \leq j \leq 8.$$

Thus the above condition is satisfied if

$$\lambda_6 - \Lambda_7 \epsilon^2 + O(\epsilon) > \max_{j=7,8}(d_j(\epsilon, M)) + \max_{1 \leq k \leq 6} d_k(\epsilon, M). \tag{2.34}$$

However, from Equation (2.32), we have

$$\max_{j=7,8}(d_j(\epsilon, M)) = O(\epsilon^2)m,$$

and

$$\max_{1 \leq k \leq 6} d_k(\epsilon, M) = O(\epsilon)m.$$

Thus, since $\lambda_6 > 0$, the condition (2.34) is satisfied for a sufficiently small ϵ when m is sufficiently small (i.e., when there is a sufficiently small percentage noise).

Therefore, for a sufficiently small m, and a sufficiently small ϵ, the 9th circular disc $\widetilde{G}_9$ is disjoint from the union $\widetilde{G}_7 \cup \widetilde{G}_8$ of the 7th and 8th circular discs, which in turn is disjoint from the union $\cup_{i=1}^{6} \widetilde{G}_i$ of the first six circular discs.

□

It follows from the second part of Proposition A.3 that $\widetilde{\lambda}_9(\epsilon)$ lies in $\widetilde{G}_9$, and from Equation (2.32), we record the following result:

Theorem 2.10. *For a sufficiently small m, and a sufficiently small ϵ, the eigenvalue $\widetilde{\lambda}_9(\epsilon)$ is simple and*

$$\widetilde{\lambda}_9(\epsilon) = O(\epsilon^2)m.$$

Moreover,

$$\begin{aligned} \widetilde{\lambda}_i(\epsilon) &= \Lambda_i\epsilon^2 + O(\epsilon^2)m, i = 7, 8; \\ \widetilde{\lambda}_i(\epsilon) &= \lambda_i + O(\epsilon) + O(\epsilon)m, i = 1, 2, 3, 4, 5, 6. \end{aligned} \tag{2.35}$$

2.6 Projection of $\widetilde{\mathbf{q}}_9(\epsilon)$ along $\mathbf{q}_k(\epsilon)$

From the preceding section, when m is small, $\widetilde{\lambda}_9(\epsilon)$ is simple for sufficiently small ϵ. Therefore, its corresponding eigen-space is 1-dimensional. Let $\widetilde{\mathbf{q}}_9(\epsilon)$ (which may not be a unit vector) be an eigenvector corresponding to the eigenvalue $\widetilde{\lambda}_9(\epsilon)$ and expressed in the form

$$\widetilde{\mathbf{q}}_9(\epsilon) = \sum_{i=1}^{9} \alpha_i(\epsilon, M)\mathbf{q}_i(\epsilon). \tag{2.36}$$

Then the perturbation introduced to $\widetilde{\mathbf{q}}_9(\epsilon)$ can be analyzed by looking at the projection coefficients $\alpha_i(\epsilon, M)$ using the same technique in [106].

The following result is simple yet useful in the sequel.

Lemma 2.11.

$$\left(\widetilde{\lambda}_9(\epsilon) - \lambda_j(\epsilon)\right)\alpha_j(\epsilon, M) = \sum_{i=1}^{9} \alpha_i(\epsilon, M)\mathbf{q}_j^T(\epsilon)B(\epsilon, M)\mathbf{q}_i(\epsilon) \tag{2.37}$$

Proof. Substituting (2.36) into (2.30), and using

$$A^T(\epsilon)A(\epsilon)\mathbf{q}_i(\epsilon) = \lambda_i(\epsilon)\mathbf{q}_i(\epsilon),$$

we have

$$\sum_{i=1}^{9} \alpha_i(\epsilon, M)\lambda_i(\epsilon)\mathbf{q}_i(\epsilon) + \sum_{i=1}^{9} \alpha_i(\epsilon, M)B(\epsilon, M)\mathbf{q}_i(\epsilon) = \widetilde{\lambda}_9(\epsilon)\left(\sum_{i=1}^{9} \alpha_i(\epsilon, M)\mathbf{q}_i(\epsilon)\right).$$

Pre-multiplying the above equation with $\mathbf{q}_j^T(\epsilon)$, we obtain the required relation (2.37) by noting that

$$\mathbf{q}_j^T(\epsilon)\mathbf{q}_i(\epsilon) = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j. \end{cases}$$

□

Lemma 2.12. *Suppose the maximum projection is given by*

$$max\{|\alpha_i(\epsilon, M)| \mid 1 \leq i \leq 9\} = |\alpha_{i_*}(\epsilon, M)|$$

for some i_* *in* $1 \leq i \leq 9$. *Then*

$$\left|\widetilde{\lambda}_9(\epsilon) - \lambda_{i_*}(\epsilon)\right| \leq \sum_{i=1}^{9} \left|\mathbf{q}_j^T(\epsilon)B(\epsilon, M)\mathbf{q}_i(\epsilon)\right| = O(\epsilon)m.$$

Proof. Dividing Equation (2.37) by $\alpha_{i_*}(\epsilon, M)$ yields

$$\left(\widetilde{\lambda}_9(\epsilon) - \lambda_{i_*}(\epsilon)\right) = \sum_{i=1}^{9} \frac{\alpha_i(\epsilon, M)}{\alpha_{i_*}(\epsilon, M)}\mathbf{q}_j^T(\epsilon)B(\epsilon, M)\mathbf{q}_i(\epsilon)$$

with $|\frac{\alpha_i(\epsilon,M)}{\alpha_{i_*}(\epsilon,M)}| \leq 1$ for $1 \leq i \leq 9$. Thus we have

$$\left|\widetilde{\lambda}_9(\epsilon) - \lambda_{i_*}(\epsilon)\right| \leq \sum_{i=1}^{9} \left| \frac{\alpha_i(\epsilon, M)}{\alpha_{i_*}(\epsilon, M)} \mathbf{q}_j^T(\epsilon) B(\epsilon, M) \mathbf{q}_i(\epsilon) \right|$$

$$\leq \sum_{i=1}^{9} \left| \mathbf{q}_j^T(\epsilon) B(\epsilon, M) \mathbf{q}_i(\epsilon) \right| = O(\epsilon) m.$$

The order follows from Lemma 2.8.

□

Theorem 2.13. *For a sufficiently small noise m (and hence M) and a sufficiently small ϵ, the maximum projection is given by*

$$max\{|\alpha_i(\epsilon, M)| \mid 1 \leq i \leq 9\} = |\alpha_9(\epsilon, M)|.$$

Proof. We first prove that for every j in $1 \leq j \leq 6$,

$$|\alpha_j(\epsilon, M)| \neq \max\{|\alpha_i(\epsilon, M)| \mid 1 \leq i \leq 9\}$$

for a sufficiently small ϵ (for a given M).

Suppose on the contrary that, for some j_* in $1 \leq j \leq 6$, there is a sequence $\{\epsilon_s\}$ with $\lim_{s\to\infty} \epsilon_s = 0$ and $|\alpha_{j_*}(\epsilon_s, M)| = \max\{|\alpha_i(\epsilon_s, M)| \mid 1 \leq i \leq 9\}$.

By Lemma 2.12, we have

$$\left|\widetilde{\lambda}_9(\epsilon_s) - \lambda_{j_*}(\epsilon_s)\right| \leq \sum_{i=1}^{9} \left| \mathbf{q}_j^T(\epsilon_s) B(\epsilon_s, M) \mathbf{q}_i(\epsilon_s) \right|$$

As $s \to \infty$, we note that the right hand side of the above inequality approaches 0, by Lemma 2.12. Using Theorem 2.10 and Proposition 2.6, the left hand side approaches λ_{j_*}, which is positive. This yields a contradiction.

Therefore, $|\alpha_j(\epsilon, M)|$ is non-maximal for $1 \leq j \leq 6$, when ϵ is sufficiently small.

Now, we shall prove, again by contradiction, that for every j where $7 \leq j \leq 8$,

$$|\alpha_j(\epsilon, M)| \neq \max\{|\alpha_i(\epsilon, M_s)| \mid 1 \leq i \leq 9\}$$

for a sufficiently small M and a sufficiently small ϵ.

Suppose for some $j_* \in \{7, 8\}$, there is a sequence $\{M_s\}$ with $\lim_{s\to\infty} \|M_s\| = 0$ and $|\alpha_{j_*}(\epsilon, M_s)| = \max\{|\alpha_i(\epsilon, M_s)| \mid 1 \leq i \leq 9\}$. By Lemma 2.12, we have

$$\left|\widetilde{\lambda}_9(\epsilon) - \lambda_{j_*}(\epsilon)\right| \leq \sum_{i=1}^{9} \left|\mathbf{q}_j^T(\epsilon) B(\epsilon, M_s) \mathbf{q}_i(\epsilon)\right|.$$

As $s \to \infty$, we note that the right hand side approaches 0, since $\lim_{s\to\infty} \|M_s\| = 0$. The left hand side approaches $\Lambda_{j_*}\epsilon^2$, by Theorem 2.10 and Proposition 2.7. However, $\Lambda_{j_*}\epsilon^2$ is positive for a sufficiently small $\epsilon > 0$. This yields a contradiction.

Therefore, for a sufficiently small m (and hence M) and a sufficiently small ϵ, the projection $|\alpha_j(\epsilon, M)|$, for $7 \leq j \leq 8$, is non-maximal.

We conclude that for a sufficiently small m and a sufficiently small ϵ, the projection $|\alpha_9(\epsilon, M)|$ is maximal. □

Note that the following vector

$$\frac{1}{\alpha_9(\epsilon, M)}\widetilde{\mathbf{q}}_9(\epsilon) = \sum_{i=1}^{9} \frac{\alpha_i(\epsilon, M)}{\alpha_9(\epsilon, M)}\mathbf{q}_i(\epsilon)$$

is an eigenvector of $\widetilde{A}^T(\epsilon)\widetilde{A}(\epsilon)$ corresponding to the eigenvalue $\widetilde{\lambda}_9(\epsilon)$ with $|\frac{\alpha_i(\epsilon,M)}{\alpha_9(\epsilon,M)}| \leq 1$ for $1 \leq i \leq 8$.

Thus, from now on, for a sufficiently small ϵ and m, we may assume that

$$\widetilde{\mathbf{q}}_9(\epsilon) = \sum_{i=1}^{9} \alpha_i(\epsilon, M)\mathbf{q}_i(\epsilon), \tag{2.38}$$

where $\alpha_9(\epsilon, M) = 1$ and $|\alpha_i(\epsilon, M)| \leq 1$ for $1 \leq i \leq 8$. Note that with $\alpha_9(\epsilon, M) = \alpha_{i*}(\epsilon, M) = 1$, both Lemmas 2.11 and 2.12 still apply. We shall now proceed to determine the upper bounds on $\alpha_i(\epsilon, M)$ for $1 \leq i \leq 8$.

Proposition 2.14. *For $1 \leq j \leq 6$, for sufficiently small m and ϵ , we have*

$$\alpha_j(\epsilon, M) = O(\epsilon)m.$$

Proof. From Lemma 2.11, we have

$$\left|\widetilde{\lambda}_9(\epsilon) - \lambda_j(\epsilon)\right| \ |\alpha_j(\epsilon, M)| \leq \sum_{i=1}^{9} \left|\mathbf{q}_j^T(\epsilon)B(\epsilon, M)\mathbf{q}_i(\epsilon)\right|,$$

as each $|\alpha_i(\epsilon, M)| \leq 1$. By Theorem 2.10 and Proposition 2.7, we have $\left|\widetilde{\lambda}_9(\epsilon) - \lambda_j(\epsilon)\right| = \lambda_j + O(\epsilon)$ while

$\sum_{i=1}^{9} \left|\mathbf{q}_j^T(\epsilon)B(\epsilon, M)\mathbf{q}_i(\epsilon)\right| = O(\epsilon)m$, by Lemma 2.8. Therefore, we have

$$\alpha_j(\epsilon, M) = O(\epsilon)m$$

for a sufficiently small ϵ and m. □

Lemma 2.15. *For $j = 7$ or 8, when m is sufficiently small, we have*

$$|\sum_{i=1}^{9} \alpha_i(\epsilon, M)\mathbf{q}_j^T(\epsilon)B(\epsilon, M)\mathbf{q}_i(\epsilon)| = O(\epsilon^2)m.$$

Proof. Note that

$$\sum_{i=1}^{9} |\alpha_i(\epsilon, M)\mathbf{q}_j^T(\epsilon)B(\epsilon, M)\mathbf{q}_i(\epsilon)|$$

$$\leq \sum_{i=1}^{6} |\alpha_i(\epsilon, M)\mathbf{q}_j^T(\epsilon)B(\epsilon, M)\mathbf{q}_i(\epsilon)| + \sum_{i=7}^{9} |\alpha_i(\epsilon, M)\mathbf{q}_j^T(\epsilon)B(\epsilon, M)\mathbf{q}_i(\epsilon)|.$$

For $7 \leq i \leq 9$, we use $|\alpha_i(\epsilon, M)| \leq 1$ and Lemma 2.8 to obtain

$$\sum_{i=7}^{9} |\alpha_i(\epsilon, M)\mathbf{q}_j^T(\epsilon)B(\epsilon, M)\mathbf{q}_i(\epsilon)| = O(\epsilon^2)m.$$

For $1 \leq i \leq 6$, by Proposition 2.14 and Lemma 2.8, we have

$$|\alpha_i(\epsilon, M)\mathbf{q}_j^T(\epsilon)B(\epsilon, M)\mathbf{q}_i(\epsilon)| = O(\epsilon^2)m^2.$$

Therefore,

$$\sum_{i=1}^{9} |\alpha_i(\epsilon, M)\mathbf{q}_j^T(\epsilon)B(\epsilon, M)\mathbf{q}_i(\epsilon)| = O(\epsilon^2)m.$$

□

We have the following bound on $\alpha_j(\epsilon, M)$ for $j = 7$ or 8.

Proposition 2.16. *For $7 \leq j \leq 8$, for sufficiently small m and ϵ, we have*

$$|\alpha_j(\epsilon, M)| = O(\epsilon^0)m.$$

Proof. Using Lemma 2.11, we obtain

$$|\alpha_j(\epsilon, M)| \leq \frac{\sum_{i=1}^{9} |\alpha_i(\epsilon, M)\mathbf{q}_j^T(\epsilon)B(\epsilon, M)\mathbf{q}_i(\epsilon)|}{\left|\widetilde{\lambda}_9(\epsilon) - \lambda_j(\epsilon)\right|}.$$

By Lemma 2.7 and Theorem 2.10, we have, for sufficiently small ϵ and m,

$$|\widetilde{\lambda}_9(\epsilon) - \lambda_j(\epsilon)| = \Lambda_j\epsilon^2 + O(\epsilon^3)m = \Lambda_j\epsilon^2\left(1 + O(\epsilon)m\right) > 0,$$

where $\Lambda_j > 0$. Therefore, together with Lemma 2.15, we conclude that $|\alpha_j(\epsilon, M)| = O(\epsilon^0)m$. □

As a consequence of the preceding proposition, we are able to set a uniform bound on m, i.e., independent of ϵ, such that $|\alpha_j(\epsilon, M)| < 1$ (and are as small as we like) for $7 \leq j \leq 8$. With this m, for a sufficiently small $\epsilon > 0$, we will also have $|\alpha_j(\epsilon, M)| < 1$ (and are as small as we like) for $1 \leq j \leq 6$. The above result proves that stability of the estimated solution $\widetilde{\mathbf{q}}_9(\epsilon)$ under the differential condition of small motion and a bounded percentage noise.

For comparison with other statistical analysis, we would like to find an explicit expression for the lowest order noise terms (i.e., the m terms) of $\alpha_j(\epsilon, M)$, via

Equation (2.37) in conjunction with the results obtained in Propositions 2.14 and 2.16 and Lemma 2.8. We shall state and prove the result in our last theorem:

Theorem 2.17. *Given a sufficiently small m, and a sufficiently small ϵ,*

$$\widetilde{\mathbf{q}}_9(\epsilon) = \sum_{i=1}^{9} \alpha_i(\epsilon, M)\mathbf{q}_i(\epsilon)$$

where $\alpha_9(\epsilon, M) = 1$,

$$\alpha_i(\epsilon, M) = -\frac{\mathbf{q}_i^T(\epsilon)B(\epsilon, M)\mathbf{q}_9(\epsilon)}{\lambda_i(\epsilon)} + O(\epsilon)m^2$$

$$= O(\epsilon)m, \ \textit{for } 1 \leq i \leq 6,$$

$$\alpha_i(\epsilon, M) = -\frac{\mathbf{q}_i^T(\epsilon)B(\epsilon, M)\mathbf{q}_9(\epsilon)}{\lambda_i(\epsilon)} + O(\epsilon^0)m^2$$

$$= O(\epsilon^0)m, \ \textit{for } 7 \leq i \leq 8.$$

Proof. From Equations (2.37) and (2.38) and Propositions 2.14 and 2.16, we have

$$\begin{aligned}
&\left(\widetilde{\lambda}_9(\epsilon) - \lambda_j(\epsilon)\right)\alpha_j(\epsilon, M) \\
&= \mathbf{q}_j^T(\epsilon)B(\epsilon, M)\mathbf{q}_9(\epsilon) + \sum_{i=1}^{8}\alpha_i(\epsilon, M)\mathbf{q}_j^T(\epsilon)B(\epsilon, M)\mathbf{q}_i(\epsilon) \\
&= \begin{cases} \mathbf{q}_j^T(\epsilon)B(\epsilon, M)\mathbf{q}_9(\epsilon) + O(\epsilon)m^2, 1 \leq j \leq 6, \\ \mathbf{q}_j^T(\epsilon)B(\epsilon, M)\mathbf{q}_9(\epsilon) + O(\epsilon^2)m^2, 7 \leq j \leq 8. \end{cases}
\end{aligned} \tag{2.39}$$

Applying Proposition 2.6 and Lemma 2.7, we have

$$\begin{aligned}\widetilde{\lambda}_9(\epsilon) - \lambda_j(\epsilon) &= -\lambda_j(\epsilon)\left(1 - \frac{\widetilde{\lambda}_9(\epsilon)}{\lambda_j(\epsilon)}\right) \\ &= \begin{cases} -\lambda_j(\epsilon)(1 + O(\epsilon^2)m), & \text{if } 1 \leq j \leq 6, \\ -\lambda_j(\epsilon)(1 + O(\epsilon^0)m), & \text{if } 7 \leq j \leq 8. \end{cases}\end{aligned}$$

Hence, by Equation (2.3), we have

$$\frac{1}{\widetilde{\lambda}_9(\epsilon) - \lambda_j(\epsilon)} = \begin{cases} \frac{1}{-\lambda_j(\epsilon)}(1 + O(\epsilon^2)m), & \text{if } 1 \leq j \leq 6, \\ \frac{1}{-\lambda_j(\epsilon)}(1 + O(\epsilon^0)m), & \text{if } 7 \leq j \leq 8. \end{cases}$$

Dividing Equation (2.39) throughout by $\widetilde{\lambda}_9(\epsilon) - \lambda_j(\epsilon)$ and using Lemma 2.8, we have the desired expression stated in the theorem. □

The above result shows that the lowest order terms in noise are the same as those derived in [[106], pages 70, 71, 83, where the noise is denoted by ϵ]. It allows us to extend much of the unbiasedness/noise whitening analysis carried out on the discrete eight point algorithm to the differential case, because the foundation of such analysis is the lowest order noise terms in the perturbation analysis. One example is the work of [74], which showed that for the so-called TLS-FC normalized variant of the discrete eight point algorithm[1], the expected value of the $\alpha_i(\epsilon, M)$ is zero for $i \neq 9$, that is, the estimated solution is unbiased.

[1] In the TLS-FC variant [74], matrix perturbation analysis was used to formulate a new data matrix $\widetilde{A'}(\epsilon)$ given by $\widetilde{A}'(\epsilon) = \widetilde{A}(\epsilon) + M'(\epsilon) = A(\epsilon) + \epsilon M + M'(\epsilon)$, where $M'(\epsilon)$ is chosen such that $\widetilde{A}'(\epsilon)$ satisfies the rank 8 constraint without making any changes to the columns of $\widetilde{A}(\epsilon)$ in which noise is not present, and that $\|M'(\epsilon)\|$ is minimized. A straightforward application of the result in Equation (2.41) in Section 2.7 shows that this new $\widetilde{A'}(\epsilon)$ is bounded by the same proportional noise regime, and thus the results from our thesis are applicable to this TLS-FC variant.

2.7 Obtaining the Rotation and Translation Parameters

To complete our investigation, we need to ensure that the subsequent decomposition of the fundamental matrix $\widetilde{F}(\epsilon)$ into the translation and the rotation estimates are stable. Several intermediate steps are also involved, including the correcting of $\widetilde{F}(\epsilon)$ to the nearest rank-two matrix, and the correcting of the recovered essential matrix $\widetilde{E}(\epsilon)$ to the nearest matrix with the desired property of having the first two singular values being equal.

In addition, to establish a proper comparison between the discrete and the differential formulation, we need to convert the rotational and translational displacements recovered from the discrete algorithm into the corresponding velocity formulations. As mentioned previously, the differential two view formulation converts the SfM problem into one independent of ϵ but involving differential entities like velocity. Accordingly, the required orders in the errors of the discrete estimates so that the corresponding velocity estimates have errors independent of ϵ are given by:

$$\begin{aligned} &\text{Error in the translation direction} = O(\epsilon^0)m, \\ &\text{Error in the rotation estimate} = O(\epsilon)m. \end{aligned} \tag{2.40}$$

The above means that when the discrete rotation estimate is divided by the time ϵ to get the rotational velocity, the latter's error would be independent of ϵ. The error in the translation estimate only needs $O(\epsilon^0)m$ instead of $O(\epsilon)m$ because the terms $\mathbf{T}_c$ and $\mathbf{T}$ in our formulation in fact represent velocities already (see

Equations (3.1) and (2.19)). This is related to the fact that we can only recover the translation direction anyway.

The overall order of the error provided in Theorem 2.17 is $O(\epsilon^0)m$ and superficially, does not give us much hope that the discrete eight point algorithm can meet the condition set out in Equation (2.40) for rotation, which requires error of the order $O(\epsilon)m$. Fortunately, Theorem 2.17 also shows that the orders of the perturbation coefficients $\alpha_i(\epsilon, M)'s$ are not all equal. In fact, only two coefficients, $\alpha_7(\epsilon, M)$ and $\alpha_8(\epsilon, M)$, are of order $O(\epsilon^0)m$, whereas the rest are of order $O(\epsilon)m$.

We can obtain a better bound if we split the recovered fundamental matrix $\widetilde{F}(\epsilon) = \widetilde{\mathbf{q}_9(\epsilon)}$ into a sum of two terms, such that the large $O(\epsilon^0)m$ noise only perturbs the translation vector. The rest of this section and Appendix A.2 are devoted to doing just such a split and keeping track of how the errors are propagated and apportioned in the subsequent decomposition into the translation and rotation estimates.

2.7.1 Some Preliminaries

Before proceeding further, the following short note on 'nearest matrix' will be used extensively in the discussion of the various errors throughout this section.

Let $\widetilde{C}(\epsilon)$ be the noise corrupted version of a matrix $C(\epsilon)$. Due to the noise, $\widetilde{C}(\epsilon)$ may lack some desired properties which are present in $C(\epsilon)$ (an example of such property is that the first two singular values are identical or the rank is 2).

As is often the case, we use the 'nearest' $\widetilde{C}'(\epsilon)$ to $\widetilde{C}(\epsilon)$ (if it exists) instead of $\widetilde{C}(\epsilon)$ in the following sense:

(a) $\widetilde{C}'(\epsilon)$ possesses the desired properties, and

(b) $\|\widetilde{C}'(\epsilon) - \widetilde{C}(\epsilon)\|$
$= \min\Big\{\|K - \widetilde{C}(\epsilon)\| \mid K \text{ possesses the desired properties}\Big\}$.

Thus, we have,

$$\|\widetilde{C}'(\epsilon) - \widetilde{C}(\epsilon)\| \leq \|C(\epsilon) - \widetilde{C}(\epsilon)\|. \tag{2.41}$$

This ensures that in using the nearest matrix, the 'correction' introduced has the same order of error.

Note that the essential/ fundamental matrix is only defined up to a scale factor. We regard the 'true' fundamental matrix $F_t(\epsilon)$ as one having unit Frobenius norm and given by

$$F_t(\epsilon) = \overbrace{\mathbf{q}_9(\epsilon)} = \widehat{\mathbf{T}}_t \Theta R^T(\epsilon)(\Theta(\epsilon))^{-1} \tag{2.42}$$

where $\mathbf{T}_t$ is parallel to $\mathbf{T}$ defined in Equation (2.19) but scaled such that $\|F_t(\epsilon)\| = 1$. The true essential matrix $E_t(\epsilon)$ is defined as the de-normalized version of $F_t(\epsilon)$,

$$E_t(\epsilon) = \Theta^T F_t(\epsilon)\Theta(\epsilon) = \Theta^T \widehat{\mathbf{T}}_t \Theta R^T(\epsilon). \tag{2.43}$$

The estimated fundamental matrix is given by

$$\widetilde{F}(\epsilon) = \widehat{\widetilde{\mathbf{q}}_9(\epsilon)}\,.$$

With noise, $\widehat{\widetilde{\mathbf{q}}_9(\epsilon)}$ may not be a unit vector and thus may not have unit norm. Letting $\widetilde{F}(\epsilon)$ stay un-normalized has the virtue of keeping the following proof simple while still obtaining error expressions that suffice for our purpose.

2.7.2 Splitting the Fundamental Matrix

We know from Theorems 2.4 and 2.17 that our estimated solution vector $\widetilde{\mathbf{q}}_9(\epsilon)$ can be expressed as

$$\begin{aligned}\widetilde{\mathbf{q}}_9(\epsilon) =& \mathbf{q}_9(\epsilon) + \sum_{i=7}^{8} \alpha_i(\epsilon, M)\mathbf{r}'_i(\epsilon) + \sum_{i=7}^{8} \alpha_i(\epsilon, M)\mathbf{z}_i(\epsilon) + \sum_{i=1}^{6} \alpha_i(\epsilon, M)\mathbf{q}_i(\epsilon) \\ =& \mathbf{q}_9(\epsilon) + \sum_{i=7}^{8} \alpha_i(\epsilon, M)\mathbf{r}'_i(\epsilon) + O(\epsilon)m.\end{aligned}$$

where

$$\widehat{\mathbf{r}'_i(\epsilon)} = \widehat{\mathbf{T}'_i(\epsilon)},\ 7 \leq i \leq 8.$$

Therefore, using the definition of $F_t(\epsilon)$ in Equation (2.42), we have

$$\begin{aligned}\widetilde{F}(\epsilon) &= \widehat{\widetilde{\mathbf{q}}_9(\epsilon)} \\ &= F_t(\epsilon) + \sum_{i=7}^{8} \alpha_i(\epsilon, M)\widehat{\mathbf{T}'_i(\epsilon)} + O(\epsilon)m\end{aligned} \tag{2.44}$$

Utilizing the relation $\Theta R^T(\epsilon)(\Theta(\epsilon))^{-1} - I = O(\epsilon)$, we can modify Equation (2.44) such that

$$\begin{aligned}\widetilde{F}(\epsilon) &= F_t(\epsilon) + \sum_{i=7}^{8} \alpha_i(\epsilon, M)\widehat{\mathbf{T}'_i(\epsilon)} \left\{\Theta R^T(\epsilon)(\Theta(\epsilon))^{-1} + O(\epsilon)\right\} + O(\epsilon)m \\ &= F_a(\epsilon, M) + O(\epsilon)m\end{aligned}$$

where

$$\begin{aligned}F_a(\epsilon, M) &= F_t(\epsilon) + \left(\sum_{i=7}^{8} \alpha_i(\epsilon, M)\widehat{\mathbf{T}'_i(\epsilon)}\right) \Theta R^T(\epsilon)(\Theta(\epsilon))^{-1} \\ &= F_t(\epsilon) + O(\epsilon^0)m\end{aligned} \tag{2.45}$$

is a part of $\widetilde{F}(\epsilon)$ that contains the true rotation but an incorrect translation. As $\widetilde{F}(\epsilon)$ may lack the rank two property associated with a fundamental matrix, we apply the algorithm described in [69] that chooses a rank 2 matrix $\widetilde{F}'(\epsilon)$ with the minimum $\|\widetilde{F}(\epsilon) - \widetilde{F}'(\epsilon)\|$. If we consider $\widetilde{F}(\epsilon)$ to be a perturbed version of the valid fundamental matrix $F_a(\epsilon, M)$ (i.e., having rank 2), then from Equation (2.41),

$$\|\widetilde{F}(\epsilon) - \widetilde{F}'(\epsilon)\| \leq \|\widetilde{F}(\epsilon) - F_a(\epsilon, M)\| = O(\epsilon)m.$$

Hence, the error in $\widetilde{F}'(\epsilon)$ takes the form

$$\widetilde{F}'(\epsilon) = F_a(\epsilon, M) + O(\epsilon)m.$$

2.7.3 Errors in the Motion Estimates

The essential matrix $\widetilde{E}(\epsilon)$ is obtained by de-normalizing $\widetilde{F}'(\epsilon)$:

$$\widetilde{E}(\epsilon) = \Theta^T \widetilde{F}'(\epsilon)\Theta(\epsilon) = E_a(\epsilon, M) + O(\epsilon)m \tag{2.46}$$

where using Equations (2.2) , (2.43) and (2.45), we have

$$\begin{aligned} & E_a(\epsilon, M) \\ & = \Theta^T F_a(\epsilon, M)\Theta(\epsilon) \\ & = \left(\Theta^T \widehat{\mathbf{T}_t}\Theta + \det(\Theta) \sum_{i=7}^{8} \alpha_i(\epsilon, M) \widehat{\Theta^{-1}\mathbf{T}'_i}(\epsilon) \right) R^T(\epsilon) \\ & = E_t(\epsilon) + O(\epsilon^0)m. \end{aligned} \tag{2.47}$$

Observe that from Equation (2.47), $E_a(\epsilon)$ is a valid essential matrix (in the sense that it has rank 2 and two identical non-zero singular values), since it is a product of a skew symmetric matric and a rotation matrix $R(\epsilon)$ in $SO(3)$.

We can treat $\widetilde{E}(\epsilon)$ as a perturbed version of $E_a(\epsilon)$. Therefore, using the algorithm in [69] to enforce on $\widetilde{E}(\epsilon)$ the condition of having the first two singular values being equal, we can obtain a valid essential matrix $\widetilde{E}'(\epsilon)$, where from Equations (2.41) and (2.47) , we have

$$\|\widetilde{E}'(\epsilon) - \widetilde{E}(\epsilon)\| \leq \|E_a(\epsilon, M) - \widetilde{E}(\epsilon)\| = O(\epsilon)m. \tag{2.48}$$

Using triangle inequality and the orders from Equations (2.46) and (2.48), we have

$$\begin{aligned} &\|\widetilde{E}'(\epsilon) - E_a(\epsilon, M)\| \\ &\leq \|\widetilde{E}'(\epsilon) - \widetilde{E}(\epsilon)\| + \|E_a(\epsilon, M) - \widetilde{E}(\epsilon)\| \\ &= O(\epsilon)m. \end{aligned} \tag{2.49}$$

Similarly, using the orders from Equations (2.46) , (2.47) and (2.48), we have

$$\begin{aligned} &\|\widetilde{E}'(\epsilon) - E_t(\epsilon, M)\| \\ &\leq \|\widetilde{E}'(\epsilon) - \widetilde{E}(\epsilon)\| + \|E_t(\epsilon, M) - \widetilde{E}(\epsilon)\| \\ &= O(\epsilon^0)m. \end{aligned} \tag{2.50}$$

The rest of the proof basically keeps track of how the errors are propagated when one uses singular value decomposition on $\widetilde{E}'(\epsilon)$ to obtain the rotation and translation estimates. The steps are nontrivial but the arguments are straightforward. Interested readers can refer to Appendix A.2 for the details. In particular, using the order in Equation (2.50) and Proposition A.8 in Appendix A.2, we obtain:

$$\text{Error in the unit translational vector} = O(\epsilon^0)m.$$

With regards to rotation, if one considers $\widetilde{E}'(\epsilon)$ to be a perturbed version of $E_a(\epsilon)$ which contains the true rotation, then using the order in Equation (2.49) and Proposition A.10 in Appendix A.2, we obtain:

$$\text{Error in the rotational matrix} = O(\epsilon)m,$$

completing the requirements set out in Equations (2.40).

2.8 Simulation Results

We present simulation results for the following linear algorithms:

HN denotes the eight point algorithm using Hartley normalization [39],

HNC denotes the eight point algorithm with Hartley normalization and estimated by Total Least Squares – Fixed Column (TLS-FC) [74],

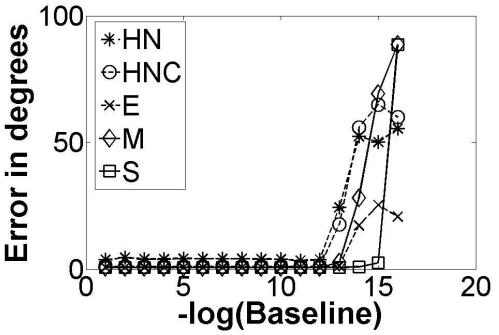
E denotes the un-normalized eight point algorithm [61],

M denotes the differential essential matrix [67], and

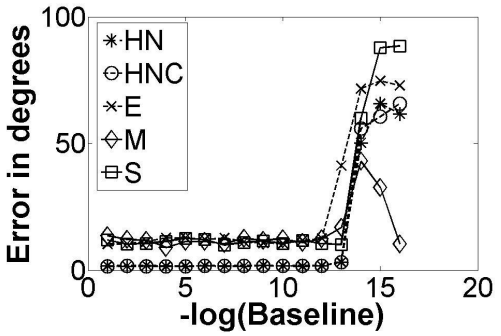
S denotes the linear subspace differential algorithm [41].

2.8.1 Decreasing Baseline

Simulation results for decreasing baseline are given in Figures 3.10 and 2.2, with those of the discrete algorithms represented by dotted lines and those of the differential algorithms by solid lines. The simulation conditions are as follows. The "scene" consisted of a point cloud containing 1000 points with an average depth of 10 units. The points were uniformly distributed between depths of 7 and 13 units. The simulated camera had a 45° field of view (FoV) with a focal length of 1 unit. The initial translation was set at $(0, 0.1, 0)$ unit, and the initial rotation at $(0.01, 0, 0.01)$ radians. Both the baseline and rotation were steadily decreased by

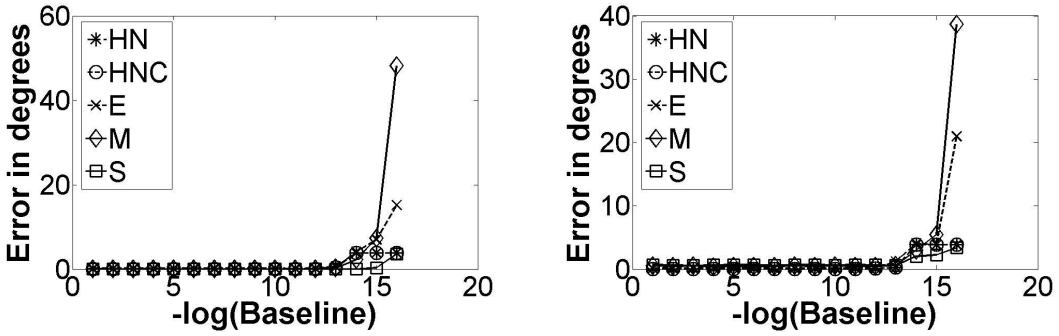


(a) Translation Direction=(0, 0, 1)



(b) Translation Direction=(1, 0, 0)

FIGURE 2.1: Error in estimating the translation direction with decreasing baseline. For lateral translation (b), the errors in E, M, and S are large.



(a) Translation Direction=(0, 0, 1)

(b) Translation Direction=(1, 0, 0)

FIGURE 2.2: Error in estimating the rotation with decreasing baseline.

factors of 10 to simulate increasingly small motion. This baseline was decreased until 10^{-16}, the limit of arithmetic precision, in order to verify our theoretical prediction. The optical flow noise was 3.5% of the average magnitude of the optical flow. The rotational errors presented in Figure 2.2 have been normalized such that

$$\text{normalized rotational error} = \frac{\text{rotational error in degrees}}{\text{baseline}}$$

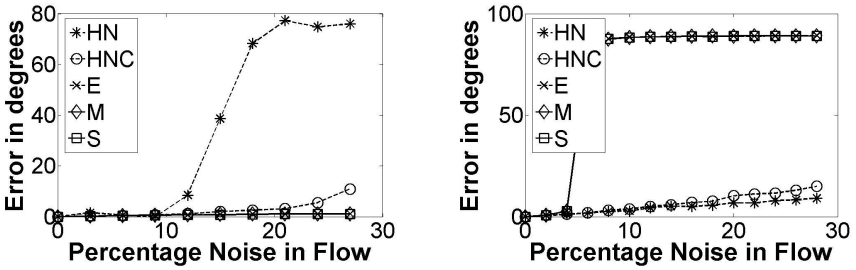
As such, a constant normalized rotation error in the graphs indicates that the actual error is decreasing proportionally to the amount moved by the camera.

2.8.2 Increasing Noise

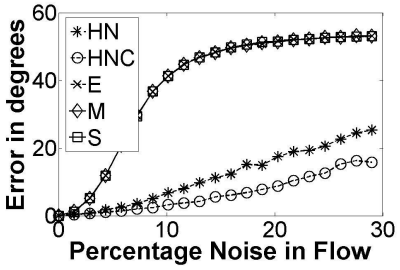
The scene is similar to that in Subsection 2.8.1. However, in this scenario, we fix the translation (see Figure 2.3 for the translation) and rotation while increasing the amount of noise. The results are presented in Figure 2.3, with each column representing different types of translational motions.

2.8.3 Observations

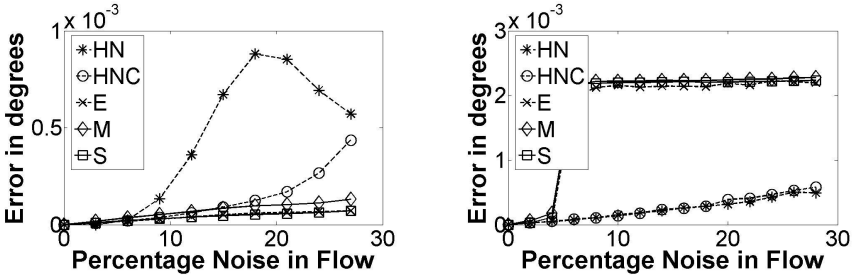
1. From Figures 3.10 and 2.2, one can see that there was no deterioration in the computation of the motion parameters using the discrete eight point algorithms (E, HN and HNC) despite reductions in the baseline to the limit of arithmetic precision. Note that the errors for the discrete algorithms shot up at about 10^{-12} or 10^{-13}: At this small baseline, the magnitude of the optical flow, being two to three orders of magnitude smaller than the baseline,



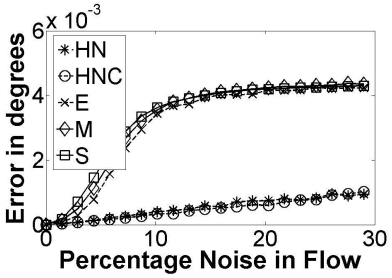
(a) Translation=(0, 0, 0.001) (b) Translation=(0.001, 0, 0)



(c) Translation=(0.001, 0.001, 0.001)



(d) Translation=(0, 0, 0.001) (e) Translation=(0.001, 0, 0)



(f) Translation=(0.001, 0.001, 0.001)

FIGURE 2.3: This figure illustrates the performances of various linear algorithms as the noise increases. The rotation parameters for all simulations in this figure is given by Rotation=(0 0 0). Figures (a) to (c) show the error in estimating the translation direction. Figures (d) to (f) show the error in estimating the rotation.

reached the limit of arithmetic precision, rendering the proportional noise model invalid and thus resulting in the breakdown of the discrete algorithms. This simulation clearly verifies that the theoretical predictions made in the previous section are correct.

2. The performances of the differential essential matrix algorithm (M) and the linear subspace algorithm (S) were extremely poor (Figures 2.1(b) and 2.3), especially in the lateral motion configuration which is susceptible to the bas-relief ambiguity [18, 22, 68, 109]. Their performances were more or less comparable to that of the un-normalized discrete approach (E). In contrast, the normalized discrete eight point algorithm with Total Least Squares – Fixed Column estimation (HNC) appeared to give very much superior results even when the motion was small, with HN's not far behind.

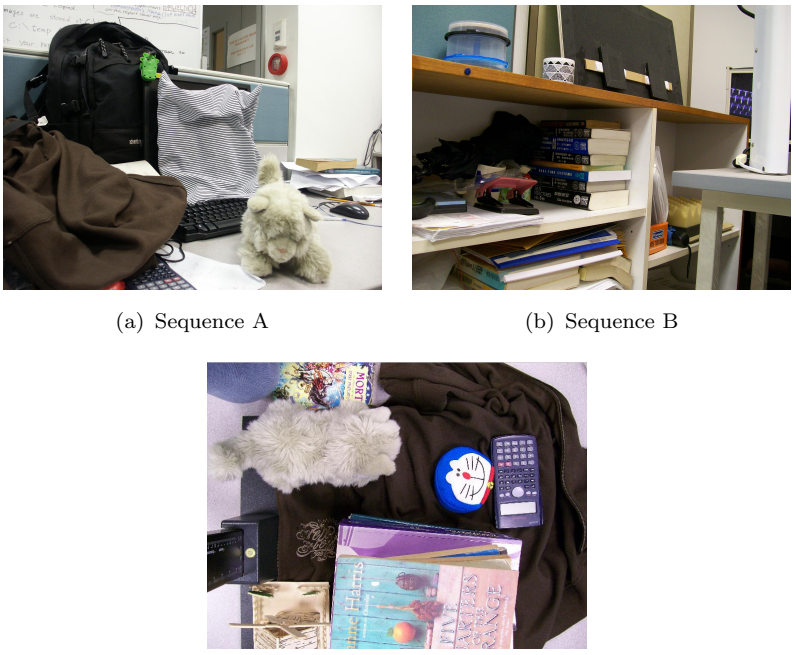
3. In Figure 2.2, the absolute rotational error declined proportionally with the baseline as predicted (i.e. the rotational error was of the order $O(\epsilon)m$).

4. Referring to Figure 2.3, the impact of noise was keenly felt for the un-normalized discrete approach (E) and the differential algorithms (M and S). Under all motion types tested, the well-known forward bias [18, 22, 68] reared its ugly head at even a low level of noise. For instance, in Figure 2.3(c), when the noise was about 10%, the forward-biased solution of 0° for the translation resulted in an error of about 45° (the true translation vector lies in the 45° direction). In the same token, for the forward translation case (Figure 2.3(a)), the excellent results of E, M, and S should be treated with caution. These algorithms had a strong forward bias and irrespective of

the true motion, tended to give a forward translation estimate whenever the noise was moderately large. On the other hand, the normalized discrete algorithms (HN and HNC) exhibited much less sensitivity to noise under all conditions tested, except in the translational estimate of HN under forward translation (Figure 2.3(a)) with the noise level greater than 10%. The results of this simulation imply that we could expect a stable performance from the discrete HNC algorithm when dealing with small motions, provided that the proportional noise in the optical flow computation is small enough.

2.9 Results on Real Image Sequences

With conventional CCD imaging technology and the mechanical stability of the measurement apparatus, it is clearly impossible to replicate with real image sequences the extremely small baseline scenario in the preceding section. Our goal in this section is to show that over a practical range of decreasing baselines, the normalized discrete algorithms can perform as well, if not better than the differential counterparts. The range of flow magnitude simulated is indicated in the first row of Table 2.1; our smallest baseline corresponds to the case where the average flow magnitude is of the order 10^{-1} pixel. This limit is reasonable as at the current technology level, the imaging noise expected for a high-quality, 12-bit, scientific imaging system may cause flow variation on the order of 0.01 pixels to 0.001 pixels, depending on the image content [96]. Such noise level would already constitute a 10% noise for a subpixel image motion of the order 10^{-1} pixel, which would be a problem for both the discrete and the differential algorithms.



(a) Sequence A (b) Sequence B

(c) Sequence C

FIGURE 2.4: Scenes that have been tested. Field of view ranges from 31 to 53°. Sequences A and B involve a pure lateral translation, while Sequence C involves a pure forward translation.

Three sequences were taken by moving a camera along a linear rail using two different consumer-grade cameras. For sequence A in Figure 2.4(a), the FoV was 31°. For sequences B and C in Figures 2.4(b) and 2.4(c), the FoV was 53°. Optical flow was estimated using the state-of-the-art algorithm provided by [91]. 4000 flows were obtained from sub-sampling the available flows and filtered using RANSAC to remove obvious outliers. In most scenes, 99% of the tested flows were considered inliers and different RANSAC trials gave little variation in the results. There was no scene-specific tuning of either the RANSAC thresholds or the parameters in the optical flow estimation algorithm. Here, we gave the average error over three trials. For computational efficiency, the number of flows were further reduced to 2000 by sub-sampling before being used for camera pose recovery. For comparison purpose, the camera pose estimated from all the linear algorithms was also refined

using the bundle adjustment algorithm [40]. The results of the linear algorithms are tabulated in the top half of Table 2.1, with the corresponding results refined by bundle adjustment reported in the bottom half of Table 2.1. As we have no means of accurately measuring the ground truth for small rotations, only the translational error is reported.

Sequences in Figures 2.4(a) and 2.4(b) involve a pure lateral translation, while that in Figure 2.4(c) involves a pure forward translation. Observe that the discrete linear estimator HNC of [74] performed much better than the differential estimator M from [67]. For example in the lateral motion sequences (Figures 2.4(a) and 2.4(b)), the discrete algorithm was able to give a good estimate even under circumstances in which its differential counterpart failed completely. These experimental results show clearly that for SfM problems involving a practical range of small motions, the normalized discrete linear algorithms out-performed their differential counterparts by a large margin, especially in lateral motion configuration which are liable to the bas-relief ambiguity. For forward translation (Figure 2.4(c)), the performance of the normalized discrete algorithms remained on par with the differential ones and was stable over decreasing baseline. We also note that random noises have apparently substantial effects on the performances of all algorithms, as can be seen from the non-smooth error figures over changing baseline in Table 2.1. This means that the subsequent step of bundle adjustment to refine the pose estimate is especially important. Given a normalized discrete algorithm that can provide an initial estimate stably over a large range of motion and over different motion configurations, the non-linear bundle adjustment step

would have a higher chance of finding the global minimum and will do so more quickly (see bottom half of Table 2.1).

2.10 Concluding remarks

We have proven that the eight point algorithm and its variants are "differential algorithms" in the sense that they can handle arbitrarily small motions given a sufficiently tight bound on the percentage noise. This proof was done using tools from matrix perturbation analysis. It shows that for a sufficiently small proportional noise, the eigenvalues of the data matrix remain separate and the solution vector can be recovered well even under very small motion. Using both real and simulation results, we have validated the theoretical analysis and shown that even under small motion, the normalized discrete eight point algorithms can perform well and indeed significantly outperform their differential counterparts. Given that much efforts have been spent in improving the discrete algorithms, and in view of our theoretical and experimental results, it seems that for now at

TABLE 2.1: Translation errors for sequences in Figure 2.4. (NL) in the bottom half of the table indicates that a nonlinear bundle adjustment step was used to refine the results obtained by the corresponding linear algorithm in the top half. The first row indicates the average magnitude of the optical flow for the sequence in that particular column.

Error (^{o})	A1	A2	A3	B1	B2	B3	C1	C2	C3
Flow Mag.	0.53	1.0	4.1	0.73	1.17	1.35	0.8	1.1	2.8
HNC	16.5	22.3	5.2	6.5	24.2	14.3	4.4	4.1	3.8
M	89.1	48.5	7.0	87.1	80.6	88.6	2.6	2.5	2.8
S	89.1	48.5	7.0	87.1	80.6	88.6	2.5	2.5	2.8
HNC (NL)	1.8	6.0	1.1	6.7	2.2	7.3	3.3	4.3	2.8
M (NL)	83.0	4.8	1.0	87.3	7.2	84.5	3.2	2.8	2.2
S (NL)	87.2	38.0	5.6	82.9	7.4	87.8	2.3	2.4	2.9

least, a properly normalized eight point algorithm should be used for SfM even in small motion.

Having obtained the theoretical results for the two-view SfM case, it would also be interesting to investigate whether the many so-called instabilities associated with small motion in various other problems are due to the instability of the specific discrete algorithms rather than the inherent sensitivity of small motion. For instance, [99] considered the case where the third view of a trifocal tensor is obtained by an infinitesimal change of a discrete two-view system. The additional constraint was obtained by differentiating the discrete epipolar constraint $\mathbf{p}^T E \mathbf{p}' = 0$ with both E and $\mathbf{p}'$ changing, which yields $\mathbf{p}^T E \dot{\mathbf{p}}' + \mathbf{p}^T \dot{E} \mathbf{p}' = 0$. While such formulation has the virtue of simplicity, the additional differential information $\mathbf{p}^T E \dot{\mathbf{p}}' + \mathbf{p}^T \dot{E} \mathbf{p}'$ can be drowned out when combined with the existing epipolar constraint $\mathbf{p}^T E \mathbf{p}'$, leading to apparent degeneracy under small changes in E and $\mathbf{p}'$. The problem is not inherently sensitive however; rather, a proper weighing and normalization scheme can do much to enhance the usefulness of the differential information and generally improve the stability of the algorithm. A full treatment of this question is beyond the scope of this thesis and presents a very interesting subject for future research.

Chapter 3

Simultaneous Camera Pose and Correspondence Estimation with Motion Coherence

Studying the interlocking relationship between applications and the correspondence recovery process can yield more than theoretical insights. In this chapter, we show how acknowledging the chicken and egg relationship between camera pose recovery and correspondence computation, permits the incorporation of non-unique edge points into the camera pose recovery process. This is achieved by fusing the camera pose recovery into a motion coherence matching framework.

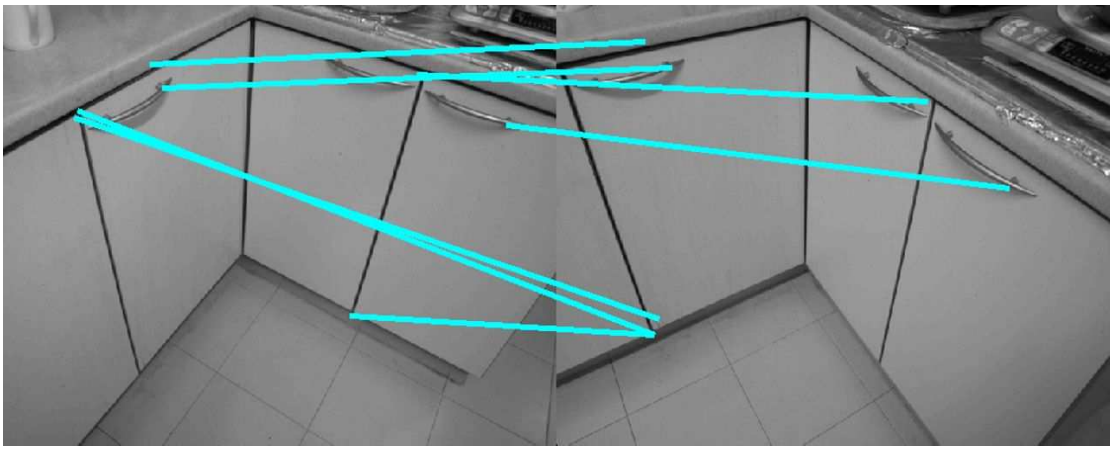


FIGURE 3.1: Illustrates the difficulty in obtaining reliable matches when there are few corners. Pose recovery on these scenes would be substantially easier if we could use the clear contour cues present.

3.1 Introduction

As mentioned in preceding chapters, a central issue that must be addressed in solving SfM is camera pose recovery. Traditionally, the camera pose recovery problem has been formulated as one of estimating the optimal camera pose given a set of point correspondences. Such approach includes, among many others, improved linear estimation [39, 82], bundle adjustment [100] as well as globally optimal estimators [28, 48]. However, despite many advances in matching techniques [7, 38, 64], obtaining correspondences across two images remains a non-trivial problem and contains a strong underlying assumption that the features are sufficiently distinct to enable unique point to point correspondence. This limits camera pose recovery to well textured scenes with abundant corner features. In this chapter, we seek to design an algorithm which can incorporate ambiguous features such as edge points into the camera pose recovery process. This allows pose recovery on more challenging SfM scenes where there are few corners; such scenes are particularly common

in man-made environment [107, 108], one example of which is illustrated in Figure 3.1. Our algorithm is, however, not limited to such scenes. Natural scenes where the visual features are highly similar or whose extraction is non-repeatable across large viewpoint change can also benefit from our approach.

While correspondence is needed to obtain camera pose, knowledge of camera pose also facilitates point correspondence. In recent years, a number of works [24, 33, 54, 70, 90] have proposed joint pose and correspondence algorithms (JPC) which explicitly acknowledge the chicken and egg nature of the pose and correspondence problem. Rather than choosing a camera pose in accordance with a pre-defined set of matches, these algorithms choose camera pose on the basis of whether the feature points can find a correspondence along the associated epipolar line. This permits the utilization of non-unique features to contribute to camera pose computation. Note that we should distinguish such JPC works from other joint estimation works such as 2D image or 3D surface registration [9, 27, 56, 111] using say, the Iterative Closest Point (ICP) technique. These registration works invariably involve a global transformation that is parameterized by a few variables (such as the affine parameters) and provides a well-defined mapping from point to point. This one-to-one mapping means the global parameters automatically preserves the relative alignment of features and largely accounts for the success in solving the registration. In contrast, in the JPC algorithms, the 3D camera pose does not define a point to point correspondence but rather a point to epipolar line relationship on the 2D image plane. This additional ambiguity means a much greater degree of freedom and associated problem complexity. More importantly

for our problem scenario where the features are highly ambiguous, it also means that the epipolar constraint alone is insufficient to resolve the ambiguity, even with the JPC approach. For example, if the feature points consisted of edge pixels that form a long connected contour, an epipolar line in any direction will eventually intersect with the contour. Thus, a JPC algorithm will have difficulty choosing a correct camera pose.

Despite such apparent ambiguity, we note that the motion-induced deformation of a 2D contours' shape contains clear perceptual cues as to the relative camera pose. One possible reason that humans can infer the camera pose might be that they perceive the contour points as a collective entity in motion (i.e. the law of shared common fate), rather than as independently moving individual points. This motivates us to impose a coherent motion constraint on the feature point displacements such that the displacements approximately preserve the overall shape of these points; in other words, points close to one another should move coherently [110].

While general non-rigid registration algorithms such as [21, 78] are generally able to preserve the overall shape of a point set, they are not designed for point-to-point correspondence and suffer from the aperture problem. As was shown in our preliminary work [58], individual contour points are poorly localized using the registration algorithm proposed in [78]. The registration is not consistent with any epipolar geometry and, hence, is not useful for obtaining camera pose.

In this chapter, we propose jointly estimating the camera pose and point correspondence while enforcing a coherent motion constraint. Such joint estimation

scheme is complex because the goodness of any point match depends not only on the camera pose and its local descriptor, but also on the matching position allocated to all other image points. The complexity is further increased because the smooth coherent motion of a contour is essentially a continuous concept, but we wish to work on discrete point sets containing possibly both corners and edge information. We adapt for this purpose the Coherent Point Drift framework of [78], which overlaid a continuous displacement field over the sparse point set, and regularized the displacement field to achieve motion coherence. The resultant scheme can compute camera pose using "ambiguous" features such as edge points (as well as the conventional corner points). It also removes the localization uncertainty of the edge point correspondence from using registration algorithm. This is illustrated in Figure 3.2. To our knowledge, this is the first attempt to integrate motion coherence, correspondence over a sparse point set and camera pose estimation into a common framework. The result makes a big difference in the perceived difficulty of a SfM scene. Our experiment showed that our algorithm can work well across large viewpoint changes, on scenes which primarily consist of long edges and few corners, as well as natural scenes with high visual clutter.

3.1.1 Related works

The core concept of using an iterative refinement of pose and correspondence has a long and rich history in SfM. Examples include RANSAC-flavored algorithms [31, 37, 75], and the Joint Pose and Correspondence/Flow algorithms [24, 54, 70,

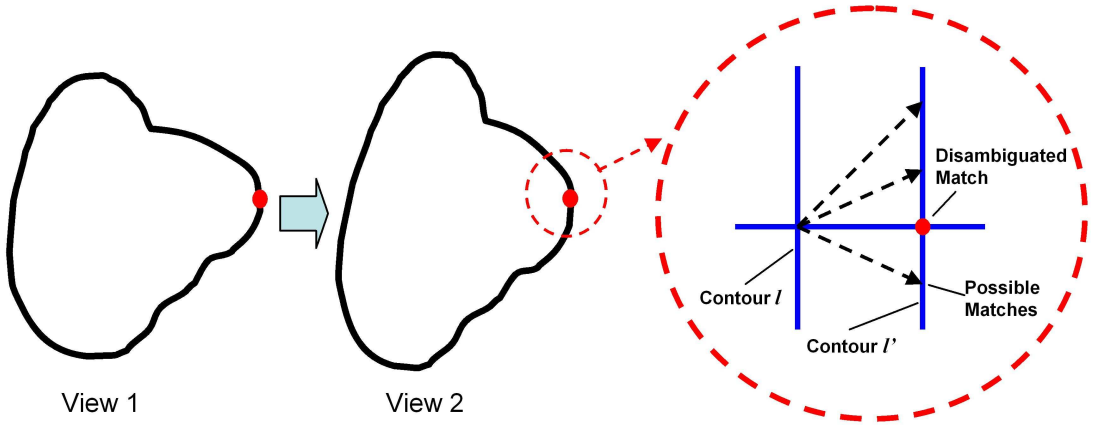


FIGURE 3.2: The dotted region represents the localization uncertainty present in the matching provided by the registration algorithm. On the right, the horizontal epipolar line allows the localization of the contour point.

84, 92, 103]. Many of these are landmark works which greatly improve SfM's stability in previously difficult scenes.

Of these, the JPC algorithms under small motion [92, 84, 103] are the ones which most closely resemble our algorithm. They overcome the aperture problem by finding an optical flow that is consistent with a camera motion. While this approach can be extended to wider baselines by applying a point set registration algorithm as initialization, such an algorithm would be inelegant and is likely to suffer from large amount of noise caused by approximating a large displacement by flow. Our approach handles large displacement naturally. It also handles the problem of disconnected point sets and isolated corners more naturally than that of optical flow formulation and would be especially useful in incorporating recently proposed edge descriptors [73]. Lastly, our approach can incorporate high-dimensional feature descriptors which give greater robustness to photometric noise.

There are many other works that jointly estimate a global transformation between

two sets of points and the point correspondence between them, but they differ from our work in some important aspects. Some of these involve multiple frames [26, 51, 77], where an initial 3D map was built from say, five-point stereo [82]. Subsequent camera poses were tracked using local bundle adjustment over the N most recent camera poses, and features are constantly added to allow the 3D map to grow in the SLAM style. In other works, the 3D models are available a priori (e.g. from a CAD model) [23, 50]. In contrast, our joint estimation is carried out over two frames in the absence of any 3D model or initial map. Other joint estimation works [9, 47, 56, 87, 111] involve aligning two sets of points which are related by some simple transformations defining a point to point mapping. The one to one mapping automatically preserves the relative alignment of the features within the point set without having a need for an additional coherence constraint. Our work differs in that the epipolar geometry does not enforce a one-to-one mapping. Instead, the unknown depth of the feature points means that the camera pose provides a point to epipolar line constraint. It also means that an additional coherence term is needed to enforce a greater coherence of shape, leading to a significantly more complex problem formulation.

For multiple views, it is also possible to make use of structure from lines algorithms to overcome the aperture problem [5, 29, 40, 93]. Interested readers might like to peruse other works dealing with various aspects of curve / line reconstruction [4, 17, 45, 94, 107, 108] as well as the merger of intensity and edge information [71, 85, 102].

3.2 Formulation

In this chapter, the problem addressed is the recovery of cameras' relative pose (i.e. orientation and position) given two different views of a static scene. The formulation emphasizes generality, allowing easy adaptation for different inputs such as corners and edges. Edges are simply described by point sets obtained by sampling the edge map of the image.

3.2.1 Definitions

Each feature point takes the form of a D dimensional feature vector,

$$\begin{bmatrix} x & y & r & g & b & \dots \end{bmatrix}^T_{1\times D},$$

with x and y being image coordinates, while the remaining optional dimensions can incorporate other local descriptors such as color, curvature, etc. We are given two point sets. A **base** point set $\mathbf{B_0}_{M\times D} = [b_{01}, ..., b_{0M}]^T$ describing M feature points in the base image and a **target** point set $\mathbf{T_0}_{N\times D} = [t_{01}, ..., t_{0N}]^T$ describing N feature points in the target image. b_{0i}, t_{0i} are D dimensional point vectors of the form given above.

We define another matrix $\mathbf{B}_{M\times D} = [b_1, ..., b_M]^T$ which is the evolved version of $\mathbf{B_0}$. We seek to evolve $\mathbf{B}$ until it is aligned to the target point set $\mathbf{T_0}_{N\times D}$, while still preserving the coherence of $\mathbf{B_0}$ (that is, the overall 2D geometric relationships between points in $\mathbf{B_0}$ should be preserved as much as possible). The evolution

of $\mathbf{B}$ consists of changing only the image coordinates (first two entries) of the b_i vectors. The remaining entries are held constant to reflect the brightness/ feature constancy assumption. When attempting to align the evolving base set $\mathbf{B}$ to the target set $\mathbf{T_0}$, we try to ensure that the resulting mapping of the image coordinates of b_{0i} to b_i are consistent with that of a moving camera viewing a static scene (i.e. abide by some epipolar constraint).

As many equations only involve the first two dimensions of b_{0i}, b_i, to simplify our notation, we define them as the sub-vectors β_{0i}, β_i respectively. We further denote the first two columns of $\mathbf{B}_0$ and $\mathbf{B}$ by $\mathfrak{B}_0$ and $\mathfrak{B}$, which are $M \times 2$ matrices formed by β_{0i} and β_i. As $\mathfrak{B}_\mathbf{0}$ and $\mathfrak{B}$ uniquely define $\mathbf{B_0}$ and $\mathbf{B}$ respectively in our case, the matrices can often be used interchangeably in probabilities and function declarations. The constancy of much of the b_i vector also means that the algorithm's run time is largely independent of the size of D. Hence one can apply high dimensional descriptors on the contour points with little additional cost.

3.2.2 Problem formulation

We seek an aligned base set $\mathbf{B}$ and the associated motion of an uncalibrated camera $\mathbf{F}$ (for calibrated cameras, one could parameterize $\mathbf{F}$ using the rotation and translation parameters without changing the formulation), which has maximum likelihood given the original base and target point sets $\mathbf{B_0}$ and $\mathbf{T_0}$ respectively. Mathematically, this can be expressed as maximizing $P(\mathbf{B}, \mathbf{F}|\mathbf{B_0}, \mathbf{T_0})$. Using Bayes' rule,

this can be formulated as,

$$\begin{aligned} P(\mathbf{B},\mathbf{F}|\mathbf{B_0},\mathbf{T_0}) &= \frac{P(\mathbf{T_0},\mathbf{B}|\mathbf{F},\mathbf{B_0})P(\mathbf{F},\mathbf{B_0})}{P(\mathbf{B_0},\mathbf{T_0})} \\ &= \frac{P(\mathbf{T_0},\mathbf{B}|\mathbf{F},\mathbf{B_0})P(\mathbf{F}|\mathbf{B_0})P(\mathbf{B_0})}{P(\mathbf{B_0},\mathbf{T_0})} \end{aligned}$$

It is clear that the likelihoods $P(\mathbf{B_0}), P(\mathbf{B_0},\mathbf{T_0})$ are constants with respect to the minimization variables $\mathbf{F},\mathbf{B}$. Furthermore, if we assume a uniform (uninformative) prior for the motion, it makes sense to assign $P(\mathbf{F}|\mathbf{B_0})$ to be a constant[1]. This allows us to simplify the probabilistic expression into

$$\begin{aligned} P(\mathbf{B},\mathbf{F}|\mathbf{B_0},\mathbf{T_0}) &\propto P(\mathbf{T_0},\mathbf{B}|\mathbf{F},\mathbf{B_0}) \\ &= P(\mathbf{B}|\mathbf{F},\mathbf{B_0})P(\mathbf{T_0}|\mathbf{B},\mathbf{F},\mathbf{B_0}). \end{aligned} \tag{3.1}$$

Observe that by expressing our formulation in terms of a warping from a base image to a target image, we treat the information from the two views in an asymmetrical manner. A symmetrical formulation may be able to better handle spurious feature and validate whether the algorithm has converged to an adequate minimum. However, the resultant scheme will be complex and is beyond the scope of the thesis.

We first study the term $P(\mathbf{B}|\mathbf{F},\mathbf{B_0})$. Given camera pose $\mathbf{F}$ and assuming independent isotropic Gaussian noise of standard deviation σ_b, the evolving base point

[1] An intuitive explanation for a uniform prior is that a camera can move to any position in the 3D world and similarly have any calibration parameters.

set $\mathbf{B}$ has an associated probability given by

$$P(\mathbf{B}|\mathbf{F}, \mathbf{B_0}) = P(\mathfrak{B}|\mathbf{F}, \mathbf{B_0}) = e^{-\lambda\Psi(\mathfrak{B})} \prod_{i=1}^{M} g(d_i, \sigma_b). \tag{3.2}$$

where $g(z, \sigma) = e^{-\frac{\|z\|^2}{2\sigma^2}}$ is a Gaussian function. We will explain the second term in more detail in Section 3.2.4 after we discuss the first term $e^{-\lambda\Psi(\mathfrak{B})}$.

3.2.3 Coherence term

The first exponent in equation (3.2) contains the regularization term $\Psi(\mathfrak{B})$ with λ controlling the relative importance of this regularization term.

Recall that we desire to enforce smoothness over a discrete point set whose points are sparsely distributed, a rather difficult operation to perform. One option is to directly penalize any deviation in the relative position of points considered as neighbors. Such an approach fits naturally into the discrete point set problem and is amenable to graph based minimization [20, 98]. However, because only the first order smoothness is imposed, it tends to penalize all deviations in relative position, rather than penalizing discontinuous changes in shape much more heavily than smooth deformation in shape caused by viewpoint changes. In other words, such first-order smoothness does not supply enough coherence of shape.

To overcome the aforementioned difficulties, we define a fictitious continuous field over the sparse point set and call it the displacement field or velocity field (in this chapter, the terms velocity and displacement are used loosely and do not imply any small motion approximation for the former). We utilize the motion coherence

framework of [110] in which higher order of smoothness is enforced on the velocity field. The smoothness is imposed mathematically by regularization in the Fourier domain of the velocity field. Our scheme has a number of advantages:

1. By imposing higher-order smoothness, it permits smooth changes in relative position that nevertheless maintains coherence in shape, rather than penalizing all changes. In fact, [110] explicitly showed that for isolated features, a smoothing operator with only first-order derivatives does not supply enough smoothness for a well-posed solution.

2. The formulation of this fictitious velocity field acts as a unifying principle for all types of motion information (isolated features, contours, brightness constancy constraint). It allows us to integrate the information provided by isolated features and contours, and yet does not require the declaration of a specific region of support when deciding which points are neighbors that should influence each others' motion.

3. While the interaction of the velocity field falls off with distance and is thus local, we obtain a resultant interaction between the isolated features that is nonlocal. This is desirable on account of the Gestalt principle. On the other hand, when there is local motion information that suggests discontinuous change in the velocity field, the rapidly falling off local interaction of the velocity field will ensure that it will be the locally measured data that are most respected, thus allowing discontinuous change in the velocity field. Preservation of such discontinuous changes is further aided by additional mechanisms introduced in the regularization scheme (more of that, later).

We define $v(.)$ as this 2D velocity field function. The velocity field covers the entire image, and at image locations β_{0i} where feature points exist, it must be consistent with the feature points' motion. Mathematically, this means that they obey the constraint

$$\beta_i = v(\beta_{0i}) + \beta_{0i}. \tag{3.3}$$

$\Psi(\mathfrak{B})$ is defined in the Fourier domain to regularize the smoothness of the velocity field function $v(.)$:

$$\Psi(\mathfrak{B}) = \min_{v'(s)} \left(\int_{\Re^2} \frac{|v'(s)|^2}{g'(s) + \kappa'(s)} ds \right), \tag{3.4}$$

where $v'(s)$ is the Fourier transform of the velocity field $v(.)$ which satisfies equation (4.1) and $g'(s)$ is the Fourier transform of a Gaussian smoothing function. The Gaussian function has a spatial standard deviation of γ which controls the amount of coherence desired of the velocity field. Without the $\kappa'(s)$ term, the above smoothness function follows the motion coherence form proposed in [110] and has been used in general regularization theory [35]; it was also subsequently adopted in the contour registration work of [78]. Such definition allows us to impose a continuous coherent motion field over the motion of a discrete point set specified by equation (4.1). Suppressing the high frequency components of the velocity field ensures that adjacent contour points have similar motion tendencies, thus preserving the overall 2D geometric relationships between points in $\mathbf{B_0}$. However, the Gaussian function drops off very sharply away from the mean, greatly penalizing the high frequency terms. In SfM where there may be occlusion and sharp velocity changes, such a penalty function can be overly restrictive. As such,

we introduce the additional $\kappa'(s)$ term, which should have limited spatial support and hence wide frequency support. In this chapter, spatial support is taken to be less than the smallest separation between any two points in $\mathbf{B_0}$. Given such limited spatial support, the exact form of the function κ is immaterial. We can just define:

$$\kappa(\beta_{0i} - \beta_{0j}) = \begin{cases} k, & i = j \\ 0, & i \neq j \end{cases} \tag{3.5}$$

where k is some pre-determined constant.

3.2.4 Epipolar term

The second term in equation (3.2) contains the epipolar constraint defined by camera pose, $\mathbf{F}$. As mentioned earlier, we desire that the image coordinate pairs β_{0i}, β_i, to be consistent with $\mathbf{F}$. Hence, d_i is the perpendicular distance of the point β_i from the epipolar line defined by point β_{0i} and pose $\mathbf{F}$, with a cap at ζ. Observe that since β_{0i} is a fixed point of unknown depth, d_i is the geometric error [40] associated with $\beta_{0i}, \beta_i, \mathbf{F}$, with an additional capping function. The capping function basically expresses the fact that the Gaussian noise error model is only valid for inlier points, while there exist a number of randomly distributed outlier points which result in much thicker tails than are commonly assumed by the Gaussian distribution.

Practically, such robust functions allow outliers to be removed from consideration by paying a certain fixed penalty. In this regards, its function is similar to statistical form of RANSAC [100]. Formally, the capped geometric distance can be

written as

$$d_i = \min(\left\| l_i^T(\beta_i - r_i) \right\|, \zeta) \tag{3.6}$$

where r_i is a two dimensional vector representing any point on the epipolar line. l_i is a two dimensional unit vector perpendicular to the epipolar line defined by $\mathbf{F}$ and β_{0i}. ζ is the maximum deviation of a point from the epipolar line, before it is considered an outlier. As our point sets often contain huge numbers of outliers, we usually set ζ to a very low value of 0.01 (the distance is defined in the normalized image space after Hartley's normalization [39]).

3.2.5 Registration term and overall cost function

We now consider $P(\mathbf{T_0}|\mathbf{B}, \mathbf{F}, \mathbf{B_0})$ in equation (3.1). Since $\mathbf{T_0}$ is independent of the ancestors $\mathbf{F}$ and $\mathbf{B_0}$ given the immediate parent $\mathbf{B}$, this probability can be simplified to just the confidence measure of $\mathbf{T_0}$ given $\mathbf{B}$. Note that the $\mathbf{T_0}$ and $\mathbf{B}$ contain a mix of descriptor and coordinate terms. We let each b_i be the D dimensional centroid of an equi-variant Gaussian function with standard deviation σ_t (we assume that the data has been pre-normalized, the normalization weights being given in section 3.3.3). The following forms the Gaussian mixture probability of $\mathbf{T_0}$:

$$P(\mathbf{T_0}|\mathbf{B}, \mathbf{F}, \mathbf{B_0}) = \prod_{j=1}^{N} \sum_{i=1}^{M} g(t_{0j} - b_i, \sigma_t). \tag{3.7}$$

This is the registration error term which includes both geometric and intensity information. Initially, $\mathbf{B}$ is not necessarily close to $\mathbf{T_0}$, thus making the above

probability very small. However, using the Expectation Maximization (EM) algorithm, we use these initial, low probabilities to better align $\mathbf{B}$ with $\mathbf{T_0}$. Note that we use the term EM loosely to describe the general minimization style although the exact mechanism is slightly unconventional.

Substituting equations (3.2) and (3.7) into (3.1) and taking the negative log of the resultant probability, our problem becomes one of finding the $\mathbf{F}$ and $\mathfrak{B}$ which maximize the probability in equation (3.1), or equivalently, minimize $A(\mathbf{B}, \mathbf{F})$, where

$$A(\mathbf{B}, \mathbf{F}) = -\sum_{j=1}^{N} log \sum_{i=1}^{M} g(t_{0j} - b_i, \sigma_t) + \sum_{i=1}^{M} \frac{d_i^2}{2\sigma_b^2} + \lambda\Psi(\mathfrak{B}). \tag{3.8}$$

The first term in $A(\mathbf{B}, \mathbf{F})$ measures how well the evolving point set $\mathbf{B}$ is registered to the target point set $\mathbf{T_0}$. The second term measures whether the evolving point set $\mathbf{B}$ adheres to the epipolar constraint. Finally, the third term ensures that the point set $\mathbf{B}$ evolves in a manner that approximately preserves the coherence of $\mathbf{B_0}$.

3.3 Joint estimation of correspondence and pose

We seek the $\mathfrak{B}$ and $\mathbf{F}$ which optimize equation (4.6) (recall that $\mathfrak{B}$ is the first two columns of $\mathbf{B}$). Observe that this is a constrained minimization but as the l_i, r_i terms in the geometric distance d_i have a non-linear relationship with the camera pose $\mathbf{F}$ and image point β_{0i}, as well as due to the presence of the regularization

term, it precludes other more straightforward minimization techniques. Using a method similar to expectation maximization, we minimize $A(\mathbf{B}, \mathbf{F})$ by alternately updating $\mathfrak{B}$ and $\mathbf{F}$. The procedure is described in the following subsections.

3.3.1 Updating registration, $\mathfrak{B}$

In this subsection, we hold the camera pose $\mathbf{F}^{old}$ constant while updating $\mathfrak{B}$. This results in a $\mathfrak{B}^{new}$ whose associated evolving base point set $\mathbf{B}^{new}$ is better aligned to the target point set $\mathbf{T_0}$, while preserving the point set's coherence and respecting the epipolar lines defined by the camera pose $\mathbf{F}^{old}$. The new registration $\mathfrak{B}^{new}$ can be computed from the $M \times 2$ linear equations in equation (C.2).

Here we provide the derivations. We define

$$\begin{aligned} \phi_{ij}(b_i, t_{0j}) &= g(t_{0j} - b_i, \sigma_t) \\ \overline{\phi_{ij}}(\mathbf{B}, t_{0j}) &= \frac{\phi_{ij}(b_i, t_{0j})}{\sum_z \phi_{zj}(b_z, t_{0j})}. \end{aligned} \tag{3.9}$$

Fo more robust correspondence with occlusion, we use a robust version of $\overline{\phi_{ij}}(\mathbf{B}, t_{0j})$ in equtaion (C.1). This is given by $\overline{\phi_{ij}}(\mathbf{B}, t_{0j}) = \frac{\phi_{ij}(b_i, t_{0j})}{\sum_z \phi_{zj}(b_z, t_{0j}) + 2\mu\pi\sigma_t^2}$. The second, $2\mu\pi\sigma_t^2$ denominator term provides a thickening of the tail compared to those of the Gaussian. The idea is similar to the robust implementation of the regularization in equation (3.6).

Using Jensen's inequality and observing that the maximum value of d_i is ζ, we can write the inequality

$$\begin{aligned} & A(\mathbf{B}^{new}, \mathbf{F}^{old}) - A(\mathbf{B}^{old}, \mathbf{F}^{old}) \\ \leq & -\sum_{j=1}^{N}\sum_{i=1}^{M} \overline{\phi_{ij}}(\mathbf{B}^{old}, t_{0j}) log \frac{\phi_{ij}(b_i^{new}, t_{0j})}{\phi_{ij}(b_i^{old}, t_{0j})} \\ & + \sum_{i \in inlier} \frac{(d_i^{new})^2 - (d_i^{old})^2}{2\sigma_b^2} \\ & + \lambda \left(\Psi(\mathfrak{B}^{new}) - \Psi(\mathfrak{B}^{old}) \right) \\ = & \Delta A(\mathbf{B}^{new}, \mathbf{B}^{old}, \mathbf{F}^{old}). \end{aligned} \tag{3.10}$$

where a point i is an inlier if $d_i^{old} < \zeta$.

Observing from equation (3.10) that $\Delta A(\mathbf{B}^{old}, \mathbf{B}^{old}, \mathbf{F}^{old}) = 0$, the $\mathbf{B}^{new}$ which minimizes $\Delta A(\mathbf{B}^{new}, \mathbf{B}^{old}, \mathbf{F}^{old})$ will ensure that

$$A(\mathbf{B}^{new}, \mathbf{F}^{old}) \leq A(\mathbf{B}^{old}, \mathbf{F}^{old})$$

since the worst $A(\mathbf{B}^{new}, \mathbf{F}^{old})$ can do is to take on the value of $A(\mathbf{B}^{old}, \mathbf{F}^{old})$.

Dropping all the terms in $\Delta A(\mathbf{B}^{new}, \mathbf{B}^{old}, \mathbf{F}^{old})$ which are independent of $\mathfrak{B}^{new}$, we obtain a simplified cost function

$$\begin{aligned} Q = & \frac{1}{2} \sum_{j=1}^{N}\sum_{i=1}^{M} \overline{\phi_{ij}}(\mathbf{B}^{old}, t_{0j}) \frac{\|t_{0j} - b_i^{new}\|^2}{\sigma_t^2} \\ & + \sum_{i \in inlier} \frac{(d_i^{new})^2}{2\sigma_b^2} + \lambda \Psi(\mathfrak{B}^{new}). \end{aligned} \tag{3.11}$$

Using a proof similar to that in [78], we show in the Appendix that the regularization term $\Psi(\mathfrak{B})$ at the minima of $A(\mathbf{B}, \mathbf{F})$ is related to $\mathfrak{B}$ and $\mathfrak{B_0}$ by

$$\Psi(\mathfrak{B}) = tr(\Gamma \mathbf{G}^{-1} \Gamma^T), \tag{3.12}$$

where $\mathbf{G}$ is a $M \times M$ matrix with its (i, j) entry given by $\mathbf{G}(i,j) = g(\beta_{0i} - \beta_{0j}, \gamma) + k\delta_{ij}$ (δ_{ij} being the Kronecker delta), $\Gamma = (\mathfrak{B} - \mathfrak{B_0})^T$, and $tr(.)$ represents the trace of a matrix. Substituting the above expression of $\Psi(\mathfrak{B})$ into Q and taking partial differentiation of Q with respect to each element of $\mathfrak{B}^{new}$, we can construct the matrix $\frac{\partial Q}{\partial \mathfrak{B}^{new}}$, where each entry is $\frac{\partial Q}{\partial \mathfrak{B}^{new}(i,j)}$. The conditions needed for achieving the minimum of Q can be obtained by setting all the entries of this matrix to zero:

$$\begin{aligned} \frac{\partial Q}{\partial \mathfrak{B}^{new}} = \begin{bmatrix} c_1 & c_2 & \dots & c_{M-1} & c_M \end{bmatrix} + 2\lambda \Gamma^{new} \mathbf{G}^{-1} &= \mathbf{0}_{2\times M} \\ \mathbf{C} + 2\lambda \Gamma^{new} \mathbf{G}^{-1} &= \mathbf{0}_{2\times M} \\ \mathbf{CG} + 2\lambda \Gamma^{new} &= \mathbf{0}_{2\times M} \end{aligned} \tag{3.13}$$

Here, the column vector c_i is computed as

$$\begin{aligned} c_i = \sum_{j=1}^{N} \overline{\phi_{ij}}(\mathbf{B}^{old}, t_{0j}) \left(\frac{\beta_i^{new} - \hat{t}_{0j}}{\sigma_t^2} \right) \\ + \begin{cases} \frac{\mathbf{q_i}^{old}(\beta_i^{new} - r_i^{old})}{\sigma_b^2} & i \in \text{inlier} \\ \mathbf{0}_{2\times 1} & \text{otherwise} \end{cases}, \end{aligned}$$

where $\mathbf{q_i}_{2\times 2}$ is a 2×2 matrix given by $\mathbf{q_i}_{2\times 2} = (l_i)(l_i^T)$, $\hat{t}_{0j}$ stands for the truncated vector of t_{0j} with the latter's first two elements, and the definitions of l_i, r_i are as given in equation (3.6). Equation (C.2) produces $M \times 2$ linear equations which

can be solved to obtain $\mathfrak{B}^{new}$.

Observe that the minimization step in equation (C.2)—in particular, the computation of c_i—is in keeping with the spirit of the outlier rejection scheme discussed in equation (3.6): "outliers" are no longer over-penalized by the camera pose but they remain incorporated into the overall registration framework.

3.3.2 Updating camera pose, F

We now update the camera pose on the basis of the new correspondence set $\mathfrak{B}^{new}$,$\mathfrak{B}_{\mathbf{0}}$. Replacing $\mathbf{B}$ in equation (4.6) with $\mathbf{B}^{new}$ and holding it constant, we seek to minimize the cost function $A(\mathbf{B}^{new}, \mathbf{F}^{new})$ with respect to only $\mathbf{F}^{new}$. Only the middle term in $A(\mathbf{B}, \mathbf{F})$ depends on $\mathbf{F}$. Using the definition of the geometric distance d_i in equation (3.6), we minimize the simplified cost function

$$\sum_{i=1}^{M} \min\left(\left\|(l_i^{new})^T(\beta_i^{new} - r_i^{new})\right\|^2, \zeta^2\right) \tag{3.14}$$

with β_i^{new} being the image coordinates of the point set $\mathfrak{B}^{new}$.

Observe that the problem of finding the $\mathbf{F}^{new}$ which in turn produces l_i^{new} and r_i^{new} that minimize the above cost function can be formulated as a bundle adjustment problem [100] with camera pose $\mathbf{F}$ initialized to $\mathbf{F}^{old}$.

After these two steps, $\mathfrak{B}^{old}$, $\mathbf{F}^{old}$ are replaced with $\mathfrak{B}^{new}$, $\mathbf{F}^{new}$ and the algorithm returns to the first step in section 3.3.1. The process is iterated until convergence as the evolving base set $\mathbf{B}$ registers itself to the target set $\mathbf{T}_{\mathbf{0}}$.

3.3.3 Initialization and iteration

Hartley normalization is performed on the image coordinates of both point sets, thus pre-registering their centroids and setting the image coordinates to have unit variance. In this chapter, SIFT [64] feature descriptors were also attached to the points. These descriptors are normalized to have magnitudes of σ_t of equation (3.7).

For initialization of the correspondence, we use SIFT flow [59] to give initial values of $\mathfrak{B}^{new}$. However, SIFT flow is not used to initialize the camera pose. As can be seen from equation (4.6), setting l_i to zero for the first EM iteration will cause the algorithm to ignore the epipolar constraint during this first iteration. Once $\mathfrak{B}^{new}$ is calculated, $\mathbf{F}^{new}$ can be calculated from $\mathfrak{B}^{new}$ and $\mathfrak{B}_{\mathbf{0}}$, after which $\mathfrak{B}^{old}, \mathbf{F}^{old}$ are replaced with $\mathfrak{B}^{new}, \mathbf{F}^{new}$. Normal EM resumes with l_i restored, and the process is iterated until convergence.

For stability, we set σ_t, σ_b to artificially large values, then steadily anneal them smaller. This corresponds to the increased accuracy expected of the camera pose estimate and the point correspondence. A summary of the algorithm is given in figure 3.3.

3.4 System implementation

In this section, we consider how one might build a complete SfM system using our proposed joint estimation framework. To do this, we must address issues such

Input: Point sets, $\mathbf{B_0}$, $\mathbf{T_0}$

Initialize σ_t, σ_b;
Initialize $\mathfrak{B}^{old}$ as $\mathfrak{B}_\mathbf{0}$, l_i to zero vector;
while σ_t, σ_b *above threshold* **do**
  while *No convergence* **do**
    Use eqn (C.1) to evaluate $\phi_{ij}(b_i^{old}, t_{0j})$ from $\mathfrak{B}^{old}$, $\mathbf{F}^{old}$;
    Use eqn (C.2) to determine $\mathfrak{B}^{new}$ from $\phi_{ij}(b_i^{old}, t_{0j})$;
    Use bundle adjustment to obtain $\mathbf{F}^{new}$ from $\mathfrak{B}^{new}$ and $\mathfrak{B}_\mathbf{0}$;
    Replace $\mathfrak{B}^{old}, \mathbf{F}^{old}$ with $\mathfrak{B}^{new}, \mathbf{F}^{new}$;
  end
  Anneal $\sigma_t = \alpha\sigma_t$, $\sigma_b = \alpha\sigma_b$, where $\alpha = 0.97$.
end

FIGURE 3.3: Algorithm to register point sets $\mathbf{B_0}$ to $\mathbf{T_0}$, while computing the camera pose in $\mathbf{F}$

as point set acquisition, occlusion detection and initialization under real world conditions.

The first step of any such system has to be the identification of point sets in both images. As our algorithm is capable of utilizing non-unique features such as edges, we do not wish to use a corner detector, which would reject all edge-like features. Edge detectors would provide edge information; however, they often detect many spurious edges [102]. In order to overcome these problems, we detect features following the seminal SIFT algorithm [64]. However, as we are not interested in uniqueness, we disabled the cornerness term which otherwise would remove feature points that are considered too edge-like. The result appears to resemble that of a rather sparse but robust edge detector as illustrated in figure 3.4 but will also provide corner information when available. The descriptors that come with the SIFT detector also contribute greatly to stability.

The next issue is one of initialization and occlusion detection. What we need at this stage is not a well localized image registration but a crude initialization and

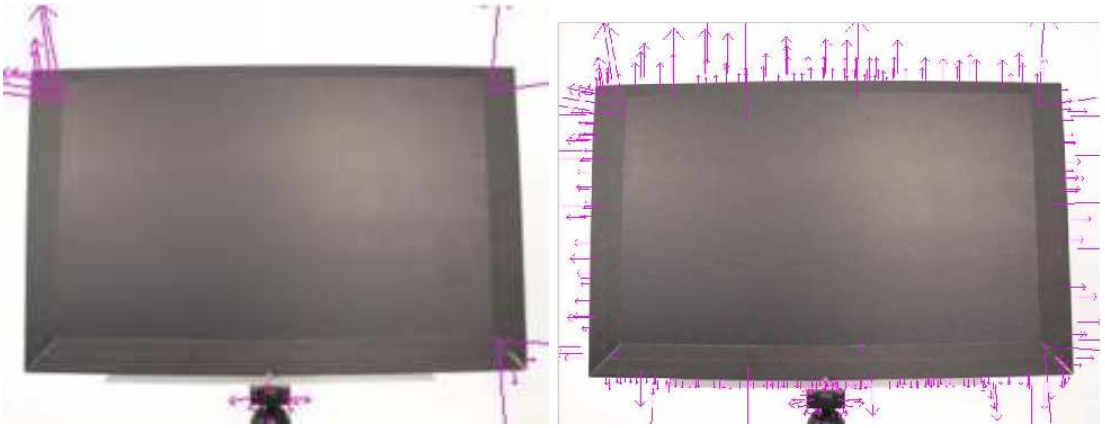


FIGURE 3.4: Left to right: Output of SIFT feature detector with and without its cornerness function.

a general idea of which sections of the image are occluded (feature points in the occluded regions need to be removed from the point sets $\mathbf{B_0}$ and $\mathbf{T_0}$). For these purposes, we utilize the dense SIFT flow algorithm to give us a crude mapping. Occluded regions are defined as regions where the SIFT flow is inconsistent, i.e. point A in image 1 maps to point B in image 2, however, point B does not map back to anywhere near point A. At very large baselines, the occlusion detector may declare the entire image as occlusion. In such situations the occlusion mask is discarded. (Note that a more sophisticated form of occlusion detection can be obtained in [8].)

Finally, one can obtain a dense $3D$ reconstruction using the computed camera pose and the Patched based Multi-view Stereo (PMVS) [32] as a dense matcher. The complete pipeline is shown in figure 3.5.

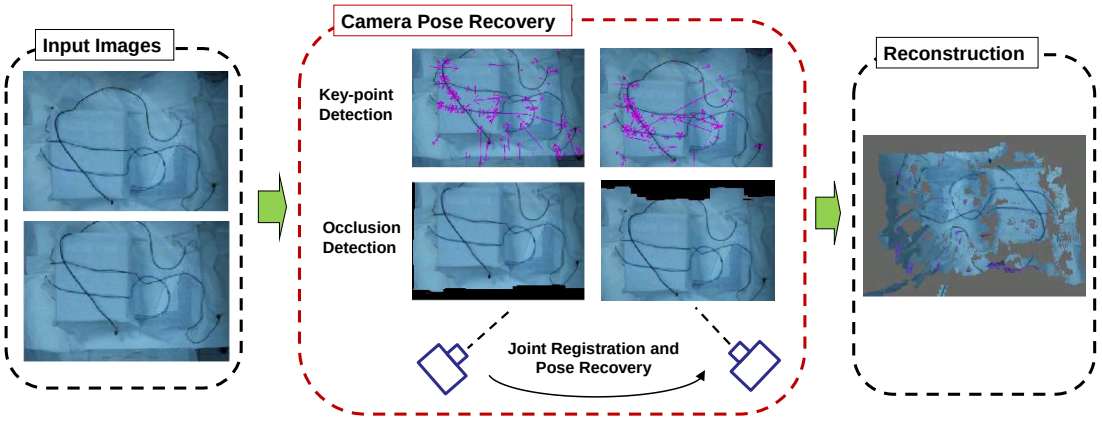


FIGURE 3.5: 3-D reconstruction pipeline. Left to right: Side view of setup, input images, key-point detection, occlusion detection (with occluded pixels set to zero) and final reconstruction obtained from camera pose.

3.5 Experiments and Evaluation

We run a series of real and simulated experiments to evaluate our algorithm, with errors reported as deviations from ground truth rotation and translation. All parameters reported are with respect to the Hartley normalized coordinates. All images are evaluated at a resolution of 640×480.

The rotational error $\widetilde{R}$ refers to the rotation angle in degree needed to align the reference frame of the computed pose to that of the true pose. The translational error $\widetilde{T}$ is the angle in degree between the computed translation and the ground truth translation. Although both the rotational and translational errors are given in degrees, in general, for typical camera and scene configuration, a large rotational error is more serious than a translational error of similar magnitude.

We test our system on a wide range of scene types and baselines. These include many "non-traditional" SfM scenes in which there are few/ no distinct corners available for matching, such as natural vegetation scenes where there is a large

amount of self occlusion and thus spurious corners, architectural scenes where the available corners are very repetitive as well as more traditional SfM scenes. This is followed by a systematic evaluation of our algorithm's handling of increasing baseline. For most scenes, ground truth camera pose is obtained by manually obtaining point correspondences until the computed camera pose is stable. An exception is made for the last two images in figure 3.7, where the extremely textureless scenes were taken using linear rail with known motion. A calibrated camera was used for all these tests.

To give the reader a general feel for the scenes' difficulty, our results are benchmarked against that of a traditional SfM technique. Correspondences are obtained using [64]. Camera pose is obtained using the five point algorithm [82] together with outlier removal by the RANSAC implementation in [53], the outliers rejection threshold being set at a Sampson distance of 0.001. The RANSAC step is followed by a bundle adjustment using the implementation of [63] to minimize the reprojection error.

The same set of parameters are used throughout the entire experiments. The two Gaussian parameters σ_b and σ_t in equations (3.2) and (3.7) are given an initial value of $\sigma_t = \sigma_b = 0.1$. They are annealed to smaller values with annealing parameter $\alpha = 0.97$. The occlusion handling parameter μ in equation (C.1) is set to 0.5, while the epipolar outlier handling parameter ζ in equation (3.6) is set to 0.01. λ, which controls the relative weight given to the smoothness function, is set to 1. k, the degree of tolerance for high frequency components in equation (3.5),

was set to 0.0001, while γ, the standard deviation of the Gaussian smoothness function, was set to 1.

3.5.1 Evaluation

We evaluate our algorithm on a variety of real and simulated scenes. In the simulated scene in figure 3.6, we illustrate our system's performance over depth discontinuities and the role of the discontinuity parameter k in equation (3.5). It shows that our algorithm can handle depth discontinuities and the pose computed is robust to the smoothness perturbations that the discontinuities induce. This is also illustrated in a number of real images of trees in figure 3.8 and a bicycle scene in figure 3.9. For the outdoor scenes, the baseline is usually a few meters. For the indoor scenes where objects are closer to the camera, the baseline is typically half a meter.

In figure 3.7, we investigate real images of scenes with sparsely distributed corners. Errors in the recovered camera parameters are reported below the images. "Ours" indicates the errors obtained by our algorithm, "SIFT flow" those obtained by running the five point algorithm and bundle adjustment on SIFT flow as correspondence input and finally, "Traditional" those obtained by running the five point algorithm with RANSAC and bundle adjustment on SIFT matches as correspondence input (traditional here refers to the dependence on unique features such as corners). In some scenes, SIFT matching returns too few matches for the traditional algorithm to give a pose estimate. In such circumstances, the pose error is given as Not Applicable (NA). The first two test images are of buildings. As

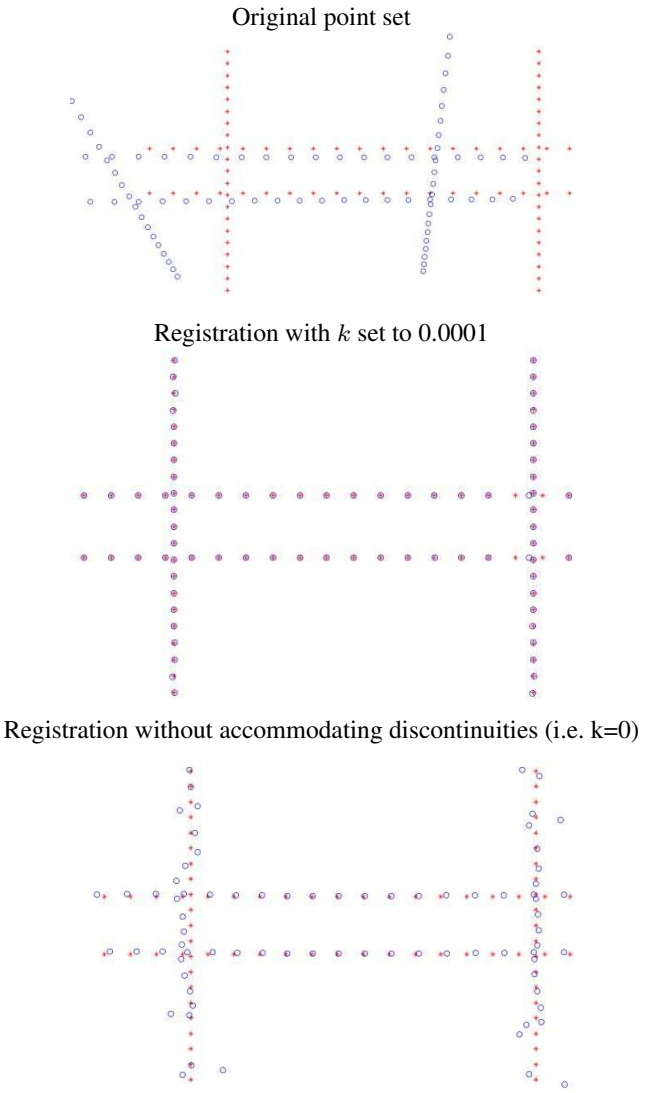


FIGURE 3.6: The vertical bars have a different depth and color (color not shown in results) from the horizontal bars. As the camera moves, the depth discontinuity causes the vertical bars to slide over the horizontal one. Setting the high frequency tolerance parameter k to 0.0001, the system retains both the smoothness constraint while accommodating the discontinuities. While there are some correspondence errors, our system is sufficiently robust to ensure that there is negligible error in the overall pose estimation. Using the standard motion coherence, where $k = 0$, the conflict between registration, smoothness and epipolar geometry cannot be resolved. The resultant pose estimate suffers, with a translational and rotation errors of 13.5^o and 5^o respectively.

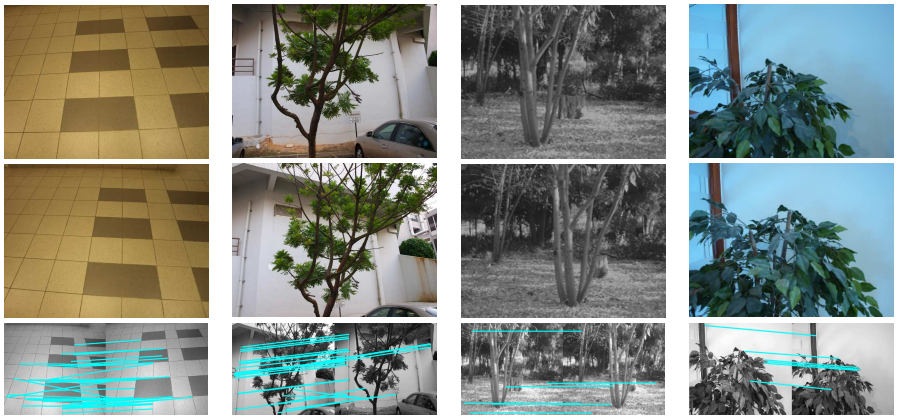
in many man-made structures, lines and edges are the predominant cues present. The problem of identifying matches needed for traditional SfM is compounded by the wide baseline. By relaxing the uniqueness requirement, our algorithm can utilize a much greater amount of information compared to the traditional approach, leading to a stable camera pose recovery. The third and fourth scenes consist of extremely sparsely distributed sets of corners. Here the primary SfM cue is the edge information. Our algorithm can utilize this edge information to convert an information-impoverished scene with very few point matches into an information-rich scene. This allows it to circumvent the difficulties faced by the traditional SfM algorithms.

	$\widetilde{R}$	$\widetilde{T}$	$\widetilde{R}$	$\widetilde{T}$	$\widetilde{R}$	$\widetilde{T}$	$\widetilde{R}$	$\widetilde{T}$
SIFT flow	15.9177	32.7802	4.8950	5.5900	13.589	81.2057	2.0970	27.8590
Ours	0.8944	2.8339	4.2943	7.5999	1.6161	12.5875	0.9923	11.5929
Traditional	14.5585	78.5412	12.8954	9.3199	15.0885	57.1229	17.7057	80.8453

FIGURE 3.7: We show a number of scenes where there are few corners and correspondingly few matches. The correspondences obtained from SIFT matching [64] are shown in the third row. The matches that exist are also poorly distributed, with the majority of matches being clustered in a small region. The fourth row shows the SIFT points used by our algorithm. By relaxing the need for unique correspondence, we can use a much richer and better distributed point set, which in turn permits a better recovery of the camera pose. The pose errors are reported below the images (with $\widetilde{R}$ and $\widetilde{T}$ representing rotation and translation errors in degrees, see text for elaboration).

In figure 3.8, we further our investigation on scenes which contain a large number of non-unique corners. This is true for the floor image, where the gird pattern tiling results in multiple corners with nearly identical feature descriptor. It also occurs in natural vegetation scenes, where the leaves form many repetitive features. For plants, the problem is made more severe because the extensive self occlusion caused by the interlocking of leaves and branches further degrades potential corner descriptors. Hence, despite the large number of corners available (nearly 1000 for some of the images), there are few SIFT matches on the foliage. For the floor scene, jointly estimating the correspondence and pose allows the handling of non-unique features and the subsequent pose recovery. For the plant images, our algorithm can ignore the noise in the degraded feature descriptors and utilize the tree trunks and their outlines to obtain a camera pose estimate. We also illustrate a failure case in the last column of figure 3.8. With most of the feature descriptors badly perturbed by self occlusion, the primary SfM cue lies in the edge information which in this case is the extremal contour of the plant. Unlike polyhedral objects, the extremal contour of the plant is view-dependent (i.e. the points on the plants that participate in generating the contour are view-dependent). This dependency effect is especially significant when the displacement is quite large (at smaller displacements our algorithm can handle this scene).

Finally, in figure 3.9 we evaluate our algorithm on traditional SfM scenes with adequate number of unique features. This shows that our algorithm also works well when the primary cue lies in disconnected but discriminative corner information. Although some scenes contain significant depth discontinuities, our algorithm can



	$\widetilde{R}$	$\widetilde{T}$	$\widetilde{R}$	$\widetilde{T}$	$\widetilde{R}$	$\widetilde{T}$	$\widetilde{R}$	$\widetilde{T}$
SIFT flow	2.0097	27.8590	7.6931	3.6912	1.3447	11.5193	15.3635	88.9147
Ours	1.2401	8.6582	1.5239	13.5200	0.8103	9.3550	48.6037	35.5356
Traditional	9.3170	56.4661	6.5460	86.6666	66.5418	89.6218	NA	NA

FIGURE 3.8: Here we experiment on images where corners are plentiful (some of the tree images have over 1000 features detected) but unique matching remains challenging. This lack of uniqueness is due to the strong repetitive pattern. For the plant images, the problem is compounded by the interlocking leaves which induce self-occlusion and corresponding feature degradation. For the floor image, our algorithm can utilize the non-unique SIFT feature to recover camera pose, while for the tree images, we can utilize the features lying along the trees branches. The final image shows a failure case where the stem is hidden by the foliage and the problem is further compounded by a view-dependent extremal contour.

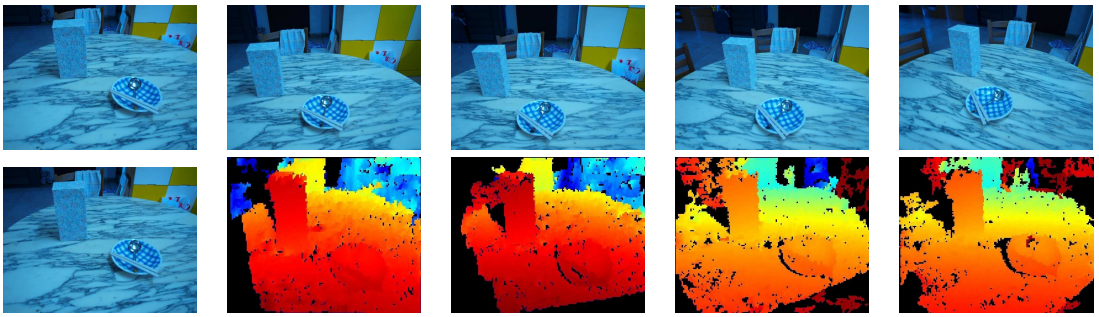
produce the same, if not better, accuracy in the camera pose estimate when compared to the results of the traditional SfM algorithms.

	$\widetilde{R}$	$\widetilde{T}$	$\widetilde{R}$	$\widetilde{T}$	$\widetilde{R}$	$\widetilde{T}$
SIFT flow	0.43	5.52	1.07	3.94	2.05	1.59
Ours	0.41	3.38	0.76	2.68	0.71	1.90
Traditional	0.31	5.20	0.94	6.20	1.26	3.93

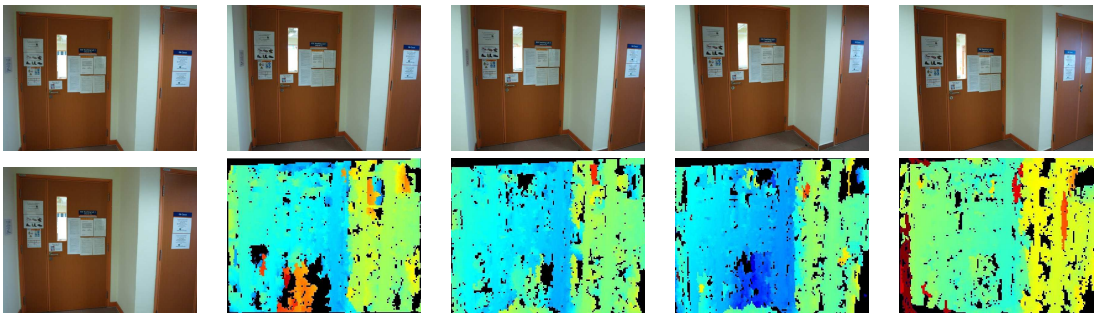
FIGURE 3.9: A set of more traditional structure from motion scenes. Observe that our algorithm performs well on these easy, well-textured structure from motion scenes.

3.5.2 Performance with increasing baseline

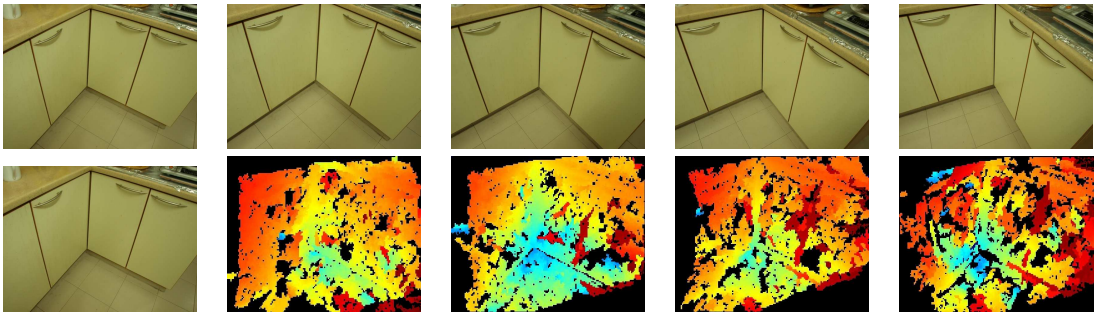
In figure 3.10, we investigate our algorithm's behavior with increasing baseline. The sequences consist of a moving camera fixated upon a scene and are arranged in increasing baseline and thus level of difficulty. The color-coded depth maps obtained by reconstructing the scene using PMVS [32] are also included. The first sequence is a traditional, well textured SfM scene. The baseline is fairly large, with the camera rotating through 33.9 degrees while fixated on the table. Our algorithm gives a stable estimate of camera pose for all images in that sequence, achieving comparable performance with the traditional approach, and slightly outperforming it for the case of the widest baseline. The second sequence is of a moderately difficult scene where our algorithm outperforms the traditional approach by remaining stable over the entire sequence. This enhanced stability is the result of our algorithm being able to utilize the edge features provided by the door frame, while the traditional approach is limited to the tightly clustered features on the posters, giving it a small effective field of view. Finally, the last sequence shows a very difficult scene. There are very few feature matches (the point matches from the second image pair are shown in figure 3.1) and by the third image of the sequence, there are insufficient matches for a traditional SfM algorithm to make a pose estimate. Furthermore, the baseline is slightly larger than that shown in the previous two scenes, with a maximum camera rotation of 35.9 degrees about the object of interest. Although the performance of our algorithm at larger baselines degrades, an estimate of the camera pose and the depth can still be recovered at very large baselines.



	$\widetilde{R}$	$\widetilde{T}$	$\widetilde{R}$	$\widetilde{T}$	$\widetilde{R}$	$\widetilde{T}$	$\widetilde{R}$	$\widetilde{T}$
SIFT flow	1.4937	5.3991	2.5160	8.632	22.9022	80.4177	28.5813	68.9093
Ours	1.0469	1.6552	0.8479	0.5030	1.7260	2.0957	2.9341	1.8525
Traditional	1.7581	3.5866	2.6261	2.0945	1.0945	1.4169	3.6841	2.5076



SIFT flow	2.0940	2.0046	3.4780	2.9594	6.4810	2.4950	11.3987	5.8146
Ours	1.6858	2.8187	0	3.3449	0.1470	2.6651	0.6746	4.9110
Traditional	7.976	86.9429	0.8721	3.9055	1.7518	3.5200	18.1968	65.5972



SIFT flow	3.3724	9.0112	15.3820	48.9364	22.4894	49.2501	53.7316	57.2713
Ours	2.6484	4.8031	4.0713	2.2964	7.1120	2.6457	12.1119	12.5768
Traditional	8.7717	10.4135	18.2654	83.1424	NA	NA	84.6166	53.1542

FIGURE 3.10: Sequences in increasing order of difficulty. Camera pose is with respect to the base image on the extreme left. Color-coded depth maps computed from our algorithm's camera pose are included in the second row of each scene sequence, with warm colors representing near depths and cold colors far depths.

3.5.3 Unresolved issues and Discussion

Throughout this chapter, we have emphasized our algorithm's ability to utilize more information than traditional SfM algorithms. However, we should caution that unless properly weighted, more information is not necessarily better. This is illustrated in figure 3.11, where an undulating cloth surface means that the edge information is subject to a great deal of "occlusion" noise, caused by the extremal contours varying with viewpoint changes. inconsistent edge detection. Despite the large amount of occlusion, our algorithm could still return a fairly good estimate; however, re-running our algorithm using only corner information improves the results. This indicates that it is the inclusion of "noisy" information without proper weighting that degrades somewhat the performance of our algorithm. We note that unique corner matches can be better incorporated into our algorithm by allowing these point matches to influence the σ_t values in our Gaussian mixture. A principled fusion of these different sources of match information, together with a well thought-out data weighting scheme would be of great practical value and remains to be properly addressed.

While our algorithm cannot attain the global minimum and more research in that direction is necessary, we would like to make some final remarks on the stability of our algorithm against local minima, whether arising from the inherent ambiguity of the SfM problem, or caused by errors in the initialization. Referring to figures 3.7, 3.8, and 3.10, it can be seen that both "SIFT flow" and "Traditional" sometimes returned a translation estimate that was almost 90 degrees off the correct solution. This is caused by the well known bias of the translation estimate towards the

center of the image (the true translation is lateral in these sequences), which becomes more acute when the feature matches are insufficient or of poor quality. Our algorithm suffers less from these well known local minima of SfM because we can use ambiguous edge features in these circumstances. While initialization with SIFT flow helps reduce the local minima problem, it can be seen from our results that we can converge to a correct solution even when the original SIFT flow initialization is fairly erroneous. This is especially obvious in the sequences with varying baseline in figure 3.10, where our algorithm degrades gracefully with increasing displacement induced noise and worsening SIFT flow initialization.

3.6 Concluding remarks

In this chapter we have extended the point registration framework to handle the two-frame structure from motion problem. Integrating the motion coherence constraint into the joint camera pose and matching algorithm provides a principled means of incorporating feature points with non-unique descriptors. This in turn allows us to recover camera pose from previously difficult SfM scenes where edges are the dominant cues and point features are unreliable.

While the results obtained so far are promising, there is also much scope for further improvements in terms of improving the initialization, incorporation of multiple views, proper weighting of cues, as well as basic improvement to the point registration mechanism.

	$\widetilde{R}$	$\widetilde{T}$
SIFT flow	5.8690	6.1868
Ours	1.4134	3.7060
Ours*	1.3083	2.5192
Traditional	1.3211	1.1139

FIGURE 3.11: Computed point sets on two images of a textured cloth. This image is easy for traditional SfM. However, our point set recovery faces large amount of "self-occlusion" caused by the extremal contours on the blanket varying under viewpoint changes. Under Ours*, we applied our algorithm using only traditional SIFT corner features. The results improve significantly, showing that when there is abundant high quality corner information present, including more noisy edge information can have a negative impact on performance. This scene also illustrates our algorithm ability to give a reasonable estimate despite large amount of noise and occlusion.

Chapter 4

Mosaicing

In this chapter, we design a mosaicing algorithm which can accommodate image parallax. As there is no single set of global parameters which can be used to remove outlying matches, we leverage on the concept that a mosaic provides cues for image correspondence, while correspondence provides cues for a mosaic. Exploiting the relationship between the application and the correspondence, coupled with a smoothly varying affine field allows us to achieve a parallax handling mosaicing algorithm which avoids the outlier correspondence problem.

4.1 Motivation

Image stitching has long been of interest in graphics and vision. Its primary goal is the integration of multiple images into a single seamless mosaic. This serves

many purposes, such as increasing the effective field of view, motion summarization and clean plate photography. Typically, image stitching relies on an underlying transform which warps pixels from one coordinate frame to another. As the transformation must ensure the perceptually accurate alignment of large (often quarter image width or greater) non-overlapping image regions, it needs to be robust to large view point changes and be able to interpolate and extrapolate the motion over significant occlusion. To handle uncontrolled outdoor environments, the transform must also accommodate illumination changes and independent motion. To ensure such a robust warping, mosaicing algorithms have traditionally sought to parameterize the warping field using a sparse set of global transformation parameters, such as the 3×3 affine or Homographic matrix [36]. This sparse parametrization ensures robustness at the expense of flexibility and as a result, is only accurate for a limited set of scenes and motions. For example, the commonly used Homographic transforms are only accurate for planar scenes or parallax free camera motion between source frames i.e. the photographers physical location must be fixed and only rotational motion is permitted.

Dornaika et al. [25] highlighted that ideally, an image stitching algorithm should allow for both general motion and scene structure. As affine and Homographic stitching can be considered a special case of a 3-D world's re-projection, and there is work on [25, 86, 60] combining image stitching with 3-D reconstruction to enable the handling of parallax in source images with general motion. However, using pre-computed 3-D points has a number of drawbacks. Firstly, 3-D reconstruction is only defined on the overlapping sections of the source image, making it difficult

FIGURE 4.1: A girl passing the time by playing chess with herself, an example of our image stitching algorithm.

to integrate the non-overlapping regions, which is the primary objective of image stitching. Secondly, as noted by Liu et al. [60], the 3-D reconstruction pipeline is brittle, with its main components, accurate camera pose recovery and outlier-free matching, still being active research issues. Thirdly, camera pose computation deteriorates if the motion contains too strong a rotational element or if the overlapping image regions are of inadequate size, both of which occur frequently in image stitching.

To achieve flexibility, we turn to the 2D non-rigid warping approaches such as thin plate spline [10], as-rigid-as-possible warping [46] and motion coherence [78]. They eschew the sparsity of parametric warping in favor of considering warping as a general matching problem with a smoothness constraint. This provides the flexibility needed to handle most motion types but at the expense of the motion generalizing ability possible with a sparse parameterizations. Hence, while warping algorithms may be used as a form of interpolation, they are seldom directly

employed to solve traditional two view stitching problems. In this chapter, we seek to adapt the warping framework to take advantage of the fact that many scenes can be modeled as having continuous piecewise smooth depth. To do this, we utilize a formulation based upon the relaxation of the global affine transform. An affine stitching field is defined over the entire coordinate frame. Every point is given an associated affine parameter that is biased towards a pre-computed global affine transform (which plays the role of a regularizer) and smoothness is enforced on the deviation of each affine parameters from the global affine parameter. This permits a region of rather un-smooth 2D motion flow (such as a strong shear, or forward translation) to become smooth as the affine stitching field can assign all pixels in that region can to a single, constant affine parameter. As neighboring regions with significant overlap will share similar affine parameters, we can fit a very smooth affine stitching field over the image. This smoothness allows for easily extrapolation over the non-overlapping regions and is an implicit product of our piecewise smooth depth assumption which provides a "sparsity" (a strong smoothing is sparse in the sense that it limits the possible solution space by severely penalizing non-smoothness) and extrapolation ability similar to that achieved by parameterizing the warping as a single global affine transform. This permits a general stitching algorithm which can extrapolate across occlusion and non-overlapping regions, yet does not require an explicit 3-D reconstruction.

the warping as a single global affine transform.

To robustly compute the desired affine stitching field over large displacements,

SIFT flow [59] Large displacement flow [14] Our Sparse SIFT warping

FIGURE 4.2: Computing the warping using dense SIFT features in the SIFT flow algorithm of [59], large displacement optical flow [14] and our algorithm. Our results are perceptually more pleasing and easier to mosaic. Our warping also extrapolates the motion for occluded regions. These results are for the image pair shown in figure 4.1

illumination change and occlusion noise is a non-trivial problem. If we directly focused on the brightness constancy constraint, even with modern large displacement optical flow algorithms [14], the degree of motion and amount of scene variation we can accommodate would be significantly limited. If however, we utilize view-invariant feature descriptors like SIFT [64] densely over the entire image, a lot of localization error will be introduced as neighboring pixels will likely share similar feature descriptors, thus making accurate and dense, descriptor-based matching difficult, a point illustrated in figure 4.2. Instead, we rely on a sparse set of corner features to compute the stitching field, which has an additional advantage in terms of computation time. While one can extrapolate a stitching field from precomputed point matches, this is extremely vulnerable to outlier matches and a varying stitching field does not permit RANSAC based outlier rejection. Instead, we observe that a good stitching field can help validate existing correspondence and

determine additional ones. These correspondences can in turn refine the stitching field. We exploit the inter-connectedness of these problems by jointly estimating both the matching and the stitching field. This prevents outlier matches, provides significantly more matches and yields a better stitching field.

To summarize our contribution:

1) We introduce a flexible image stitching algorithm that retains much of the motion generalization properties associated with global parametric transforms like affine/ homography. This permits the handling of general scenes and motions provided there are no abrupt protrusions. While our results do not always conform to the ground truth, it provides a good approximate which enables the creation of a perceptually correct composite.

2) We explore a range of applications made possible by this flexibility. These include novel scene generation illustrated in fig 4.1, computation of point correspondence and mosaicing of panoramas from translational motion.

4.1.1 Related Work

There is also a large amount of work which seek to refine conventional parametric image stitching. Interested readers can refer to the comprehensive tutorial by Szeliski [95] for an overview of such image stitching and blending techniques. In contrast to the conventional image stitching techniques which rely on a fixed global parameter for warping, we utilize a flexible stitching to warp the images together.

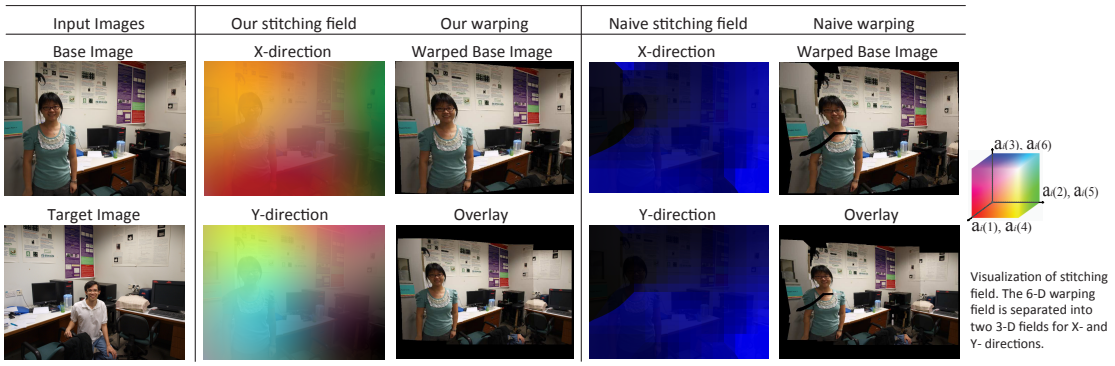


FIGURE 4.3: The affine stitching field transfers the base image to the target. We color code the deviation of each points affine parameters from the global affine parameter and overlay it on the base image. Affine parameters are divided into 2 groups according to the axis they operate on. Parameters in each group are assigned to one RGB color channels. The greater the deviation from the global affine parameter, the brighter the color. We present both our method and a naive method where the stitching field is computed by averaging the affine parameters computed form correspondences within a window. Observe that the naive method's stitching field is strongly biased towards regions where there are many correspondences. This makes it difficult to extrapolate the field to occluded regions such as the girl. Our algorithm can create a smooth field (seen in the color transitions) over the right angled corner, and has better extrapolation ability.

In terms of using affine parameters, our stitching field is related to the affinely over-parameterized optical flow algorithm of Tal et al. [80]. However, it is unclear how the framework of [80] can be adapted to utilize the sparse high dimensional features and a bias towards a pre-defined affine parameters, needed to handle large displacements. For this purpose, we utilize the motion coherence framework of Yuille et al. [110] and Myronenko et al. [78] to fit the a dense affine stitching field over the entire image and anchored at a set of sparse corner matches which we simultaneously estimate.

Our work is also related to the 3-D reconstruction based image stitching methods mentioned in the introduction. These techniques have difficulty integrating the non-overlapping image regions. While this is not important for applications like

Liu et al.s [60] work on 3-D video stabilization, it is the central issue in forming large panoramas. Another solution is to perform 3-D reconstruction using many images. Having a large model naturally suggest one could re-project to recover a large field-of-view image and the 3-view case was discussed in [25]. However, as discussed earlier, 3-D reconstruction can be brittle and may not be well suited to many of the motion types common in mosaicing. Further, this does not answer the central mosaicing issue of how to relate the non-overlapping image regions, thus making it impossible to form mosaics using only two views. An alternative is offered by Qi et.al. [86], where the 3-D reconstruction is used to generate virtual camera images from whom strips are cut to ensure a smooth transition between the non-overlapping regions. This averages the error over the mosaic rather than attempting to align the images and is unsatisfactory because the error is incurred in the constrained overlapping region, rather than the unconstrained non-overlapping region. The lack of an underlying warping field also makes it difficult to handle occlusion and perform blending operations and limits the algorithms applicability to other image editing task such as the image integration example given in figure 4.1.

There is also work seeking to attain a perceptually accurate large field of view image through inputs other than conventional image stills. An interesting work is that of Kopf et al.[52] who generated virtual cameras from a series of "bubbles" or 360^o panoramas, thus creating a long street view. Carrol et al.'s [15] introduced a warping which enables the un-distortion of a very large field-of-view if the user defines a number of straight lines. For video sequences, Rav-Acha et al. [88]

showed it is possible to leverage on the trackability and redundancy present in closely spaced video frames to incrementally stitch a large mosaic from general motion. However, the formulation does not extend to the large displacement discrete-view stitching considered in this chapter.

4.2 Our Approach

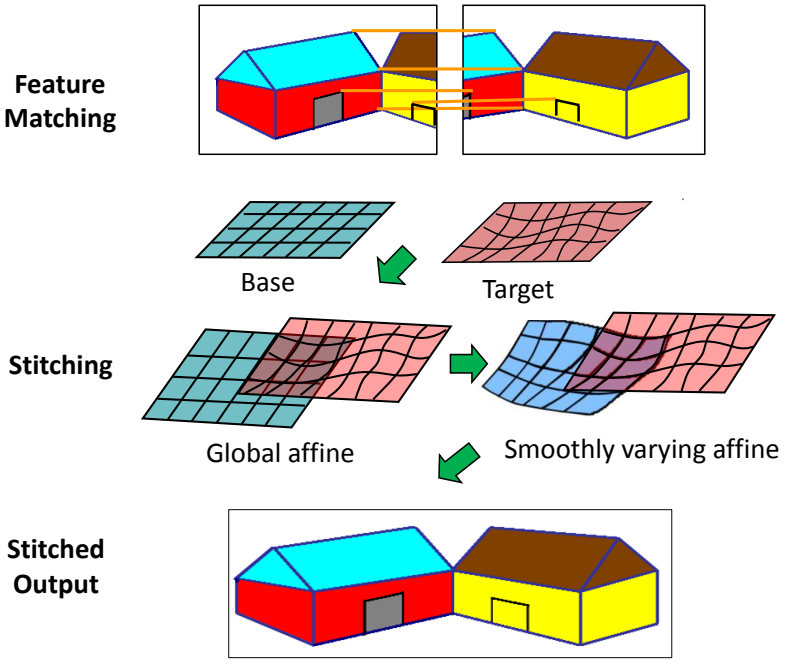


FIGURE 4.4: System overview. Point correspondence is used to obtain a global affine parameter which we relax to form a smooth affine stitching field. The images are warped together and their overlapping regions blended to form a composite.

A naive method of computing an affine stitching field would be to compute local affine parameters from SIFT correspondences within a sliding window. These affine parameters could then be averaged together to give a smooth, dense affine stitching field with the parameters for non-overlapping regions being obtained by extrapolation. This method produces a good, smooth stitching field in regions

where the point correspondence is fairly plentiful. However, the performance declines significantly for regions where there are few/ no point correspondences and the extrapolation is generally poor. The reason is as follows. For any point (or small patch), there are many possible affine parameters that can approximate its motion. Pre-computing an affine parameter from correspondence forces us to choose one of the possibilities. While this choice may be locally optimal, it may not extrapolate well over the rest of the scene. The result is an affine stitching field that fits the regions of dense correspondence very well but does not give due weight to the sparser correspondences from the outlying regions. This problem is illustrated in fig 4.3, where even though we use a fairly large window (which helps avoid local over-fitting) with a length of a quarter image width, the affine stitching field computed still has difficulty extrapolating over regions with few correspondence.

In contrast, our problem is formulated as finding the smoothest stitching field which can align the feature points of both images. This avoids a hard pre-assignment of local affine parameters, while the choice of stitching field carries within it an implicit extrapolation because the stitching field is computed over the whole image. An overview of our system is given in figure 4.4.

Our formulation's primary constraint consists of two sets of unmatched SIFT feature points. We denote the M features from the first, base image as $\mathbf{b}_{0i}$, while the N features from the second, target image, are denoted as $\mathbf{t}_{0j}$, with i and j running from 1 to M, N respectively. The first two entries of the $\mathbf{b}_{0i}, \mathbf{t}_{0j}$ vectors represent image coordinates, with the remaining entries containing SIFT feature

descriptors (This concatenation is done for notational simplicity and the necessary associated normalization is discussed in section 4.3). To obtain the stitching of the $\mathbf{b}_{0i}$'s to $\mathbf{t}_{0j}$'s, we define using a continuous affine stitching field $v(\mathbf{z}_{2\times1}) : \mathbb{R}^2 \rightarrow \mathbb{R}^6$, whose output represents the deviation from a global affine parameters $\mathbf{a}_{global}$. Using $\mathbf{b}_{0i(1)}, \mathbf{b}_{0i(2)}$ to represent the first two entries of vector b_0, this can be expressed as

$$(\Delta\mathbf{a}_i)_{6\times1} = v\left(\left[\mathbf{b}_{0i(1)}; \mathbf{b}_{0i(2)}\right]\right), \tag{4.1}$$

where $\Delta\mathbf{a}_i$ is the deviation of feature i's affine term, $\mathbf{a}_i$, from the global affine parameters, i.e., $\mathbf{a}_i = \mathbf{a}_{global} + \Delta\mathbf{a}_i$.

We use $\mathbf{b}_i$ to represent the stitched feature points. $\mathbf{b}_i$ value depends only on the affine $\mathbf{a}_i$ term associated with the stitching field $v(.)$ and their original position $\mathbf{b}_{0i}$. This relationship can be expressed by the affine transform

$$\mathbf{b}_i = \left[\begin{array}{cc|c} \mathbf{a}_{i(1)} & \mathbf{a}_{i(2)} & \\ & & \mathbf{0}_{2\times S} \\ \mathbf{a}_{i(4)} & \mathbf{a}_{i(5)} & \\ \hline \mathbf{0}_{S\times2} & & \mathbf{I}_{S\times S} \end{array}\right] \mathbf{b}_{0i} + \left[\begin{array}{c} \mathbf{a}_{i(3)} \\ \\ \mathbf{a}_{i(6)} \\ \\ \mathbf{0}_{S\times1} \end{array}\right]. \tag{4.2}$$

To facilitate easy reference to these affine parameters, we also define matrices

$$\mathbf{A}_{M\times6} = [\mathbf{a}_1, ..., \mathbf{a}_M]^T, \Delta\mathbf{A}_{M\times6} = [\Delta\mathbf{a}_1, ..., \Delta\mathbf{a}_M]^T.$$

We relate the base point set's alignment to the target point set, using the conditional probability based on a robust Gaussian mixture

$$P(\mathbf{t}_{01:N}|\mathbf{b}_{1:M}) = \prod_{j=1}^{N} \left(\left(\sum_{i=1}^{M} g(\mathbf{t}_{0j} - \mathbf{b}_i, \sigma_t) \right) + 2\kappa\pi\sigma_t^2 \right), \tag{4.3}$$

where $g(\mathbf{z}, \sigma) = e^{-\frac{\|\mathbf{z}\|^2}{2\sigma^2}}$ is a Gaussian function and κ controls the strength of the uniform pdf which provides a thickening of the Gaussian tails. κ is usually set to 0.5.

Apart form SIFT features, we also desire to incorporate a number of soft constraints. As mentioned earlier, we assume that the stitching field is a relaxation of a single global affine parameter. Hence, we impose a smoothness constraint on the deviation of each points affine parameters from the global affine parameters. We incorporate these soft constraints into a smoothing regularization term. As mentioned in previous chapters, it is difficult to define smoothness over a discrete set of points, as such, we turn to the motion coherence framework where the smoothness of the affine stitching field is defined in the Fourier domain. This gives

$$\int_{\mathbb{R}^2} \frac{|v'(\omega)|^2}{g'(\omega)} d\omega, \tag{4.4}$$

where $v'(\omega)$ denotes the Fourier transform of the continuous stitching field $v(.)$ and $g'(\omega)$ represents the Fourier transform of a Gaussian with spatial distribution γ. The regularization term biases the affine stitching field towards the global affine parameters and ensures smooth transition between the constrained stitching field in the overlapping regions and the extrapolated stitching field in the occluded

regions. While $\mathbf{A}$ is a discrete quantity and the velocity field $v(.)$ is continuous, the regularization term can be re-expressed in terms of $\mathbf{A}$ by choosing the smoothest velocity field which satisfies eqn (4.1). This yields

$$\Psi(\mathbf{A}) = \min_{v'(\omega)} \left(\int_{\mathbb{R}^2} \frac{|v'(\omega)|^2}{g'(\omega)} d\omega \right), \tag{4.5}$$

We combine the negative log of eqn (4.3) with the regularization term in eqn (4.5) and a λ weighting term, to form a single cost function,

$$E(\mathbf{A}) = -\sum_{j=1}^{N} log \left(\left(\sum_{i=1}^{M} g(\mathbf{t}_{0j} - \mathbf{b}_i, \sigma_t) \right) + 2\kappa\pi\sigma_t^2 \right) + \lambda\Psi(\mathbf{A}) \tag{4.6}$$

which can then be minimized with respect to the variables in $\mathbf{A}$. The minimization employs the EM formulation, successfully used in [78].

4.2.1 Minimization

Our minimization procedure is similar to that in the previous chapter and most of the details are placed in the appendix.

We define

$$\begin{aligned} \phi_{ij}(\mathbf{b}_i, \mathbf{t}_{0j}) &= g(\mathbf{t}_{0j} - \mathbf{b}_i, \sigma_t), \\ \overline{\phi_{ij}}(\mathbf{A}, \mathbf{t}_{0j}) &= \frac{\phi_{ij}(\mathbf{b}_i, \mathbf{t}_{0j})}{\sum_l \phi_{lj}(\mathbf{b}_l, \mathbf{t}_{0j}) + 2\kappa\pi\sigma_t^2}. \end{aligned} \tag{4.7}$$

We then follow a minimization procedure which computes an $\mathbf{A}^{k+1}$, using the $M \times 6$ linear equations (4.8) defined by $\mathbf{A}^k$. Compared to $\mathbf{A}^k$, $\mathbf{A}^{k+1}$ lowers the overall cost cost defined in equation (4.6), with the process being iterated until convergence. As mentioned earlier, the derivation of equation 4.8 is placed in the appendix.

$$\begin{aligned}\frac{\delta Q}{\delta \mathbf{A}^{k+1}} &= \begin{bmatrix} \mathbf{c}_1 & \mathbf{c}_2 & \dots & \mathbf{c}_M \end{bmatrix} + 2\lambda \left(\Delta \mathbf{A}^{k+1}\right)^T \mathbf{G}^{-1} \\ &= \mathbf{C} + 2\lambda \left(\Delta \mathbf{A}^{k+1}\right)^T \mathbf{G}^{-1} = \mathbf{0}_{6\times M} \\ \Rightarrow &\mathbf{CG} + 2\lambda \left(\Delta \mathbf{A}^{k+1}\right)^T = \mathbf{0}_{6\times M},\end{aligned} \tag{4.8}$$

$$\mathbf{c}_i = \sum_{j=1}^{N} \frac{\overline{\phi_{ij}}(\mathbf{A}^k, \mathbf{t}_{0j})}{\sigma_t^2} \mathbb{D}(\mathbf{b}_i^{k+1} - \mathbf{t}_{0j}) \mathbb{V}(\mathbf{b}_{0i}).$$

$\mathbb{D}(.), \mathbb{V}(.)$ are simultaneous truncation and tiling operators. They re-arrange **only the first two entries** of an input vector $\mathbf{z}$ (where $\mathbf{z}$ must have a length greater or equal to 2) to form the respective output matrices

$$\mathbb{D}(\mathbf{z})_{6\times 6} = \left[\begin{array}{c|c} \mathbf{z}_{(1)}\mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 3} \\ \hline \mathbf{0}_{3\times 3} & \mathbf{z}_{(2)}\mathbf{I}_{3\times 3} \end{array}\right]$$

$$\mathbb{V}(\mathbf{z})_{6\times 1} = \begin{bmatrix} \mathbf{z}_{(1)} & \mathbf{z}_{(2)} & 1 & \mathbf{z}_{(1)} & \mathbf{z}_{(2)} & 1 \end{bmatrix}^T$$

From the definition of $\mathbf{A}$ in (4.2), we know that $\mathbf{b}_i^{k+1}$ can be expressed as a linear combination of the entries of $\mathbf{A}^{k+1}$. Hence, equation (4.8) produces $M \times 6$ linear equations which can be used to estimate $\mathbf{A}^{k+1}$. $\mathbf{A}^{k+1}$ is used to estimate $\mathbf{A}^{k+2}$ and the process is repeated until convergence.

After convergence, the continuous stitching field $v(.)$ at any point $\mathbf{z}_{2\times1}$ can be obtained from $\mathbf{A}$ using a weighted sum of Gaussian given by

$$\begin{aligned} \mathbf{W}_{M\times6} &= [\mathbf{w}_1, ...\mathbf{w}_M]^T = \mathbf{G}^+\Delta\mathbf{A}, \\ v(\mathbf{z}_{2\times1}) &= \sum_{i=1}^{M} \mathbf{w}_i g(\mathbf{z} - \begin{bmatrix} \mathbf{b}_{0i(1)} & \mathbf{b}_{0i(2)} \end{bmatrix}^T, \gamma), \end{aligned} \tag{4.9}$$

where $\mathbf{G}^+$ is the pseudo-inverse of $\mathbf{G}$ and the $6 \times 1, \mathbf{w}_i$ vectors can be considered weights for the Gaussians. The detailed proof is given in the appendix.

4.3 Implementation

Input: Base image features $\mathbf{b}_i$, target image features $\mathbf{t}_j$, global affine matrix a_{global}
while σ_t *above threshold* **do**
  while *No convergence* **do**
    Use eqn (4.7) to evaluate $\phi_{ij}(\mathbf{b}_i^k, \mathbf{t}_{0j})$ from $\mathbf{A}^k$;
    Use eqn (4.8) to determine $\mathbf{A}^{k+1}$ from $\phi_{ij}(\mathbf{b}_i^{old}, \mathbf{t}_{0j})$
  end
  Anneal $\sigma_t = \alpha\sigma_t$, where $\alpha < 1$.
end
Output: $\mathbf{A}^{converged}$

FIGURE 4.5: Algorithm to compute stitching field.

We now discuss system implementation. A process overview is given in fig 4.4, with stitching field computation algorithm in fig 4.5. In the formulation section, we have a global affine regularization term, $\mathbf{a}_{global}$. $\mathbf{a}_{global}$ is computed from sift correspondences using a RANSAC [31] for outliers removal. As $\mathbf{a}_{global}$'s regularization role lies in ensuring a smoother stitching field, its precise value is not important. All the initial $\mathbf{a}_i$ vectors in $\mathbf{A}$ are originally set to $\mathbf{a}_{global}$. The affine

stitching field then computed by repeatedly minimizing the cost in eqn (4.6) with increasingly smaller values of σ_t. Each step in this annealing process uses the previously calculated stitching field is as an initialization. We begin with $\sigma_t = 1$ and decrement it by a factor of $\alpha = 0.97$, until it reaches the value of 0.1. The progressively smaller σ_t values increase the penalty for deviation between the target and base point sets, forcing the affine stitching field to evolve such that the base point coordinates register onto the target points.

For notational simplicity, SIFT descriptors and point coordinates are condensed into a single. This implies a need for normalization. The point coordinates for the target and base points are normalized to have zero mean, unit variance, thus making the remaining parameter settings invariant to image size. We normalize the SIFT descriptors to have magnitudes of $10\sigma_t$, which gives good empirical results. The smoothing weight λ and outlier handling term κ are assigned values of $10, 0.5$ respectively. The γ term which penalizes un-smooth flow, is set to 1. The stitching field is used to align the images, which we then blend into a single mosaic using the Poisson blending with optimal stitch finding algorithm of [16].

4.4 Analysis

The computed smooth affine stitching field is a "sparse" representation of the true warping function and errors will be incurred by smoothing over depth boundaries and extrapolating from a small set of feature points. In figure 4.6, we use two simulated scenes. The first scene contains no major depth discontinuities and the

average warping error was 1.92 pixels, while the maximum error was 12.573 pixels. This is quite low, considering we did not perform any of plane segmentation. The second scene is more difficult, with a depth discontinuity of approximately half of the average depth. Smoothing over the depth discontinuity greatly increases the maximum error to 42.87 pixels, however the overall warping remains stable and the average error is still low at 4.57 pixels. Generally, our algorithm shows good extrapolation abilities as it can generalized the motion of a 0.25 megapixels using 625 feature points, a ratio of $1 : 400$.

In fig 4.7, we provide a qualitative errors analysis of an image pair taken as the camera fixates on an object. The stitched images are overlayed. In the overlapping region, the green color channel is from the base image while the red and blue channels come from the target image. This allows a visualization of alignment errors which appear in the form of ghosting. While our algorithm incurs some errors along depth boundaries, they can be removed by blending.

4.5 Applications

Our algorithms flexibility means that it can stitch images even when the photographer does not maintain a fixed position. This opens up a range of different possibilities.

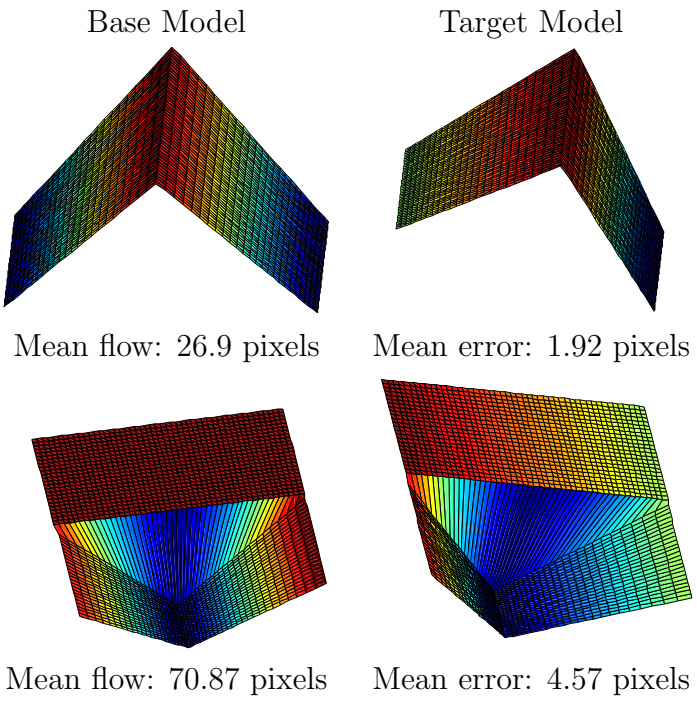


FIGURE 4.6: Quantitative analysis of our algorithms motion generalization ability. The camera rotates 0.3 radians about the object. Using 625 uniformly distributed, unique features, we generalize the motion of a 500×500 image (0.25 megapixels), a 1:400 ratio.

4.5.1 Re-shoot

Bae et al.[2], noted that if the photographer has moved away from the original location, it is difficult to recover the exact view point. Our algorithms good motion generalization and flexible stitching capability mean we can "re-shoot" a scene to incorporate information from different time instances, without having to ensure the photographer's position is exactly the same.

Observe that image editing using "re-shoot" differs from a "cut and paste" method of overlaying an object onto a background image. In "cut and paste", the overlay must be a discrete object such as a man or a car, with no attached background. As our stitching algorithm automatically warps the appended region to fit smoothly

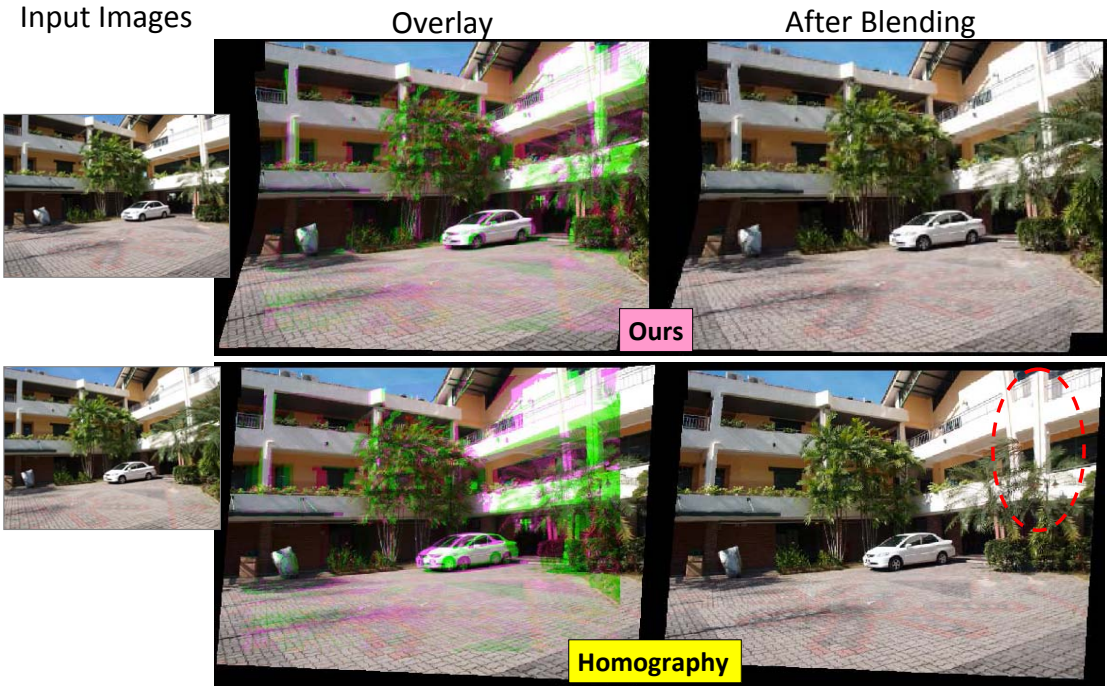


FIGURE 4.7: Results before and after Poisson blending. For the pre-blended images, the overlapping regions take the green color channel from the base image and the red, blue channels from the target image. This enables visualization of alignment errors which appears in the form of ghosting. Our algorithm incurs some errors along the depth boundaries. However, after blending, the errors are not noticeable. The homographic mosaicing, incurs much larger errors and even after applying the same blending, clear artifacts remain.

with the target image, "re-shooting" allows the overlay of an entire region, including the complex background and the subjects interactions with it.

In the first two scenes of fig 4.8, we insert a person into an image where he/she was not originally present and conversely, remove a person from the image. This allows interesting compositions such as a girl playing chess with herself in a cafe and allows two people to alternate as photographers to obtain a group photo. Note that the images are not taken from the same view point. The following two scenes of fig 4.8 test our algorithm's limit by using internet images. These are much more challenging because photometric changes affect the SIFT feature invariance our algorithm depends on. However, it permits more dramatic effects such as

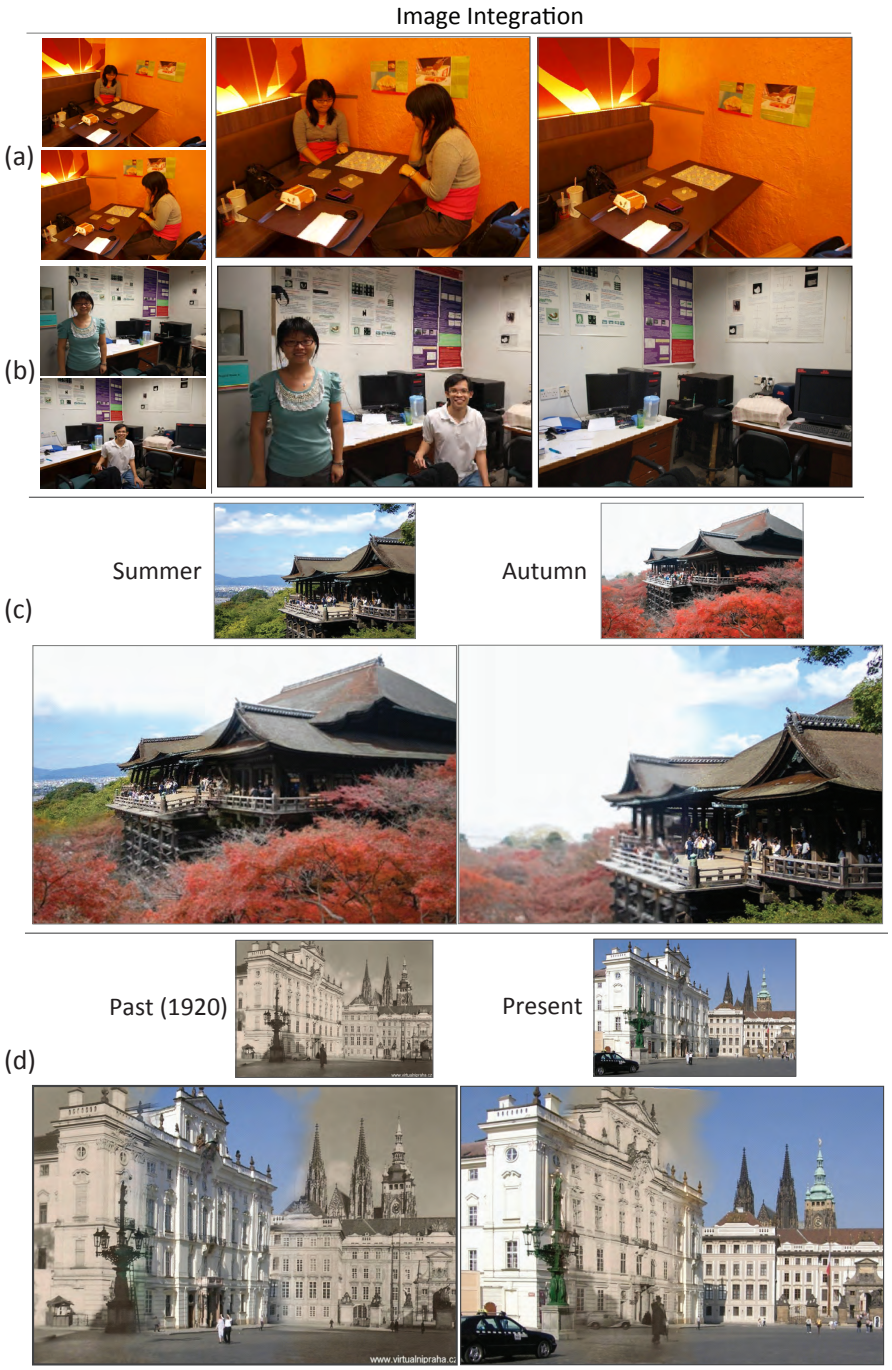


FIGURE 4.8: "Re-shoot" permits the integration of image pairs to create novel composites formed by fusing different image portions. Our composites permit the subject to interact with the environment, something which is not possible using conventional "cut and paste" image fusing methodology. In sub-images **(a,b)** we fuse the images in the left column to form composites on the right. **(a)** shows a girl passing the time by playing chess with herself, while **(b)** shows two people alternating as photographers to obtain a group photo. In **(c,d)**, we fuse images from different time periods. In **c** we show the changing seasons at the famous Kiyomizu temple in Japan while **(d)** marks the passage of time at the Archbishop's palace in Prague.

integration of summer time vista with the spectacular autumn foliage at Kiyomizu temple in Japan, as well as an image of a young couple walking from Prague's present into its past. We believe that our algorithm can be adapted to permit changes in the SIFT feature which would significantly improve its performance on internet images.

Technical discussion: "Re-shoot" is more challenging than panorama formation as the available blending region is narrow and the amount of occlusion typically very large. To ensure image consistency, we normalize the image colors. Blending is based on the Poisson blending with optimal cut implementation in [16] and is followed by a additional alpha blending to merge the colors. This is carried out on a 25 pixel wide boundary along a user defined transfer region. For the shots using internet images, the blending boundary is set to be 50 pixels wide to accommodate the photometric variations and color normalization is discarded. In the Prague scene, the global affine was not pre-computed (due to a shortage of reliable matches) but set to an identity matrix. A more sophisticated blending for "re-shoot", can by obtained from [1].

4.5.2 Panoramic stitching

Our algorithm can be used for panorama creation. Its ability to handle general motion allows image stitching from un-conventional sequences, such as a series of images taken from different windows of a high-rise flat. As most windows are set back from the facade, this is not possible with homographic mosaicing [12] which requires a large un-occluded rotational field of view from a single window. Results

are shown in figs 4.9,4.10. Observe that many of the views have only limited overlap, making camera pose recovery and hence mosaicing via 3-D reconstruction difficult.

4.5.3 Matching

Our algorithm can serve as a matcher across two views that can be related by a smoothly varying affine field (it will not match independent motion). As it matches features as a set, rather than individually, there is reduced dependency on the feature descriptor uniqueness. In fig 4.11 we show that applying our algorithm with traditional SIFT descriptors [64], we can obtain 40% more matches. This is more than using a nearest neighbor matcher with more sophisticated A-SIFT [76] descriptors.

4.6 Concluding remarks

We present an image stitching algorithm based on a smoothly varying affine stitching field. It is significantly more tolerant to parallax than traditional homographic stitching but retains much of homographie's ability to extrapolate motion over occlusion. Its flexibility enables integration of views taken from different physical locations, permitting a number of interesting applications like panorama creation from a translating camera or integration of images taken at different times. Our algorithm's primary limitation is the violation of affine coherence at depth boundaries. While our results show these errors are often small enough to be blended

FIGURE 4.9: Input images in **(a)** are taken from a series of windows. Our mosaic in **(b)** is perceptually accurate while homographic mosaicing using AutoStitch [12] in **(c)** has difficulty merging the fore-ground buildings, a close up view being show in **(d)**.

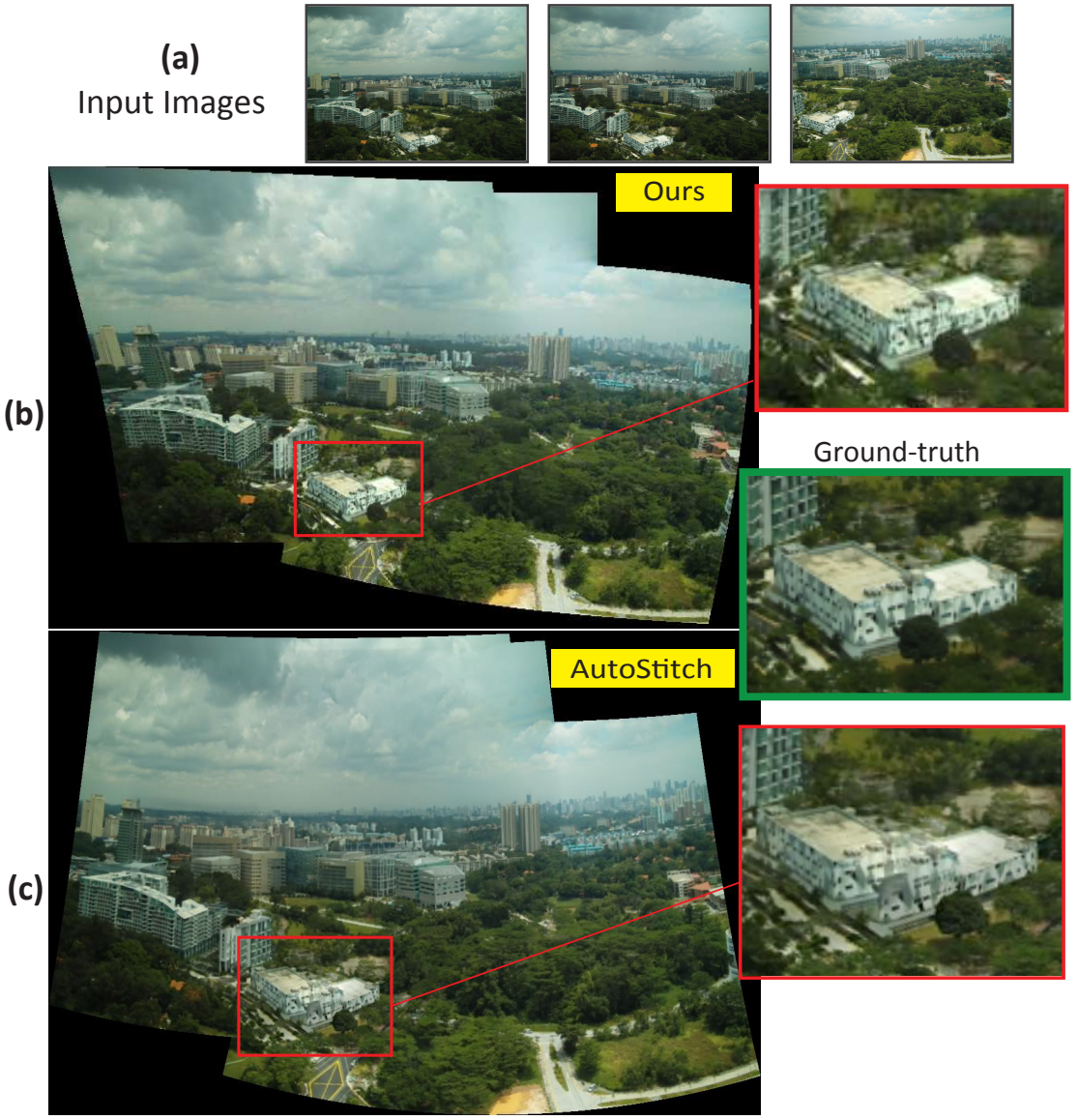


FIGURE 4.10: Input images in **(a)** are taken from a series of windows. Our mosaic in **(b)** is perceptually accurate. **(c)** shows homographic mosaicing using AutoStitch [12], which has difficulty merging the fore-ground buildings.

over, explicit detection and handling would be better. In this regard, our results provide an excellent starting point for further refinement.

FIGURE 4.11: We show the results obtained by using our algorithm as a matcher and compare against conventional nearest neighbor SIFT [64] and A-SIFT [76] feature matching. Although we use traditional SIFT [64] descriptors, we can obtain more matches than applying nearest neighbor matching to the more sophisticated A-SIFT descriptor. The above figures shows that the additional matches do not come at the expense of accuracy and the matching is stable to significant occlusion.

Chapter 5

Conclusions and Future Work

In this thesis we explore the interlocking relationship between the underlying 3-D structure, applications seeking to exploit it and correspondence noise. As such, our work has resulted in 3 primary contributions.

Firstly, we show that focusing on the relationship between correspondence and noise can provide interesting insights which enable the bridging of the differential and discrete Structure from Motion problems, something we illustrated in chapter 2, where we studied this concept using perturbation theory analysis.

Secondly we exploit the interlocking relationship between camera pose and correspondence noise in a practical applications such as incorporating edge information into the Structure from Motion problem, which we achieved in chapter 3 by fusing camera pose estimation into a motion coherence matching framework. This enables computation of camera pose using edge information, without explicitly assuming straight lines, something which was mpt achievable before.

Thirdly, in chapter 4, we employed a similar coherent motion technique to create a parallax handling mosaicing algorithms which stably computed the inter-image correspondence using a smoothly varying affine stitching field does not require explicit 3-D reconstruction.

We observe that the full relationship between matching and 3-D structure exploitation remains a relatively unexplored field, with many fundamental issues relating to cue weighting, noise modeling and ideal optimization formulation remaining unexplored. There are also many alternative applications, such as independent motion detection, and constraints like straight line or ground plane preservation, which have not been addressed in this thesis. Many of these issues can be handled by extending the approaches we have discussed in this thesis and we are hopeful that future work will explore these issues more thoroughly.

On a higher level, we observe that computer vision is a very compartmentalized field, with many researchers delving deeply into a wide array of sub-problems, ranging from independent motion detection, interest point tracking, ground plane detection and camera pose recovery. Most of these fields have strongly interrelated components. If a region is a ground plane, it can't be independently moving, if we have tracked a point moving from left to right and find our camera in the opposite direction, either camera motion is incorrect, the point tracks are incorrect or possibly both. We feel that such interrelated concepts should be combined in a fashion which reflects their mutual dependence, with an estimation algorithm which does not hurry to fix either one quantity or another but jointly estimates both. Such work which brings together results from two very well researched fields

may yield surprising results and is something we are very interested to pursue. On that note, we would like to conclude the thesis.

Appendix A

Proofs related to Chapter 2

A.1 Perturbation of Eigenvalues and Eigenvectors

We record some results on perturbation theory from Wilkinson [106]. The first two results are due to Gerschgorin. These Gerschgorin Disc Theorems give us a method of estimating the eigenvalues of a matrix based solely on the entries of the matrix.

Theorem A.1. *([106], Theorem 3, page 71.) Every eigenvalue λ of an $n \times n$ matrix C lies in at least one of the circular discs with centers c_{ii} and radii $\sum_{j \neq i} |c_{ij}|$, where c_{ij} is the entry of the matrix C on its ith row and jth column.*

The above circular disc is called a Gerschgorin disc.

Theorem A.2. *([106], Theorem 4 , page 71.) If k of the Gerschgorin disc form a connected domain which is isolated from the other discs, then there are precisely k eigenvalues of C within this connected domain.*

The next result is a slight modification of the above Gerschgorin's Theorems. It is applied to the matrix $\widetilde{A}^T(\epsilon)\widetilde{A}(\epsilon)$ in Equation (2.28) in Subsection 2.5.1.

Proposition A.3. *Let $\widetilde{C} = C+H$, where $\widetilde{C}, C$ and H are $n\times n$ matrices. Suppose there is an invertible matrix K such that $K^{-1}CK = D$, where $D = Diag(d_i)$ is a diagonal matrix with diagonal entries d_i. Then every eigenvalue $\widetilde{\lambda}$ of $\widetilde{C}$ lies in at least one of the circular discs $\widetilde{G}_i$ with center d_i and radius $\sum_{j=1}^{n} |(K^{-1}HK)_{ij}|$, where $(K^{-1}HK)_{ij}$ is the (ij)-entry of the matrix $K^{-1}HK$.*

Moreover, if k of the above circular discs form a connected domain which is isolated from the other discs, then there are precisely k eigenvalues of $\widetilde{C}$ within this connected domain.

Proof. Firstly, we note that the matrices $K^{-1}\widetilde{C}K$ and $\widetilde{C}$ have the same set of eigenvalues. Now, by the first Gerschgorin's result, namely Theorem A.1, every eigenvalue of $K^{-1}\widetilde{C}K$, and hence of $\widetilde{C}$, lies in one of its Gerschgorin discs. The ith Gerschgorin disk G_i of the matrix $K^{-1}\widetilde{C}K$ is given by

$$G_i = \left\{ \lambda \;\middle|\; |\lambda - \left(d_i + (K^{-1}HK)_{ii}\right)| \leq \sum_{j\neq i} |(K^{-1}HK)_{ij}| \right\}. \tag{A.1}$$

Applying the triangle inequality to the inequality in (A.1) gives

$$|\lambda - d_i| \leq \sum_j |(K^{-1}HK)_{ij}|,$$

which defines a circular disc $\widetilde{G}_i$ centered at d_i and with radius $\sum_j |(K^{-1}HK)_{ij}|$. This circular disc contains the ith Gerschgorin disk G_i. Consequently, every eigenvalue of $\widetilde{C}$ lies in one of such circular discs. The second part of the proposition now follows readily from Theorem A.2. □ □

Before we proceed to obtain the perturbation of eigenvectors, we first include a simple proof of the next lemma which is used to provide us with a unit vector.

Lemma A.4. *Suppose* $\mathbf{q}(\epsilon) = \mathbf{r}(\epsilon) + \mathbf{z}(\epsilon)$ *where* $\mathbf{r} = O(\epsilon^0)$ *and* $\mathbf{z}(\epsilon) = O(\epsilon)$. *Then the unit vector*

$$\breve{\mathbf{q}}(\epsilon) = \frac{\mathbf{q}(\epsilon)}{\|\mathbf{q}(\epsilon)\|}$$

can be expressed as

$$\breve{\mathbf{q}}(\epsilon) = \breve{\mathbf{r}}(\epsilon) + \mathbf{w}(\epsilon)$$

where $\breve{\mathbf{r}}(\epsilon) = \frac{\mathbf{r}(\epsilon)}{\|\mathbf{r}(\epsilon)\|}$ *and* $\mathbf{w}(\epsilon) = O(\epsilon)$.

Proof. Note that $\|\mathbf{q}(\epsilon)\| = \|\mathbf{r}(\epsilon)\| + O(\epsilon)$. Thus,

$$\begin{aligned}
\breve{\mathbf{q}}(\epsilon) &= \tfrac{\mathbf{q}(\epsilon)}{\|\mathbf{q}(\epsilon)\|} \\
&= \tfrac{1}{\|\mathbf{r}(\epsilon)\|+O(\epsilon)} \left(\mathbf{r}(\epsilon) + \mathbf{z}(\epsilon)\right) \\
&= \tfrac{1}{\|\mathbf{r}(\epsilon)\|(1+O(\epsilon))} \left(\mathbf{r}(\epsilon) + \mathbf{z}(\epsilon)\right) \\
&= \breve{\mathbf{r}}(\epsilon) + \mathbf{w}(\epsilon)
\end{aligned}$$

where $\mathbf{w}(\epsilon) = O(\epsilon)$. We have made use of $\frac{1}{1+O(\epsilon)} = 1 + O(\epsilon)$ from Equation (2.3). □ □

For a perturbed symmetric matrix, we first have the following result on its perturbed eigenvalues from [106].

Theorem A.5. *([106], Wielandt-Hoffman Theorem, page 104.) Suppose* $\widetilde{C}(H) = C + H$*, where* $\widetilde{C}(H), C$ *and* H *are* $n \times n$ *real symmetric matrices. If* $\widetilde{C}(H)$ *and* C *have eigenvalues* $\widetilde{\lambda}_i(H)$ *and* λ_i *respectively and they are arranged in non-increasing order, then*

$$\sum_{i=1}^{n} (\widetilde{\lambda}_i(H) - \lambda_i)^2 \leq \|H\|^2.$$

It follows that for each i,

$$|\widetilde{\lambda}_i(H) - \lambda_i| \leq \|H\|. \tag{A.2}$$

In the above statement, we have used the symbol $\widetilde{C}(H)$ instead of $\widetilde{C}$ for $C + H$ to stress the dependence of its eigenvalue $\widetilde{\lambda}_i(H)$ on H.

To obtain the perturbed eigenvectors, we may apply a technique in [106] which we have also used in Section 2.6. The idea is quite simple and we thus state the result without proof.

Lemma A.6. *Let* $\widetilde{C}(H) = C + H$*, where* $\widetilde{C}(H), C$ *and* H *are* $n \times n$ *real symmetric matrices. Suppose* $\{\mathbf{r}_i | 1 \leq i \leq n\}$ *is a basis of eigenvectors of* C*, where each* $\mathbf{r}_i$ *is an eigenvector that corresponds to eigenvalue* λ_i*. For a fixed* k*, let*

$$\mathbf{q}_k(H) = \sum_{i=1}^{n} \alpha_i(H)\mathbf{r}_i$$

be an eigenvector of $\widetilde{C}(H)$ *corresponding to eigenvalue* $\widetilde{\lambda}_k(H)$*. Suppose* j *is such that* $\lambda_j \neq \lambda_k$*. Then for a sufficiently small* $\|H\|$*, the projection* $\alpha_j(H)$ *of* $\mathbf{q}_k(H)$ *on* $\mathbf{r}_j$ *is non-maximal, i.e.,*

$$|\alpha_j(H)| \neq max\{|\alpha_i(H)| \mid 1 \leq i \leq n\}.$$

Theorem A.7. *Suppose* $\widetilde{C}(H) = C + H$*, where* $\widetilde{C}(H), C$ *and* H *are* $n \times n$ *real symmetric matrices. The unit eigenvectors* $\widetilde{\mathbf{q}}_k(H)$ *and* $\mathbf{r}_k$ *of* $\widetilde{C}(H)$ *and* C *corresponding to* $\widetilde{\lambda}_k(H)$ *and* λ_k *respectively are related by*

$$\widetilde{\mathbf{q}}_k(H) = \mathbf{q}'_k(H) + O(\|H\|),$$

where $\mathbf{q}'_k(H)$ *is a unit vector and is a linear combination of all those eigenvectors* $\mathbf{r}_j$ *of* C*, whose associated eigenvalue* λ_j *is identical to* λ_k*.*

A.2 Errors in the Translation Vector and Rotation Matrix

In this section, we show that the decomposition of an essential matrix into its rotational and translational terms is stable. This means that in general, lowering the amount of noise will improve the the rotational as well as the translational estimate, rather than only one or the other.

Let $E(\epsilon)$ be an essential matrix with a finite Frobenius norm. Recall $\widetilde{E}'(\epsilon)$ is the corrupted version of $E(\epsilon)$ but it has been corrected to possess the desired properties of an essential matrix. The rotation and translation estimates are obtainable from the SVD of $\widetilde{E}'(\epsilon)$ using the algorithm in [40]. The SVD process involves the eigenvalues and eigenvectors of the real symmetric matrices $\widetilde{E}'^T(\epsilon)\widetilde{E}'(\epsilon)$ and $\widetilde{E}'(\epsilon)\widetilde{E}'^T(\epsilon)$, of whom we note the following (ϵ is here suppressed temporarily):

$$\begin{aligned} \|\widetilde{E}'^T\widetilde{E}' - E^TE\| &= \|\widetilde{E}'^T(\widetilde{E}' - E) + (\widetilde{E}'^T - E^T)E\| \\ &\leq \|\widetilde{E}'^T\|\ \|\widetilde{E}' - E\| + \|(\widetilde{E}'^T - E^T)\|\ \|E\| \\ &\leq \|\widetilde{E}' - E\| \left(2\|E\| + \|\widetilde{E}' - E\|\right), \end{aligned} \tag{A.3}$$

which has the same order as $\|\widetilde{E}'(\epsilon) - E(\epsilon)\|$, since $\|E(\epsilon)\|$ is finite. The same result can be obtained for $\|\widetilde{E}'(\epsilon)\widetilde{E}'^T(\epsilon) - E(\epsilon)E^T(\epsilon)\|$. Thus, both errors have the same order as $\|\widetilde{E}'(\epsilon) - E(\epsilon)\|$.

Consider the SVD of the matrix $E(\epsilon)$

$$E(\epsilon) = U(\epsilon) \begin{bmatrix} \sqrt{\lambda(\epsilon)} & 0 & 0 \\ 0 & \sqrt{\lambda(\epsilon)} & 0 \\ 0 & 0 & 0 \end{bmatrix} V^T(\epsilon) \tag{A.4}$$

where $\lambda(\epsilon) = \lambda + O(\epsilon)$ is a positive real number, and $U(\epsilon)$ and $V(\epsilon)$ are orthogonal matrices. Each ith column $\mathbf{v}_i(\epsilon)$ of $V(\epsilon)$ is a unit eigenvector of $E^T(\epsilon)E(\epsilon)$ that corresponds to the eigenvalue $\lambda(\epsilon)$ for $i = 1, 2$ and 0 for $i = 3$.

Likewise, we have the corresponding SVD of $\widetilde{E}'(\epsilon)$:

$$\widetilde{E}'(\epsilon) = U'(\epsilon) \begin{bmatrix} \sqrt{\lambda'(\epsilon)} & 0 & 0 \\ 0 & \sqrt{\lambda'(\epsilon)} & 0 \\ 0 & 0 & 0 \end{bmatrix} (V'(\epsilon))^T. \tag{A.5}$$

where the ith column of $V'(\epsilon)$ is the unit eigenvector of $\widetilde{E}'^T(\epsilon)\widetilde{E}'(\epsilon)$.

Using Equations (A.4) and (A.5), for $i = 1, 2$, the ith columns $\mathbf{u}_i(\epsilon)$, $\mathbf{v}_i(\epsilon)$, $\mathbf{u}'_i(\epsilon)$ and $\mathbf{v}'_i(\epsilon)$ of the respective matrices $U(\epsilon)$, $V(\epsilon)$, $U'(\epsilon)$ and $V'(\epsilon)$ are related as follows,

$$\begin{aligned} \sqrt{\lambda(\epsilon)}\mathbf{v}_i(\epsilon) &= E^T(\epsilon)\mathbf{u}_i(\epsilon), \\ \sqrt{\lambda'(\epsilon)}\mathbf{v}'_i(\epsilon) &= \widetilde{E}'^T(\epsilon)\mathbf{u}'_i(\epsilon). \end{aligned} \tag{A.6}$$

From [40], the translation directions associated with $\widetilde{E}'(\epsilon)$ and $E(\epsilon)$ are given by the third columns $\mathbf{u}'_3(\epsilon)$ and $\mathbf{u}_3(\epsilon)$ respectively. The next result gives the error involved in these translation vector estimates.

Proposition A.8. *For the unit translational vectors* $\mathbf{u}_3(\epsilon)$ *and* $\mathbf{v}_3(\epsilon)$*, the errors* $\|\mathbf{u}'_3(\epsilon) - \mathbf{u}_3(\epsilon)\|$ *and* $\|\mathbf{v}'_3(\epsilon) - \mathbf{v}_3(\epsilon)\|$ *have the same order as* $\|\widetilde{E}'(\epsilon) - E(\epsilon)\|$.

Proof. The vectors $\mathbf{u}_3(\epsilon)$, $\mathbf{v}_3(\epsilon)$ $\mathbf{u}'_3(\epsilon)$ and $\mathbf{v}'_3(\epsilon)$ are unit eigenvectors corresponding to the simple eigenvalue 0 of the real symmetric matrices $E^T(\epsilon)E(\epsilon)$, $E(\epsilon)E^T(\epsilon)$, $\widetilde{E}'^T(\epsilon)\widetilde{E}'(\epsilon)$ and $\widetilde{E}'(\epsilon)\widetilde{E}'^T(\epsilon)$ respectively.

The result now follows readily from Theorem A.7 in Appendix A.1 and the error obtained in (A.3). □ □

Next, we relate $U(\epsilon)$ to $U'(\epsilon)$ and $V(\epsilon)$ to $V'(\epsilon)$ when ϵ is sufficiently small. This relationship is then used to determine the error in the estimate of the rotation matrix.

Lemma A.9. *Both* $\|U'(\epsilon) - U(\epsilon)\|$ *and* $\|V'(\epsilon) - V(\epsilon)\|$ *have the same order as* $\|\widetilde{E}'(\epsilon) - E(\epsilon)\|$.

Proof. The non-zero eigenvalue of the real symmetric matrix $E^T(\epsilon)E(\epsilon)$ (and hence also $E(\epsilon)E^T(\epsilon)$) is repeated twice. Hence, the corresponding eigen space has dimension 2. Therefore, we choose $\mathbf{u}_2(\epsilon)$ and $\mathbf{u}_2'(\epsilon)$ such that

$$\mathbf{u}_2'(\epsilon) = \mathbf{u}_2(\epsilon) + O(\|\widetilde{E}'(\epsilon) - E(\epsilon)\|). \tag{A.7}$$

(Here we have used Equation (A.3).)

Now, we view $\widetilde{E}'^T(\epsilon)\widetilde{E}'(\epsilon)$ as a perturbation of $E^T(\epsilon)E(\epsilon)$ with $\|\widetilde{E}'^T(\epsilon)\widetilde{E}'(\epsilon) - E^T(\epsilon)E(\epsilon)\| = O(\|\widetilde{E}'(\epsilon) - E(\epsilon)\|)$. By the Wielandt-Hoffman Theorem (recorded as Theorem A.5 in Appendix A.1), the perturbed eigenvalue $\lambda_i'(\epsilon)$ is

$$\lambda_i'(\epsilon) = \lambda_i(\epsilon) + O(\|\widetilde{E}'(\epsilon) - E(\epsilon)\|), i = 1, 2,$$

so that, using Equations (A.6), (A.7) and (2.3), we obtain

$$\begin{aligned} \mathbf{v}_2'(\epsilon) &= \frac{(\widetilde{E}'(\epsilon))^T\mathbf{u}_2'(\epsilon)}{\sqrt{\lambda(\epsilon)+O(\|\widetilde{E}'(\epsilon)-E(\epsilon)\|)}} \\ &= \frac{1}{\sqrt{\lambda(\epsilon)}}(\widetilde{E}'(\epsilon))^T\mathbf{u}_2(\epsilon) + O(\|\widetilde{E}'(\epsilon) - E(\epsilon)\|). \end{aligned}$$

Therefore, we have

$$\|\mathbf{v}_2'(\epsilon) - \mathbf{v}_2(\epsilon)\| \leq \|\widetilde{E}'(\epsilon) - E(\epsilon)\| \ \frac{\|\mathbf{u}_2(\epsilon)\|}{\sqrt{\lambda(\epsilon)}} + \|O(\|\widetilde{E}'(\epsilon) - E(\epsilon)\|)$$

which is of the same order as $\|\widetilde{E}'(\epsilon) - E(\epsilon)\|$.

Now, the vector $\mathbf{u}_1(\epsilon)$ (respectively $\mathbf{v}_1(\epsilon)$), being orthogonal to both $\mathbf{u}_2(\epsilon)$ and $\mathbf{u}_3(\epsilon)$ (respectively $\mathbf{v}_2(\epsilon)$ and $\mathbf{v}_3(\epsilon)$), can be obtained by taking the unit vector along $\widehat{\mathbf{u}_2(\epsilon)}\mathbf{u}_3(\epsilon)$ (respectively $\widehat{\mathbf{v}_2(\epsilon)}\mathbf{v}_3(\epsilon)$).

Similarly, we have $\mathbf{u}_1'(\epsilon)$ as the unit vector along $\widehat{\mathbf{u}_2'(\epsilon)}\mathbf{u}_3'(\epsilon)$. The error

$$\|\mathbf{u}_1'(\epsilon) - \mathbf{u}_1(\epsilon)\| = \|\widehat{\mathbf{u}_2'(\epsilon)}\mathbf{u}_3'(\epsilon) - \widehat{\mathbf{u}_2(\epsilon)}\mathbf{u}_3(\epsilon)\|$$

can be shown to have the same order as $\|\widetilde{E}'(\epsilon) - E(\epsilon)\|$. The same argument applies for the error

$$\|\mathbf{v}_1'(\epsilon) - \mathbf{v}_1(\epsilon)\|.$$

Hence, we have proven the result. □ □

Finally, we discuss the error in the rotation matrix.

Proposition A.10. *Let the rotation matrices associated with $E(\epsilon)$ and $\widetilde{E}'(\epsilon)$ be denoted as $R(\epsilon)$ and $R'(\epsilon)$ respectively. Then the Frobenius norm of the error in estimating the rotation matrix is of the same order as $\|\widetilde{E}'(\epsilon) - E(\epsilon)\|$.*

Proof. The rotation matrices associated with $E(\epsilon)$ and $\widetilde{E}'(\epsilon)$ are $R(\epsilon)$ and $R'(\epsilon)$ respectively. Using the algorithm in [40], $R(\epsilon)$ and $R'(\epsilon)$ are given by

$$\begin{aligned} R(\epsilon) &= U(\epsilon)W(V(\epsilon))^T \\ R'(\epsilon) &= U'(\epsilon)W(V'(\epsilon))^T \end{aligned} \tag{A.8}$$

where W may take the form

$$\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

The correct form of W can be identified by enforcing the positive depth constraint.

Assume that we have identified the true W and we call it W_0. Using Equation (A.8), the difference between $R(\epsilon)$ and $R'(\epsilon)$ is given by (ϵ is again suppressed temporarily)

$$\begin{aligned} \|R - R'\| &= \|U'W_0V'^T - UW_0V^T\| \\ &\leq \|U'W_0V'^T - U'W_0V^T\| + \|U'W_0V^T - UW_0V^T\| \\ &\leq \|U'\|\ \|W_0\|\ \|V'^T - V^T\| + \|U' - U\|\ \|W_0\|\ \|V^T\| \end{aligned}$$

which has the same order as $\|\widetilde{E}'(\epsilon) - E(\epsilon)\|$. □ □

Appendix B

Proofs related to Chapter 3

This appendix deals with how the smoothness function $\Psi(\mathfrak{B})$ can be simplified into a more tractable form for the minimization process. In particular, we want to show that at the minima of $A(\mathbf{B}, \mathbf{F})$, $\Psi(\mathfrak{B})$ is related to $\mathfrak{B}$ and $\mathfrak{B}_{\mathbf{0}}$ by $\Psi(\mathfrak{B}) = tr(\Gamma \mathbf{G}^{-1} \Gamma^T)$.

At the minima, the derivative of equation (4.6) with respect to the velocity field expressed in the Fourier domain $v'(.)$ must be zero. Hence, utilizing the Fourier

transform relation, $v(\beta_{0i}) = \int_{\Re^2} v'(s) e^{2\pi k <\beta_{0i}, s>} ds$, we obtain the constraint

$$
\begin{aligned}
&\frac{\partial A(v', \mathbf{F})}{\partial v'(z)} = \\
&- \sum_{j=1}^{N} \frac{\sum_{i=1}^{M} \left(\frac{1}{\sigma_t^2} (\beta_i - \hat{t}_{0j}) \right) g(t_{0j} - b_i, \sigma_t) \int_{\Re^2} \frac{\partial v'(s)}{\partial v'(z)} e^{2\pi k <\beta_{0i}, s>} ds}{\sum_{i=1}^{M} g(t_{0j} - b_i, \sigma_t)} \\
&+ \sum_{i \in inlier} \frac{1}{\sigma_b^2} l_i l_i^T (\beta_i - r_i) \int_{\Re^2} \frac{\partial v'(s)}{\partial v'(z)} e^{2\pi k <\beta_{0i}, s>} ds \\
&+ \lambda \int_{\Re^2} \frac{\partial}{\partial v'(z)} \frac{|v'(s)|^2}{g'(s) + \kappa'(s)} ds \\
&= - \sum_{j=1}^{N} \frac{\sum_{i=1}^{M} \left(\frac{1}{\sigma_t^2} (\beta_i - \hat{t}_{0j}) \right) g(t_{0j} - b_i, \sigma_t) e^{2\pi k <\beta_{0i}, z>}}{\sum_{i=1}^{M} g(t_{0j} - b_i, \sigma_t)} \\
&+ \sum_{i \in inlier} \frac{1}{\sigma_b^2} l_i l_i^T (\beta_i - r_i) e^{2\pi k <\beta_{0i}, z>} + 2\lambda \frac{v'(-z)}{g'(z) + \kappa'(z)}, \\
&= 0_{2 \times 1}
\end{aligned}
\tag{B.1}
$$

where $\hat{t}_{0j}$ denotes a two dimensional vector made of the first two elements of t_{0j}.

Simplifying equation (C.4), we obtain

$$
-2\lambda \sum_{i=1}^{M} w_i e^{2\pi k <\beta_{0i}, z>} + 2\lambda \frac{v'(-z)}{g'(z) + \kappa'(z)} = 0
$$

where the two dimensional vectors w_i act as placeholders for the more complicated terms in (C.4).

Substituting z with $-z$ into the preceding equation and making some minor rearrangements, we have

$$v'(z) = (g'(-z) + \kappa'(-z)) \sum_{i=1}^{M} w_i e^{-2\pi k < \beta_{0i}, z >}. \tag{B.2}$$

where the two dimensional vectors, w_i, can be considered as weights which parameterize the velocity field.

Using the inverse Fourier transform relation

$$\begin{aligned} &\int_{\Re^2} w_i^T w_j (g'(z) + \kappa'(z)) e^{+2\pi k < \beta_{0j} - \beta_{0i}, z >} dz \\ &= w_i^T w_j \left(g(\beta_{0j} - \beta_{0i}, \gamma) + \kappa(\beta_{0j} - \beta_{0i}) \right), \end{aligned}$$

and equation (C.5), we can rewrite the regularization term of equation (4.6) as

$$\begin{aligned} \Psi(\mathfrak{B}) &= \int_{\Re^2} \frac{(v'(z))^T (v'(z))^*}{g'(z) + \kappa'(s)} dz \\ &= \int_{\Re^2} \frac{(g'(z) + \kappa'(s))^2 \sum_{i=1}^{M} \sum_{j=1}^{M} w_i^T w_j e^{+2\pi k < \beta_{0j} - \beta_{0i}, z >}}{g'(z) + \kappa'(s)} dz \\ &= \sum_{i=1}^{M} \sum_{j=1}^{M} \int_{\Re^2} w_i^T w_j (g'(z) + \kappa'(s)) e^{+2\pi k < \beta_{0j} - \beta_{0i}, z >} dz \\ &= tr(\mathbf{W^T G W}), \end{aligned} \tag{B.3}$$

where * represents the complex conjugate operation, $tr(.)$ represents the trace of a matrix, and

$$\mathbf{W}_{M \times 2} = [w_1, ..., w_M]^T,$$

$$\mathbf{G}(i, j) = g(\beta_{0i} - \beta_{0j}, \gamma) + \kappa(\beta_{0i} - \beta_{0j}).$$

If, as in the main text, one takes $\kappa(.)$ to be a function with spatial support less than the smallest separation between two feature points in $\mathbf{B_0}$, the above expression for $\mathbf{G}(i,j)$ can be simplified into

$$\mathbf{G}(i,j) = \begin{cases} g(\beta_{0i} - \beta_{0j}, \gamma) + k, & i = j \\ g(\beta_{0i} - \beta_{0j}, \gamma), & i \neq j \end{cases} \tag{B.4}$$

where k is some pre-determined constant.

Lastly, taking the inverse Fourier transform of equation (C.5), we obtain

$$\begin{aligned} v(z) &= (g(z,\gamma) + \kappa(z)) * \sum_{i=1}^{M} w_i \delta(z - \beta_{0i}) \\ &= \sum_{i=1}^{M} w_i (g(z - \beta_{0i}, \gamma) + \kappa(z - \beta_{0i})). \end{aligned}$$

where δ is the Dirac delta. Hence,

$$\mathfrak{B} - \mathfrak{B}_\mathbf{0} = \mathbf{GW}. \tag{B.5}$$

Substituting equation (C.8) into (C.6), we see that the regularization term $\Psi(\mathfrak{B})$, has the simplified form used in the main text

$$\Psi(\mathfrak{B}) = tr(\mathbf{W^T GW}) = tr((\mathfrak{B} - \mathfrak{B}_\mathbf{0})^T \mathbf{G}^{-1} (\mathfrak{B} - \mathfrak{B}_\mathbf{0})). \tag{B.6}$$

Appendix C

Proofs related to Chapter 4

C.1 Minimization of Smoothly varying Affine field

We follow a minimization procedure which computes an $\mathbf{A}^{k+1}$, using the $M \times 6$ linear equations (4.8) defined by $\mathbf{A}^k$. Compared to $\mathbf{A}^k$, $\mathbf{A}^{k+1}$ lowers the overall cost cost defined in equation (4.6). The process is iterated until convergence.

Copying the main bodies functional definition of ϕ, we have

$$\begin{aligned} \phi_{ij}(\mathbf{b}_i, \mathbf{t}_{0j}) &= g(\mathbf{t}_{0j} - \mathbf{b}_i, \sigma_t), \\ \overline{\phi_{ij}}(\mathbf{A}, \mathbf{t}_{0j}) &= \frac{\phi_{ij}(\mathbf{b}_i, \mathbf{t}_{0j})}{\sum_l \phi_{lj}(\mathbf{b}_l, \mathbf{t}_{0j}) + 2\kappa\pi\sigma_t^2}. \end{aligned} \tag{C.1}$$

Note that the second functions argument is given as $\mathbf{A}$ because, as can be seen from equation (4.2) $\mathbf{b}_i$'s are the base features after being warped by $\mathbf{A}$ and are wholly dependent on the $\mathbf{A}$.

Using Jensen's inequality, we can write

$$\begin{aligned}
&E(\mathbf{A}^{k+1}) - E(\mathbf{A}^k) \\
\leq &- \sum_{j=1}^{N}\sum_{i=1}^{M} \overline{\phi_{ij}}(\mathbf{A}^k, \mathbf{t}_{0j}) log \frac{\phi_{ij}(\mathbf{b}_i^{k+1}, \mathbf{t}_{0j})}{\phi_{ij}(\mathbf{b}_i^{k}, \mathbf{t}_{0j})} + \lambda \left(\Psi(\mathbf{A}^{k+1}) - \Psi(\mathbf{A}^k) \right) \\
= &\Delta E(\mathbf{A}^{k+1}, \mathbf{A}^k).
\end{aligned}$$

From the above, we know $\Delta E(\mathbf{A}^k, \mathbf{A}^k) = 0$. Hence, an $\mathbf{A}^{k+1}$ which minimizes $\Delta E(\mathbf{A}^{k+1}, \mathbf{A}^k)$ will ensure $E(\mathbf{A}^{k+1}) \leq E(\mathbf{A}^k)$.

Dropping all the terms in $\Delta E(\mathbf{A}^{k+1}, \mathbf{A}^k)$ which are independent of $\mathbf{A}^{k+1}$, we obtain a simplified cost function

$$Q = \frac{1}{2} \sum_{j=1}^{N}\sum_{i=1}^{M} \overline{\phi_{ij}}(\mathbf{A}^k, \mathbf{t}_{0j}) \frac{\left\| \mathbf{t}_{0j} - \mathbf{b}_i^{k+1} \right\|^2}{\sigma_t^2} + \lambda \Psi(\mathbf{A}^{k+1}).$$

Using a proof similar to that in Myronenko et al.[78], we show in the flooowing section that the regularization term $\Psi(\mathbf{A})$ has the simplified form $\Psi(\mathbf{A}) = tr(\Delta \mathbf{A}^T \mathbf{G}^{-1} \Delta \mathbf{A})$ where $\mathbf{G}(i,j) = g(\mathbf{b}_{0i(1:2)} - \mathbf{b}_{0j(1:2)}, \gamma)$. Substitute this definition of $\Psi(\mathbf{A})$ into Q, take partial differentiation of Q with respect to $\mathbf{A}^{k+1}$ and post multiply $\mathbf{G}$ throughout, we have

$$\begin{aligned}
\frac{\delta Q}{\delta \mathbf{A}^{k+1}} &= \begin{bmatrix} \mathbf{c}_1 & \mathbf{c}_2 & \dots & \mathbf{c}_M \end{bmatrix} + 2\lambda \left(\Delta \mathbf{A}^{k+1} \right)^T \mathbf{G}^{-1} \\
&= \mathbf{C} + 2\lambda \left(\Delta \mathbf{A}^{k+1} \right)^T \mathbf{G}^{-1} = \mathbf{0}_{6 \times M} \\
=> &\mathbf{CG} + 2\lambda \left(\Delta \mathbf{A}^{k+1} \right)^T = \mathbf{0}_{6 \times M},
\end{aligned} \tag{C.2}$$

$$\mathbf{c}_i = \sum_{j=1}^{N} \frac{\overline{\phi_{ij}}(\mathbf{A}^k, \mathbf{t}_{0j})}{\sigma_t^2} \mathbb{D}(\mathbf{b}_i^{k+1} - \mathbf{t}_{0j}) \mathbb{V}(\mathbf{b}_{0i}).$$

$\mathbb{D}(.), \mathbb{V}(.)$ are simultaneous truncation and tiling operators. They re-arrange **only the first two entries** of an input vector $\mathbf{z}$ (where $\mathbf{z}$ must have a length greater or equal to 2) to form the respective output matrices

$$\mathbb{D}(\mathbf{z})_{6\times 6} = \left[\begin{array}{c|c} \mathbf{z}_{(1)}\mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 3} \\ \hline \mathbf{0}_{3\times 3} & \mathbf{z}_{(2)}\mathbf{I}_{3\times 3} \end{array}\right]$$

$$\mathbb{V}(\mathbf{z})_{6\times 1} = \begin{bmatrix} \mathbf{z}_{(1)} & \mathbf{z}_{(2)} & 1 & \mathbf{z}_{(1)} & \mathbf{z}_{(2)} & 1 \end{bmatrix}^T$$

From the definition of $\mathbf{A}$ in (4.2), we know that $\mathbf{b}_i^{k+1}$ can be expressed as a linear combination of the entries of $\mathbf{A}^{k+1}$. Hence, equation (C.2) produces $M \times 6$ linear equations which can be used to estimate $\mathbf{A}^{k+1}$. $\mathbf{A}^{k+1}$ is used to estimate $\mathbf{A}^{k+2}$ and the process is repeated until convergence.

After convergence, the continuous stitching field $v(.)$ at any point $\mathbf{z}_{2\times 1}$ can be obtained from $\mathbf{A}$ using a weighted sum of Gaussian given by

$$\begin{aligned} \mathbf{W}_{M\times 6} &= [\mathbf{w}_1, ...\mathbf{w}_M]^T = \mathbf{G}^+ \Delta\mathbf{A}, \\ v(\mathbf{z}_{2\times 1}) &= \sum_{i=1}^{M} \mathbf{w}_i g(\mathbf{z} - \begin{bmatrix} \mathbf{b}_{0i(1)} & \mathbf{b}_{0i(2)} \end{bmatrix}^T, \gamma), \end{aligned} \tag{C.3}$$

where $\mathbf{G}^+$ is the pseudo-inverse of $\mathbf{G}$ and the $6 \times 1, \mathbf{w}_i$ vectors can be considered weights for the Gaussians. The detailed proof is given in the folowing section

C.2 Affine Smoothness

This section deals with how the affine smoothness function can be simplified into a more computationally tractable form. This proof is similar to that used in Chapter 3, with minor modifications to adapt the formulation from 2 to 6 dimensions.

At the minima, the derivative of the energy term in (4.6) with respect to the stitching field $v'(.)$, must be zero. Hence, utilizing the fourier transform relation, $(\Delta\mathbf{a}_i)_{6\times1} = v(\mu_i) = \int_{\mathbb{R}^2} v'(\omega)e^{2\pi\iota<\mu_i,\omega>}d\omega$, where $\mu_i = \begin{bmatrix} \mathbf{b}_{0i(1)} & \mathbf{b}_{0i(2)} \end{bmatrix}^T$, we obtain the constraint

$$\begin{aligned}
&\frac{\delta E(v')}{\delta v'(\mathbf{z})} = \mathbf{0}_{6\times1}, \forall \mathbf{z} \in \mathbb{R}^2 \\
&-\sum_{j=1}^{N} \frac{\sum_{i=1}^{M} \left(\frac{g(\mathbf{t}_{0j} - \mathbf{b}_i, \sigma_t)}{\sigma_t^2} \right) diag\left(\mathbb{D}(\mathbf{b}_i - \mathbf{t}_{0j})\mathbb{V}(\mathbf{b}_{0i})\right) \int_{\mathbb{R}^2} \frac{\delta v'(\omega)}{\delta v'(\mathbf{z})} e^{2\pi\iota<\mu_i,\omega>} d\omega}{\sum_{i=1}^{M} g(\mathbf{t}_{0j} - \mathbf{b}_i, \sigma_t) + 2\kappa\pi\sigma_t^2} + \lambda \int_{\mathbb{R}^2} \frac{\delta}{\delta v'(\mathbf{z})} \frac{|v'(\omega)|^2}{g'(\omega)} d\omega = \mathbf{0}_{6\times1} \\
&-\sum_{j=1}^{N} \frac{\sum_{i=1}^{M} \left(\frac{g(\mathbf{t}_{0j} - \mathbf{b}_i, \sigma_t)}{\sigma_t^2} \right) diag\left(\mathbb{D}(\mathbf{b}_i - \mathbf{t}_{0j})\mathbb{V}(\mathbf{b}_{0i})\right) e^{2\pi\iota<\mu_i,\mathbf{z}>}}{\sum_{i=1}^{M} g(\mathbf{t}_{0j} - \mathbf{b}_i, \sigma_t) + 2\kappa\pi\sigma_t^2} + 2\lambda \frac{v'(-\mathbf{z})}{g'(\mathbf{z})} = \mathbf{0}_{6\times1}
\end{aligned} \tag{C.4}$$

$\mathbb{D}(.), \mathbb{V}(.)$ are simultaneous truncation and tiling operators. They re-arrange **only the first two entries** of an input vector $\mathbf{z}$ (where $\mathbf{z}$ must have a length greater or equal to 2) to respectively form the 6×6 and 6×1 output matrices

$$\mathbb{D}(z)_{6\times6} = \left[\begin{array}{c|c} \mathbf{z}_{(1)}\mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \hline \mathbf{0}_{3\times3} & \mathbf{z}_{(2)}\mathbf{I}_{3\times3} \end{array}\right]$$

$$\mathbb{V}(z)_{6\times1} = \begin{bmatrix} \mathbf{z}_{(1)} & \mathbf{z}_{(2)} & 1 & \mathbf{z}_{(1)} & \mathbf{z}_{(2)} & 1 \end{bmatrix}^T$$

$diag(.)$ is a diagonalization operator which converts a k dimensional vector $\mathbf{z}$ into a diagonal matrix, such that

$$diag(\mathbf{z}_{k\times 1}) = \begin{bmatrix} \mathbf{z}_{(1)} & 0 & \cdots & 0 \\ 0 & \mathbf{z}_{(2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{z}_{(k)} \end{bmatrix}_{k\times k}.$$

Simplifying eqn (C.4), we obtain

$$-2\lambda \sum_{i=1}^{M} \mathbf{w}_i e^{2\pi\iota<\mu_i,\mathbf{z}>} + 2\lambda \frac{v'(-\mathbf{z})}{g'(\mathbf{z})} = 0$$

where the six dimensional vectors $\mathbf{w}_i$ act as placeholders for the more complicated terms in (C.4).

Substituting $\mathbf{z}$ with $-\mathbf{z}$ into the preceding equation and making some minor rearrangements, we have

$$v'(\mathbf{z}) = g'(-\mathbf{z}) \sum_{i=1}^{M} \mathbf{w}_i e^{-2\pi\iota<\mu_i,\mathbf{z}>}. \tag{C.5}$$

where the six dimensional vectors, $\mathbf{w}_i$, can be considered as weights which parameterize the stitching field.

Using the inverse Fourier transform relation

$$\int_{\mathbb{R}^2} \mathbf{w}_i^T \mathbf{w}_j g'(\mathbf{z}) e^{+2\pi\iota<\mu_j-\mu_i,\mathbf{z}>} d\mathbf{z} = \mathbf{w}_i^T \mathbf{w}_j g(\mu_j - \mu_i, \gamma),$$

and eqn (C.5), we can rewrite the regularization term of eqn (4.6) as

$$\begin{aligned}\Psi(\mathbf{A}) &= \int_{\mathbb{R}^2} \frac{(v'(\mathbf{z}))^T (v'(\mathbf{z}))^*}{g'(\mathbf{z})} d\mathbf{z} \\ &= \int_{\mathbb{R}^2} \frac{g'(\mathbf{z})^2 \sum_{i=1}^{M} \sum_{j=1}^{M} \mathbf{w}_i^T \mathbf{w}_j e^{+2\pi\iota<\mu_j-\mu_i,\mathbf{z}>}}{g'(\mathbf{z})} d\mathbf{z} \\ &= \sum_{i=1}^{M} \sum_{j=1}^{M} \int_{\mathbb{R}^2} \mathbf{w}_i^T \mathbf{w}_j g'(\mathbf{z}) e^{+2\pi\iota<\mu_j-\mu_i,\mathbf{z}>} d\mathbf{z} \\ &= tr(\mathbf{W^T G W}),\end{aligned} \tag{C.6}$$

where

$$\mathbf{W}_{M\times 6} = [\mathbf{w}_1, ..., \mathbf{w}_M]^T,$$

$$\mathbf{G}(i,j) = g(\mu_i - \mu_j, \gamma).$$

Taking the inverse Fourier transform of eqn (C.5), we obtain

$$v(\mathbf{z}) = g(\mathbf{z},\gamma) * \sum_{i=1}^{M} \mathbf{w}_i \delta(\mathbf{z}-\mu_i) = \sum_{i=1}^{M} \mathbf{w}_i g(\mathbf{z}-\mu_i,\gamma). \tag{C.7}$$

As $\Delta\mathbf{a}_j = v(\mu_j)$,

$$\Delta\mathbf{A} = \mathbf{GW}. \tag{C.8}$$

Substituting eqn (C.8) into (C.6), we see that the regularization term $\Psi(\mathbf{A})$, has the simplified form used in the main body

$$\Psi(\mathbf{A}) = tr(\mathbf{W^T G W}) = tr(\Delta\mathbf{A}^T \mathbf{G}^{-1} \Delta\mathbf{A}). \tag{C.9}$$

It can also be seen from eqn (C.8) that the stitching field $v(.)$ can be defined in

terms of $\mathbf{A}$. This is done by using the matrices $\Delta\mathbf{A}, \mathbf{G}$ to compute the weighting matrix $\mathbf{W}$ via,

$$\mathbf{W} = \mathbf{G}^{+}\Delta\mathbf{A}. \tag{C.10}$$

Using equation (C.7), we can then define the stitching field at any point $\mathbf{z}_{2\times 1}$.

Bibliography

[1] A. Agarwala, M. Dontcheva, M. Agrawal, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen. Interactive digital photomontage. *ACM Trans. Graph.*, 2004.

[2] S. Bae, A. Agarwala, and F. Durand. Computational rephotography. *ACM Trans. Graph.*, 2010.

[3] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski. Database and Evaluation Methodology for Optical Flow. *In Proc. Int'l Conf. on Computer Vision*, 2007.

[4] A. Bartoli and P. Sturm. The 3d line motion matrix and alignment of line reconstructions. *In Proc. of Computer Vision and Pattern Recognition*, 2001.

[5] A. Bartoli and P. Sturm. Structure-from-motion using lines: Representation, triangulation, and bundle adjustment. *Computer Vision and Image Understanding*, 2005.

[6] L.Baumela, L.Agapito, I.Reid, and P.Bustos. Motion Estimation Using the Differential Epipolar Equation. *In Proc. of Computer Vision and Pattern Recognition*,2000.

[7] H. Bay, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. *In Proc. European Conf. on Computer Vision*, 2006.

[8] V.G Bellile, A. Bartoli and P.Sayd. Deformable Surface Augmentation in spite of Self-Occlusions. *In Proc. of International Symposium on Mixed and Augmented Reality*, 2007.

[9] P. Besl and N. MacKay. A method for registration of 3-d shapes. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[10] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Trans. Pattern Analysis and Machine Intelligence*,1989.

[11] M.J. Brooks, W. Chojnacki, and L. Baumela. Determining the Egomotion of an Uncalibrated Camera from Instantaneous Optical Flow. *Journal Optical Soc. America A*, 1997.

[12] M. Brown and D. Lowe. Automatic panoramic image stitching using invariant features. *Int'l Journal of Computer Vision*, 2007.

[13] A.Bruhn, J.Weickert, and C.Schnörr. Lucas/Kanade Meets Horn/Schunck: Combining Local and Global Optic Flow Methods. *Int'l Journal of Computer Vision*, 2005.

[14] T.Brox and J.Malik1. Large Displacement Optical Flow: Descriptor Matching in Variational Motion Estimation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2010.

[15] R. Carroll, M. Agrawala, and A. Agarwala. Optimizing content-preserving projections for wide-angle images. *ACM Trans. Graph.*, 2009.

[16] L. H. Chan. and A. A. Efros. Automatic generation of an infinite panorama. *Technical Report, Carnegie Mellon University*, 2007.

[17] S.Z.Chang. Epipolar parameterization for reconstructing 3d rigid curve. *Pattern Recognition*, 1997.

[18] A. Chiuso, R. Brockett, and S. Soatto. Optimal Structure From Motion: Local Ambiguities And Global Estimates. *Int'l Journal of Computer Vision*, 2000.

[19] W. Chojnacki, M. J. Brooks, A. van den Hengel, and D. Gawley. Revisiting Hartley's Normalised Eight-Point Algorithm. *IEEE Trans. Pattern Analysis and Machine Intelligence*,2003.

[20] S. Christian and S. Christoph. Probabilistic Subgraph Matching Based on Convex Relaxation. *Energy Minimization Methods in Computer Vision and Pattern Recognition*, 2005.

[21] H. Chui and A. Rangarajan. A new algorithm for non-rigid point matching. *In Proc. of Computer Vision and Pattern Recognition*, 2000.

[22] K. Daniilidis and M.E. Spetsakis. Understanding Noise Sensitivity in Structure from Motion. In *Visual Navigation: From Biological Systems to Unmanned Ground Vehicles*, Y. Aloimonos (Ed.), Lawrence Erlbaum Assoc. Pub. 1997.

[23] P. David, D. Dementhon, R. Duraiswami, and H. Samet. Simultaneous pose and correspondence determination using line features. *Int'l Journal of Computer Vision*, 2002.

[24] F. Dellaert, S. Seitz, C. Thorpe, and S. Thurn. Structure from motion without correspondence. *In Proc. of Computer Vision and Pattern Recognition*, 2000.

[25] F. Dornaika and R. Chung. Mosaicking images with parallax. *Signal Processing: Image Communication*, 2004.

[26] C. Engels, H. Stewenius, and D. Nister. Bundle adjustment rules. In *Photogrammetric Computer Vision*, 2006.

[27] O. Enqvist, and F. Kahl. Robust Optimal Pose Estimation. *In Proc. European Conf. on Computer Vision*, 2008.

[28] O. Enqvist, and F. Kahl. Two view geometry estimation with outliers. *British Conference on Machine Vision*, 2009.

[29] O. Faugeras and B. Mourrain. On the geometry and algebra of the point and line correspondences between n images. *In Proc. Int'l Conf. on Computer Vision*, 1995.

[30] C. Fermüller, "Passive Navigation as a Pattern Recognition Problem," *Int'l Journal of Computer Vision*, 1995.

[31] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 1981.

[32] Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. *In Proc. of Computer Vision and Pattern Recognition*, 2007.

[33] P. Georgel, A. Bartoli, and N. Navab. Simultaneous In-Plane Motion Estimation and Point Matching Using Geometrical Cues Only In *Workshop on Motion and Video Computing*, 2009.

[34] G. L. Gimel'farb and J. Q. Zhang. Initial matching of multiple-view images by affine approximation of relative distortions. *Proc of International Workshops on Advances in Pattern Recognition*, 2000.

[35] F. Girosi, M. Jones, and T. Poggio. Regularization theory and neural networks architectures. *Neural Computation*, 1995.

[36] C. Glasbey and K. Mardia. A review of image warping methods. *Journal of Applied Statistics*, 1998.

[37] L. Goshen and I. Shimshoni. Balanced exploration and exploitation model search for efficient epipolar geometry estimation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2008.

[38] C. Harris and M. Stephens. A combined corner and edge detector. *In Proc. Alvery Vision Conference*, 1988.

[39] R. Hartley. In Defense of the Eight-Point Algorithm. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1997.

[40] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[41] D.J.Heeger, J. Kosecka, and S. Sastry. Subspace Methods for Recovering Rigid Motion I: Algorithm and Implementaions. *Int'l Journal of Computer Vision*, 1992.

[42] H. T. Ho and R. Goecke. Optical Flow Estimation Using Fourier Mellin Transform. *In Proc. of Computer Vision and Pattern Recognition*, 2008.

[43] B.K.P. Horn and B. Schunck. Determining Optical Flow. *Artificial Intelligence*, 1981.

[44] B.K.P. Horn and E.J. Weldon Jr.. Direct Method for Recovering Motion. *Int'l Journal of Computer Vision*, 2:51–76, 1988.

[45] S. Hou, Ramani, and Karthik. Structure-oriented contour representation and matching for engineering shapes. *Comput. Aided Des.*, 40(1):94–108, 2008.

[46] T. Igarashi, T. Moscovich, and J. F. Hughes. As-rigid-as-possible shape manipulation. *ACM Trans. Graph.*, 2005.

[47] H. Jiang and S.X Yu. Linear solution to scale and rotation invariant object matching. In *In Proc. of Computer Vision and Pattern Recognition*,2009.

[48] F. Kahl, S. Agarwal, M.K. Chandraker, D. Kriegman, S. Belongie. Practical Global Optimization for Multiview Geometry. *Int'l Journal of Computer Vision*, 2008.

[49] K.Kanatani. 3d Interpretation Of Optical Flow By Renormalization. *Int'l Journal of Computer Vision*,1993.

[50] G. Klein and T. Drummond. Robust visual tracking for non-instrumented augmented reality. In *International Symposium on Mixed and Augmented Reality*, 2003.

[51] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *International Symposium on Mixed and Augmented Reality*, 2007.

[52] J. Kopf, B. Chen, R. Szeliski, and M. Cohen. Street slide: Browsing street level imagery. *ACM Trans. Graph.*, 2010.

[53] P. D. Kovesi. MATLAB and Octave functions for computer vision and image processing. School of Computer Science & Software Engineering, The University of Western Australia. Available from: <http://www.csse.uwa.edu.au/∼pk/research/matlabfns/>.

[54] S. Lehman, A. P. Bradley, I. V. L. Clarkson, J. Williams, and P. J. Kootsookos. Correspondence-free determination of the affine fundamental matrix. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2007.

[55] V. Lempitsky, S. Roth, and C. Rother. Fusionflow: Discrete-Continuous Optimization for Optical Flow Estimation. *In Proc. of Computer Vision and*

Pattern Recognition, 2008.

[56] H. Li and R. Hartley. The 3d-3d registration problem revisited. In *In Proc. Int'l Conf. on Computer Vision*, 2007.

[57] W.Y. Lin, G.C. Tan, L.F. Cheong, C.H. Yan. When Discrete Meets Differential Assessing the Stability of Structure from Small Motion. *Int'l Journal of Computer Vision*, 2009.

[58] W.Y. Lin, G. Dong, P. Tan, L.F. Cheong, C.H. Yan. Simultaneous Camera Pose and Correspondence Estimation in Cornerless Images. *In Proc. Int'l Conf. on Computer Vision*, 2009.

[59] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman. Sift flow: Dense correspondence across different scenes. *In Proc. European Conf. on Computer Vision*, 2008.

[60] F. Liu, M. Gleicher, H. Jin, and A. Agarwala. Content-preserving warps for 3d video stabilization. *ACM Trans. Graph.*, 2009.

[61] H. C. Longuet-Higgins. A Computer Algorithm for Reconstructing a Scene from Two Projections. *Nature*, 1981.

[62] H. C. Longuet-Higgins and K. Prazdny. The Interpretation of a Moving Retinal Image. *Proceedings of the Royal Society of London*, Series B, 1980.

[63] M. Lourakis and A. Argyros. The design and implementation of a generic sparse bundle adjustment software package based on the Levenberg-Marquardt algorithm. *Technical report, Institute of Computer Science - FORTH, Heraklion, Crete, Greece*, 2004. Available from: < http://www.ics.forth.gr/ lourakis/sba>.

[64] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int'l Journal of Computer Vision*,2004.

[65] B. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. *Proceedings of DARPA Image Understanding Workshop*, 1981.

[66] Q.T. Luong and O. Faugeras. The Fundamental Matrix: Theory, Algorithms and Stability Analysis. *Int'l Journal of Computer Vision*, 1996.

[67] Y. Ma, J. Kosecka, and S. Sastry. Linear Differential Algorithm for Motion Recovery: A Geometric Approach. *Int'l Journal of Computer Vision*, 2000.

[68] Y. Ma, J. Kosecka, and S. Sastry. Optimization Criteria, Sensitivity and Robustness of Motion and Structure Estimation. *Int'l Journal of Computer Vision*,2001.

[69] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry. *An Invitation to 3-D Vision*. Springer-Verlag, New York, 2003.

[70] A. Makadia, C. Geyer, and K. Daniilidis. Correspondence-free structure from motion. *International Journal of Computer Vision*, 2007.

[71] L. Masson, F. Jurie, and M. Dhome. Contour/texture approach for visual tracking. In *Scandinavian conference on Image analysis*, 2003.

[72] S. Maybank. *Theory of Reconstruction from Image Motion*. Springer-Verlag, Berlin, 1992.

[73] J. Meltzer and S. Soatto. Edge descriptors for robust wide-baseline correspondence. *In Proc. of Computer Vision and Pattern Recognition*, 2008.

[74] M. Muhlich and R. Mester. The Role of Total Least Squares in Motion Analysis. *In Proc. European Conf. on Computer Vision*, 1998.

[75] L. Moisan and B. Stival. A probabilistic criterion to detect rigid point matches between two images and estimate the fundamental matrix. In *Int'l Journal of Computer Vision*, 2004.

[76] J. Morel and G. Yu. ASIFT: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, 2009.

[77] E. Mouragnon, F. Dekeyser, P. Sayd, M. Lhuillier, and M. Dhome. Real time localization and 3d reconstruction. In *In Proc. of Computer Vision and Pattern Recognition*, 2006.

[78] A. Myronenko, X. Song, and M. Carreira-Perpinan. Non-rigid point set registration: Coherent point drift. *In Proc. Neural Information Processing Systems*, 2007.

[79] S. Negahdaripour. Critical surface pairs and triplets. *Int'l Journal of Computer Vision*, 1989.

[80] T. Nir, A. M. Bruckstein, and R. Kimmel. Over-Parameterized Variational Optical Flow. *Int'l Journal of Computer Vision*, 2007.

[81] A.Ohta. Uncertainty Models of the Gradient Constraint for Optical Flow Computation. *IEICE Trans. on Information and Systems*, 1996.

[82] D. Nister. An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2004.

[83] T. Pajdla and J. Matas. High Accuracy Optical Flow Estimation Based on a Theory for Warping. *In Proc. European Conf. on Computer Vision*, 2004.

[84] T. Papadopoulo and O. Faugeras. Computing structure and motion of general 3d rigid curves from monocular sequences of perspective images. *In Proc. European Conference on Computer Vision*, 1996.

[85] M. Pressigout and E. Marchand. A model free hybrid algorithm for real time tracking. In *International Conference on Image Processing*, 2005.

[86] Z. Qi and J. Cooperstock. Overcoming parallax and sampling density issues in image mosaicing of non-planar scenes. *In Proc. British Machine Vision Conference*, 2007.

[87] A. Rangarajan, H. Chui, and F. Bookstein. The softassign Procrustes matching algorithm. In *International Conference on Information Processing in Medical Imaging*, 1997.

[88] A. Rav-Acha, P. Kohli, C. Rother, and A. Fitzgibbon. Unwrap mosaics: a new representation for video editing. *ACM Trans. Graph.*, 2008.

[89] X. Ren. Local Grouping for Optical Flow. *In Proc. of Computer Vision and Pattern Recognition*, 2008.

[90] O. Ricardo, C. Joao, and X. Joao. Contour point tracking by enforcement of rigidity constraints. *3DIM : International Conference on 3-D Digital Imaging and Modeling*, 2005.

[91] P. Sand and S. J. Teller. Particle Video: Long-Range Motion Estimation Using Point Trajectories. *In Proc. of Computer Vision and Pattern Recognition*, 2006.

[92] Y. Sheikh, A. Hakeem, and M. Shah. On the direct estimation of the fundamental matrix. *In Proc. of Computer Vision and Image Processing*, 2007.

[93] M. E. Spetsakis and J. Aloimonos. Motion and structure from point and line matches. *In Proc. Int'l Conf. on Computer Vision*, 1987.

[94] R. Szeliski and R. Weiss. Robust shape recovery from occluding contours using a linear smoother. *Int'l Journal of Computer Vision*, 1993.

[95] R. Szeliski. Image alignment and stitching: A tutorial. *From Microsoft Research*, 2005.

[96] S.J. Timoner and D.M. Freeman. Multi-Image Gradient-Based Algorithms for Motion Estimation. *Optical Engineering*, 2001.

[97] P. Torr and D. Murray. The Development and Comparison of Robust Methods for Estimating the Fundamental Matrix. *Int'l Journal of Computer Vision*, 1997.

[98] L. Torresani, S. Kolmogorov and C. Rother. Feature Correspondence via Graph Matching: Models and Global Optimization. *In Proc. European Conf. on Computer Vision*, 2008.

[99] B.Triggs. Differential Matching Constraints. *In Proc. Int'l Conf. on Computer Vision*, 1999.

[100] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment - a modern synthesis. *Vision Algorithms: Theory and Practise*, 1999.

[101] K. Uno and H. Miike. A stereo vision through creating a virtual image using affine transformation. *MVA*, 1996.

[102] L. Vacchetti, V. Lepetit, and P. Fua. Combining edge and texture information for real-time accurate 3d camera tracking. In *International Symposium on Mixed and Augmented Reality*, 2004.

[103] L. Valgaerts, A. Bruhn, and J. Weickert. A variational model for the joint recovery of the fundamental matrix and the optical flow. *In Proc. of Pattern Recognition*, 2008.

[104] A. Verri and T. Poggio. Motion Field and Optical Flow: Qualitative Properties. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1989.

[105] T. Viéville and O. Faugeras. Motion Analysis with a Camera with Unknown and Possibly Varying Intrinsic Parameters. *In Proc. Int'l Conf. on Computer Vision*, 1995.

[106] J. H. Wilkinson. *The Algebraic Eigenvalue Problem*. Clarendon Press. Oxford, 1965.

[107] K.-Y. K. Wong and R. Cipolla. Structure and motion estimation from apparent contours under circular motion. *Image and Vision Computing*, 2001.

[108] K.-Y. K. Wong and R. Cipolla. Structure and motion from silhouettes. *In Proc. Int'l Conf. on Computer Vision*, 2001.

[109] T. Xiang and L.F. Cheong. Understanding the Behavior of SFM Algorithms: A Geometric Approach. *Int'l Journal of Computer Vision*, 2003.

[110] A. L. Yuille and N. M. Grywacz. The motion coherence theory. *In Proc. Int'l Conf. on Computer Vision*, 1988.

[111] Z. Zhang. Iterative point matching for registration of free-form curves. In *Int'l Journal of Computer Vision* , 2004.