Geometric Optimization for Computer Vision

Pei Yean Lee

A thesis submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy in Engineering
at the Department of Information Engineering
Research School of Information Sciences and Engineering
Australian National University

Canberra, April 2005

To Eng Hock Lee and Siew Guat Tan, my parents.

Statement of Originality

I hereby declare that this submission is my own work, in collaboration with others, while enrolled as a PhD student at the Department of Information Engineering, Research School of Information Sciences and Engineering, Australian National University. To the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgement has been made in the text.

Most of the technical discussions presented in this thesis are based on the following publications:

- P. Y. Lee, J. B. Moore, "Pose Estimation via Gauss-Newton-on-manifold Approach", *Proceedings of the 16th International Symposium on Mathematical Theory of Networks and Systems*, Leuven, Belgium, 2004.

- U. Helmke, K. Hüper, P. Y. Lee, and J. Moore, "Essential Matrix Estimation via Newton-type Methods", *Proceedings of the 16th International Symposium on Mathematical Theory of Networks and Systems*, Leuven, Belgium, 2004.

- P. Y. Lee, J. B. Moore, "Geometric Optimization for 3D Pose Estimation of Quadratic Surfaces", *Proceedings of the 38th Asilomar Conference on Signals, Systems and Computers*, California, United States, 2004.

- S. Krishnan, P. Y. Lee, J. B. Moore, S. Venkatasubramanian, "Global Registration of Multiple 3D Point Sets via Optimization-on-a-Manifold", *Proceedings of the 3rd Eurographics Symposium on Geometric Processing*, Vienna, Austria, 2005.

Acknowledgments

I express deep gratitude to my supervisor, Professor John Moore for being supportive and motivating throughout this research. His invaluable guidance and insights are very much appreciated.

My sincere thanks to Dr Knut Hüper for teaching me mathematical formalism and Dr Jochen Trumpf for his thoughtful comments on part of the work. I offer my thanks to Dr Robert Mahony and Dr Kim Blackmore for giving me the opportunity to tutor at undergraduate courses. It has been an enjoyable experience.

I am grateful to Professor Uwe Helmke and Professor Robert Elliot for their supports during my visit to University of Würzburg and University of Calgary. Thanks also to Professor Martin Buss for giving me the opportunity to present my work at Technical University of Berlin. Moreover, I am indebted to Dr Shankar Krishnan and Dr Suresh Venkatasubramanian for inviting me to AT&T Shannon Lab in United States and for their generous supports during my visit. My special thanks to departmental staff and fellow students for creating friendly and stimulating research environment.

I would like to thank authors of the prior work in which this thesis is build upon. I acknowledge the following institutions for their financial support that gave me the opportunity of international visits and collaborations,

- Australian National University,
- National ICT Australia Limited (NICTA),
- Australian-German Joint Research Collaboration Scheme,
- National Institute of Engineering and Information Sciences (NIEIS).

Finally, wordless thanks to my parents and siblings for their loves and encouragement that have sustained throughout this studies and my life.

Abstract

Drawing ideas from differential geometry and optimization, this thesis presents novel parameterization-based framework to address optimization problems formulated on a differentiable manifold. The framework views the manifold as a collection of local coordinate charts. It involves successive parameterizations of a manifold, carrying out optimization of the local cost function in parameter space and then projecting the optimal vector back to the manifold. Several algorithms based on this approach are devised and applied to four computer vision tasks involving recovering pose information from images. First, we cast 2D-3D pose estimation as an optimization problem on the intersection of the special orthogonal group and a cone. We move on to estimate the essential matrix by minimizing a smooth function over the essential manifold. This is followed by formulating the problem of locating quadratic surfaces as an optimization problem cast on the special Euclidean group. Last, we demonstrate how one could simultaneously register multiple partial views of a 3D object within a common coordinate frame by solving an optimization problem involving the N-fold product of the special orthogonal group with itself. A mathematical proof establishes the local quadratic convergent rate of the Newton-like algorithms. Simulation results demonstrate the robustness of techniques against measurement noise and / or occlusion. New closed form calculations for the problems serve as a good initial estimate for any iterative algorithm presented, and give exact solution in the noise free case. The algorithmic technique and mathematical insights developed appear applicable to many problems in computer vision, as well as in other areas.

Contents

List of Figures

Notations

$\mathbb{R}$	The real numbers.
$\mathbb{R}^n$	The set of all real $n \times 1$ vectors.
$\mathbb{R}^{m\times n}$	The set of all real $m \times n$ matrices.
O_n	The set of real $n \times n$ orthogonal matrices, $\mathcal{O}_n := \{X \in \mathbb{R}^{n\times n} \vert X^\top X = I_n\}$.
SO_n	The set of real $n \times n$ special orthogonal matrices, $SO_n := \{X \in \mathbb{R}^{n\times n} \vert X^\top X = I_n, \det(X) = 1\}$.
SE_n	The set of real $n \times n$ special Euclidean matrices, $SE_n := \left\{ X = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{(n+1)\times(n+1)} \,\middle\vert\, R \in SO_n,\ t \in \mathbb{R}^n \right\}.$
$\mathfrak{so}_n$	The set of real $n \times n$ skew symmetric matrices, $\mathfrak{so}_n := \{X \in \mathbb{R}^{n\times n} \vert X = -X^\top\}$.
$\mathbb{RP}^n$	The n-dimensional projective space, i.e. the set of lines through the origin in $\mathbb{R}^{n+1}$.
$\mathbb{S}^n$	The n-dimensional unit sphere in $\mathbb{R}^{n+1}$, $\mathbb{S}^n := \{x \in \mathbb{R}^{n+1} \vert x^\top x = 1\}$.
$\mathcal{E}$	The normalized essential manifold.
$T_x M$	The tangent space of a manifold M at $x \in M$.
$T_x^{\mathrm{aff}} M$	The affine tangent space of a manifold M at $x \in M$.
TM	The tangent bundle of a manifold M, the disjoint union of all tangent spaces.
$T^{\mathrm{aff}} M$	The affine tangent bundle of a manifold M, the disjoint union of all affine tangent spaces.
$[A, B]$	The Lie bracket of two matrices, $[A, B] = AB - BA$.
δ_{ij}	The Kronecker delta function, $\delta_{ij} = 1$ if $i = j$, $\delta_{ij} = 0$ if $i \neq j$.
I_n	The $n \times n$ identity matrix.
$A^\top$	The transpose of a matrix A.
$\Vert x \Vert$	The Euclidean norm of a vector $x \in \mathbb{R}^n$, $\Vert x \Vert = \sqrt{x^\top x}$.
$\Vert A \Vert$	The Frobenius norm of a matrix $A \in \mathbb{R}^{m\times n}$, $\Vert A \Vert = \sqrt{\mathrm{tr}(AA^\top)}$.
$\det(A)$	The determinant of matrix A.
$\mathrm{diag}(\alpha_1, \cdots, \alpha_n)$	The diagonal matrix with elements $(\alpha_1, \cdots, \alpha_n)$.
$\mathrm{vec}(A)$	The operation that creates a column vector by stacking each column of the matrix A below one another.
$\otimes$	Kronecker product: $A \otimes B$ is formed by replacing each element of A, denoted a_{ij}, by $a_{ij}B$. Recall, $(A \otimes B)^\top = A^\top \otimes B^\top$, $(A \otimes B)(C \otimes D) = (AB \otimes CD)$, $\mathrm{vec}(ABC) = (C^\top \otimes A)\mathrm{vec}(B)$, $(A \otimes B)^{-1} = (A^{-1} \otimes B^{-1})$ when the inverses exist.

Chapter 1

Introduction

Integrating ideas from differential geometry and optimization, this thesis proposes a novel geometrical framework to study optimization problems formulated on smooth manifolds, particularly those which arise in computer vision. In this chapter, we introduce the subject of computer vision and geometric optimization, including motivation for the study. Developments in the two fields are reviewed and research issues that arise are highlighted. Building upon these, we state the research goals and describe the proposed methodology. Finally, the scope and key assumptions of the research are put forward and an outline of the thesis is presented.

1.1 Computer Vision

As humans, our vision system allows us to understand the world that we see. Can computers have this capability? This becomes the central challenge of computer vision. Many years ago, computer vision was not widely used due to expensive hardware and sophisticated software which could hardly be used for real time applications. In recent years, computers with high speed processors and larger memories at reduced prices have emerged. This, together with cheap off-the-shelf digital imaging devices which can produce good quality images have made computer vision system increasingly used for various real world applications.

In robotics, computer vision have been used by industrial robots for parts assembly and by autonomous vehicles (mobile robots) as a sensor to navigate and localize in an environment. In the area of human computer interaction, computer vision algorithms are used to create a user friendly computer interface. They also find applications in the automobile industry. In January 2004, Toyota Motor Corporation offered an option package for its Prius model (a gas-electric hybrid car) that included a computer vision system together with steering sensors which could parallel park the car “automatically”. In the entertainment industry, computer vision techniques are adopted to construct 3D computer models of real objects or scenes to make animation in games or movies more natural and realistic.

All applications discussed above involve recovering pose information from images, one of the classical problems in computer vision. Pose refers to the knowledge about “what is the location of the object and how is the object oriented relative to the camera?” For example, mobile robots and intelligent cars need to know where the obstacle is in order to avoid it. Industrial robot needs to know the pose of the parts in order to pick them up and assemble them correctly. Computer software needs to know how the head of the user is oriented in order to decide whether to scroll the page up or down. To construct a 3D computer model, multiple partial views of the real object need to be taken and the unknown pose information between the views need to be recovered. These applications motivate us to revisit the pose estimation problem using mathematical tools from differential geometry and optimization theory and thereby develop new enhanced algorithms that carefully exploit the underlying geometry of the problem.

1.2 Geometric Optimization

Various computer vision problems, including pose estimation can be addressed using an optimization framework. Optimization is a branch of applied mathematics consisting of two main areas, namely numerical optimization and variational methods. Numerical optimization, also known as mathematical programming, is concerned with finding the optimal point that minimizes or maximizes a cost function, possibly subject to constraints. It can be classified further depending on the nature of the cost function and its constraints. On the other hand, variational methods are devoted to finding the optimal function that minimizes or maximizes a functional, possibly subject to constraints.

Traditionally, researchers have focused on developing techniques for unconstrained optimization problems, or equivalently those formulated on Euclidean space. When the problem is constrained, as often occurs in many computer vision tasks, the widely used strategy is to transform the constrained problem into an unconstrained problem using the method of Lagrange multipliers [68]. This approach is relatively simple and allows one to exploit the wealth of existing Euclidean optimization techniques. However, it merely treats constrained problem as a 'black box' and solves it using algebraic manipulation. Algorithms developed using this strategy only preserve the constraint assymptotically. Any premature termination of the algorithm may fail to satisfy the constraint set. Also, this approach always increases the dimensionality of the optimization.

An alternative strategy, known as geometric optimization, which we adopt in this thesis, is to exploit the geometry of the underlying parameter space. If the underlying parameter space is the Euclidean space, then Euclidean geometry such as distance and angle are exploited. This has been the main principle behind developing classical optimization techniques. On the other hand, if the underlying parameter space is non-Euclidean, as often occurs in constrained problems, then there can be advantage in exploiting the geometric structure of this space. That is, develop a strategy which views constrained problems as equivalent to unconstrained problems posed on the constraint sets. Unlike the traditional Lagrange multiplier method, this approach deals with the original problem directly and ensures that the constraints are fulfilled at every iteration. Philosophically, this approach is geometrically more intuitive, conceptually simpler and mathematically more elegant. From a practical perspective, geometric optimization ensures that the constraints are satisfied at every iteration. Thus, any premature termination still gives a meaningful solution.

In this thesis, we focus on optimization problems formulated on manifolds. Generically, a manifold is a space which 'locally looks like' Euclidean space but does not do so globally. For example, Euclidean space is itself a manifold, spheres are manifolds, as is the surface of the earth on which we live. Since often a manifold can be implicitly defined by a set of equality constraints, it deals with a rather large class of optimization problems, for background theory of manifolds, see [47].

1.3 Literature Review

The development of numerical optimization on manifolds can be traced back more than a century ago when the Rayleigh Quotient Iteration, or essentially the Newton's iterate on a sphere was first introduced to compute eigenvectors of a symmetric matrix [19]. In this section, we first review the classical Riemannian approach of generalizing Euclidean optimization techniques to a manifold setting before turning our attention to the relatively recent non-Riemannian approach.

Riemannian Approach

Recall that manifolds are 'locally' Euclidean but not globally so. Hence, it is not surprising that some nice Euclidean mathematical objects such as a straight line, gradient and Hessian of the cost function have their counterparts in a manifold setting. The Riemannian approach proposes the concept of endowing manifolds with a metric structure. This gives rise to the corresponding mathematical objects such as a geodesic, Riemannian gradient and Riemannian Hessian. Most existing numerical algorithms on manifolds have been proposed based on this approach. Each varies from others based on the type of manifold for which the algorithms is formulated and the Euclidean optimization technique that is being generalized, as discussed below.

Gradient method on manifolds. The gradient method is one of the most popular Euclidean optimization techniques to be generalized to a manifold due to its nice global convergence properties. The method was first conceptually extended to manifolds by Luenberger [56, 57] in 1972, see also the work by Gabay [26]. In the early 1990s, gradient methods on manifolds were popularized by researchers in the area of systems and control theory to address problems which arise in linear algebra and linear systems theory using a calculus approach. Some fundamental work in this area were carried out by Brockett [11], Helmke and Moore [32], Smith [78], and Mahony [59]. See also independent work by Udriste [83].

Newton's method on manifolds. The gradient method on manifolds, as for its Euclidean counterpart has a slow (linear) local convergence rate. Thus, it is not always suited for applications requiring online computation. As a result, many researchers turn their attention to the classical Newton's method which is well known for its fast (quadratic) local convergence rate. In 1982, Gabay [26] proposed Newton's method formulated on a submanifold of Euclidean space and each update is carried out along a geodesic. There is independent work by Shub [76] who defined Newton's method for finding the zeros of a vector field on a manifold. Other independent work to extend Newton's method for general optimization problems on Riemannian manifolds has been carried out by Smith [78] and Mahony [59, 60] restricting to compact Lie group, and by Udriste [83] restricting to convex optimization problems on Riemannian manifold. Edelman, Arias and Smith [20] developed Newton's method specific to orthogonality constraints, being then Stiefel and Grassmann manifolds. This method was later adapted by Ma, Košecká and Sastry [58] for solving the structure from motion problem in computer vision. There is also a recent paper by Adler and Dedieu [3] which studied Newton's method to find zero of a vector field defined on general Riemannian manifolds with applications in medicine.

Quasi-Newton method on manifolds. Newton's method requires computation of the inverse of a symmetric matrix containing second order local information of the cost function, known as the Hessian. However, in certain applications, it is time consuming to evaluate the Hessian inverse. To overcome this problem, in Euclidean space, a variant of Newton's method, known as the quasi-Newton method was introduced by Davidon [18] in late 1950s. The technique approximates Hessian inverse using first order gradient information and has a fast (superlinear) local convergence rate. To the best of our knowledge, the paper by Gabay [26] is the first and only published work focusing on generalizing the quasi-Newton method to a manifold.

Conjugate gradient method on manifolds. Quasi-Newton's method faces difficulties when there are large scale optimization problems with sparse Hessian matrices. This problem can be overcome using the conjugate gradient method. In Euclidean space, this technique was originally introduced by Hestenes and Stiefel [35] in the 1950s to solve large scale systems of linear equations. In 1960s, Fletcher and Reeves [24] extended the method to solve general unconstrained optimization problems. It was about 30 years later Smith [78] first generalized the method to manifolds. Since then, he also collaborated with Edelman and Arias [20] to study the method specifically on Grassmann and Stiefel manifolds.

Others. In Euclidean space, it is well known that there is a trade off between gradient methods (good global convergence properties but slow local convergence rate) and Newton's method (fast local convergence rate but poor global convergence properties). Two strategies have been presented to either globalize the Newton's method or speed up the gradient method, namely the line search and the trust region methods. Since both methods also exhibit similar properties in a manifold setting, Absil [1] has extended the trust region strategy to a manifold, while line search strategies on manifold can be found in the work of Udriste [83] and Yang [85].

Non-Riemannian Approach

The traditional Riemannian approach is somewhat natural and intuitive, and has since established its dominance in the field. Problems arise when it is difficult to find Riemannian metrics such as in the case for non-compact Lie groups [61]. As pointed out in [63], the Riemannian approach is justified when the cost function is somewhat related to the Riemannian geometry and works best when the cost function is convex with respect to Riemannian geometry. Moreover, many algorithms proposed based on this approach are either abstract conceptually and difficult to implement on a computer, or are computationally inefficient. Consequently, it is not suitable for many computer vision tasks which require online processing.

At the beginning of this research, we are unaware of any framework other than the Riemannian approach and those that exploits Lie group structures [61, 69]. Later, we became aware of two independent prior works which present a similar framework as ours: The unpublished report by Taylor and Kriegman [82] and the paper by Manton [62] to address problems arise in computer vision and signal processing, respectively.

Since our work, Hüper and Trumpf also present a related approach in a recent paper [41]. Since all these works, including the one proposed here, are somewhat similar, detailed discussions and comparisons will be deferred to a later chapter.

1.4 Research Issues

Having introduced the areas of study and reviewed its development, several research issues naturally arise. First, there exists a gap between mathematical development of the geometric optimization and its practical application. On the one hand, researchers develop rigorous abstract mathematical concepts using tools from differential geometry and optimization, that is less accessible to engineers and computer scientists. On the other hand, computer scientists and engineers are rather sceptical about the practicality of these concepts. Notice that the development of geometric optimization starts with theory in early 1970s. Abstract algorithms started to appear in the 1980s but its practical applications start only in early 1990s, mainly in the areas of linear algebra and linear systems theory [11, 32]. Later, only a handful of these algorithms find applications in robotics [12, 33], signal processing [20, 79], computer vision [58] and the medical area [3]. Although algorithms on manifolds have evolved from abstract notions [26, 78, 59] to concrete implementable recursions [20, 2, 58], it is still not suitable for many real time applications.

Second, the majority of iterative computer vision algorithms simply adopt standard optimization techniques without exploiting the geometric structure of the underlying parameter space.

1.5 Aims and Methodology

The aim of this thesis is to address the research issues just highlighted, giving rise to three specific objectives,

- To devise new iterative computer vision algorithms that exploits the geometric structure of the underlying parameter space, and explicitly preserve constraints at every iteration,

- To explore and extend a relatively new geometrical framework for optimization on manifolds that is simpler and more accessible to computer scientists and engineers,

- To develop the practical importance of geometric optimization algorithms as a tool that is not only implementable, but also suitable for real time applications, particularly in the area of Computer Vision.

To achieve these, we present and develop further a relatively new parameterization-based framework, focussing on the special class of problems arising in computer vision. Since a manifold 'locally looks like' Euclidean, this approach views a manifold as a collection of local coordinate charts. It works as follow: At each iterate, a local parameterization of the manifold at a point is constructed, the point is pulled back to Euclidean space via the parameterization. The optimal vector that maximizes or minimizes the cost function expressed in local parameter space is then pushed forward to the manifold via the same parameterization. Various classical optimization techniques are extended to manifolds using this framework and implemented to address four problems arising in computer vision under the theme of pose recovery.

1.6 Scope and Key Assumptions

In this report, we are mainly interested in recovering pose information from two commonly used sensor modalities in computer vision, namely the 2D intensity image and the 3D range image. The sensors are assumed to be calibrated. The input data from the sensors is modelled as a set of points. The proposed algorithms work with real-valued cost functions with at least twice continuous differentiability and the constraint set is a smooth manifold.

1.7 Outline of the Thesis

The thesis consists of an introduction, four technical chapters and a conclusion chapter. Aligned with our goal to make the subject of geometric optimization more accessible to readers with a background in engineering and computer science, we do not offer a systematic mathematical treatment of the proposed framework, rather we demonstrate the approach by practical problems arising in computer vision.

Chapter 2: 2D-3D Pose Estimation. In this chapter, we are interested in estimating the relative position and orientation between a 3D object and its projection on a 2D image plane from a set of corresponding points. The task is formulated as an optimization problem on the intersection of a rotation group and a cone constraint. Newton-type iterations based on a proposed geometrical framework are devised. A key feature of the proposed algorithms, not used in earlier studies, is an analytic geodesic search. This increases the probability of escape from local minima and convergence to a global minimum without the need to reinitialize the algorithm. For the case of finite number of local minima on the manifold, including periodically random line search directions then as the iterations become infinite, such escape can be achieved with probability one. We also present a new analytical method which in the noise free case, gives the exact solution and a good initial estimate for any iterative algorithm. The proposed algorithm is proved to be locally quadratically convergent. Simulations also suggests that the algorithms give lower parameter estimation errors than earlier methods and convergence to a global minimum occurs in typically 5–10 iterations.

Chapter 3: Estimation of Essential Matrix. It is well known in the computer vision literature that relative pose information between a pair of calibrated 2D images is encapsulated by the essential matrix. In this chapter, we focus on recovering this matrix by minimizing a smooth function over the essential manifold. We have implemented Newton-like optimization techniques using the proposed parameterization-based framework. The algorithm is proved to be locally quadratically convergent. The efficiency is demonstrated by means of computer simulations, where convergence typically occurs in 5–20 iterations.

Chapter 4: 3D Localization of Quadratic Surfaces. Quadratic surfaces are commonly occurring shapes in man made objects. This chapter addresses the problem of locating quadratic surfaces from 3D range image data as minimizing a smooth function over the special Euclidean group. The optimization is based on locally quadratically convergent Newton-type iterations on this constraint manifold. To achieve this, analysis of the underlying geometric constraint is required. Analysis shows that the proposed algorithm is relatively robust against additive Gaussian noise and occlusion.

Chapter 5: Global Registration of Multiple 3D Point Sets. To construct a 3D computer model of a real object, multiple partial views of the object need to be taken and relative pose between these partial views need to be recovered. This chapter demonstrates how one can simultaneously register multiple 3D point sets. We formulate the task as an unconstrained optimization problem on N-fold product of special orthogonal groups. The optimization is based on locally quadratically convergent Newton-type iterations on this constraint manifold. The proposed algorithm is fast, converges at a local quadratic rate, computation per iteration is low since the iteration cost is independent of the number of data points in each view. In addition, we present a new closed form calculation for the simultaneous registration of multiple point sets. In the noise free case, it gives correct registrations in a single step. In the presence of noise an additional projection step to the constraint manifold is required. This analytical solution is a useful initial estimate for any iterative algorithm.

The final chapter revisits the parameterization-based approaches, draws conclusions from the development and implementation of the proposed framework and identifies continuing research themes that have arisen out of this work. To avoid disturbing the flow of the report, most of the proofs, derivations and details of problems for future work have been relegated to Appendix.

1.8 Summary

This chapter outlines the basis of the thesis. It provides an overview of computer vision and geometric optimization, including the thrusts behind the study. Developments in the fields have been reviewed and research issues identified. Building on this, research goals has been stated and the proposed methodology briefly described. This has been followed by discussion of the research scope and summary of each chapter. Based on these, we are now ready to proceed to detailed descriptions of the thesis research.

Chapter 2

2D-3D Pose Estimation

As pointed out in the previous chapter, recovering the pose information from images plays a fundamental role in many computer vision tasks and its subsequent applications ranging from robotics to the entertainment industry. In this chapter, we are mainly interested in estimating the relative position and orientation between a 3D object and its 2D image from a set of point correspondences. With an appropriate choice of reference frame, the problem can be viewed as either determining the camera pose with respect to a known object frame or as estimating the object pose with respect to a known camera frame. It is also known as absolute orientation or extrinsic camera calibration. Most of the material in this chapter has been published in [48].

Existing Work There are many algorithms, but perhaps only two approaches for solving this problem presented in the literature. The first approach adopts linear methods that yield closed-form solutions [23, 73, 4]. These linear methods are very sensitive to noise because the orthonormality constraint of the rotation matrix is not taken into account, yet are simple enough to implement in online computation. The second approach is to formulate the task as a constrained nonlinear optimization problem [30, 53, 54, 55]. As opposed to linear methods, these iterative techniques provide numerical solutions that are more accurate and robust to measurement noise. Also, when Newton-based recursions are used, fast (quadratic) local convergence rate can be achieved. However, the methods rely on a good initial pose estimate. Also, without random reinitializations and selection of the lowest cost function outcome, such methods might converge to a local minimum which is not the global minimum, or converge to an infeasible solution, for which the object, or part thereof, is estimated to be behind the camera. Among the existing recursive algorithms, only the technique proposed by Lu, Hager and Mjolsness [55] adopts the geometric optimization framework.

Our Work In this chapter, the 2D-3D pose recovery task is formulated as minimizing a smooth function over the intersection of a rotation group and a cone, as depicted in Fig. 2.1. To address the problem, we introduce a novel framework based on successive parameterization of the constraint manifold. In contrast to the 'fast' linearly convergent geometric algorithm of [55], the Newton-like recursions devised based on this framework is locally quadratically convergent. The main novelty of our approach is the use of closed-form global geodesic search step, requires only the solution of a quartic equation. It assists in escaping local minimum and infeasible domain, as well as converges to a global minimum without the need to reinitialize the algorithm. The *Newton decrement* or its estimate is used as an indicator for selecting the optimal directions for a geodesic search, assists in the decision for selecting Newton step, and for algorithm termination. We also put forward a method for algorithm initialization, which leads to an exact solution in the noise free case, and a good initial estimate in the presence of noise. Simulation results suggests the proposed algorithms achieve significantly lower parameter estimation errors than techniques presented by [55] and convergence to a global minimum occurs in typically 5–10 iterations.

Chapter Outline In Section 2.1, the 2D-3D pose estimation task is formulated as an unconstrained optimization problem cast on a constraint manifold. We propose a parameterization-based geometric optimization algorithm to address the problem in Section 2.2. The analytical noise free solution and noisy initialization procedures are presented in Section 2.3. This is followed by an implementation summary in Section 2.4. Simulation studies is presented in Section 2.5. A modified version of the algorithm which takes into account outliers is described in Section 2.6. This is followed by a chapter summary in Section 2.7.

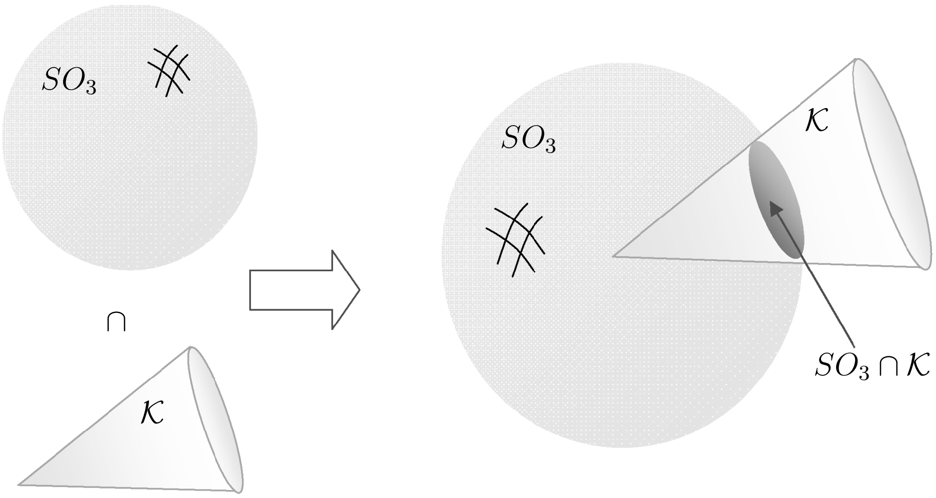


Figure 2.1: The special orthogonal group SO_3, the cone constraint $\mathcal{K}$, and the resulting constraint manifold $SO_3 \cap \mathcal{K}$, which is the intersection between SO_3 and $\mathcal{K}$.

2.1 Problem Formulation

In this section, we provide definitions of relevant terms before formulate the pose recovery task as an optimization problem on a constraint manifold.

2.1.1 Definitions

Model The model of a known 3D object is a set of points described in an object centered frame that lie within the field of view of a camera, as

$$\{m_i\}_{i=1,\cdots,n}, \quad m_i := [x_i \;\; y_i \;\; z_i]^\top \in \mathbb{R}^3.$$

Transformed model To represent each model point in the camera centered frame $m_i' := [x_i' \;\; y_i' \;\; z_i']^\top$, a rigid body transformation is performed as follows,

$$m_i' = Rm_i + t,$$

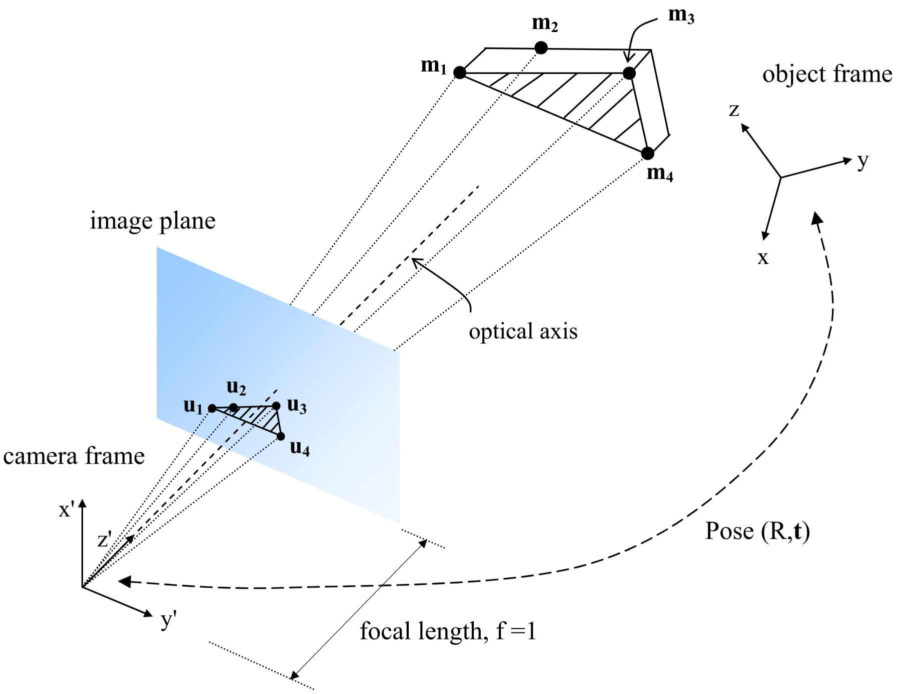


Figure 2.2: The 2D-3D pose estimation problem: given the model $\{m_1, \cdots, m_4\}$ expressed in object frame, and its corresponding image $\{u_1, \cdots, u_4\}$ expressed in camera frame, find the relative pose (R, t) between the object frame and the camera frame.

where $R \in SO_3$, i.e. $R^\top R = I,\ \det(R) = 1$, representing the rotation, and $t \in \mathbb{R}^3$ is the translation vector.

Image Each transformed model point m_i' is observed by a camera. The origin of the camera frame coincides with the center of projection, and the optical axis is along the positive z'-axis, as illustrated in Fig. 2.2. The point m_i' on the image plane is described in pixel coordinates, denoted p_i. Such an image point is then normalized by the camera calibration matrix F, assumed known, to obtain the corresponding normalized image point u_i, as

$$u_i = F^{-1} p_i.$$

The camera calibration matrix F is an upper triangular matrix consisting of intrinsic camera parameters, details can be found in [31]. Under full perspective projection, each point in the model m_i is related to its corresponding normalized image point by the following equation,

$$u_i = \frac{Rm_i + t}{z'_i}, \quad z'_i = e_3^\top (Rm_i + t), \quad e_3 := [0 \;\; 0 \;\; 1]^\top. \tag{2.1}$$

Here, $\{z'_i\}_{i=1,\cdots,n}$ are depth parameters which must satisfy the cone constraint $\mathcal{K}$, that is $\{z'_i > 0\}_{i=1,\cdots,n}$ to ensure that the estimated pose always locates the object in front of the camera.

2.1.2 Object Space Error

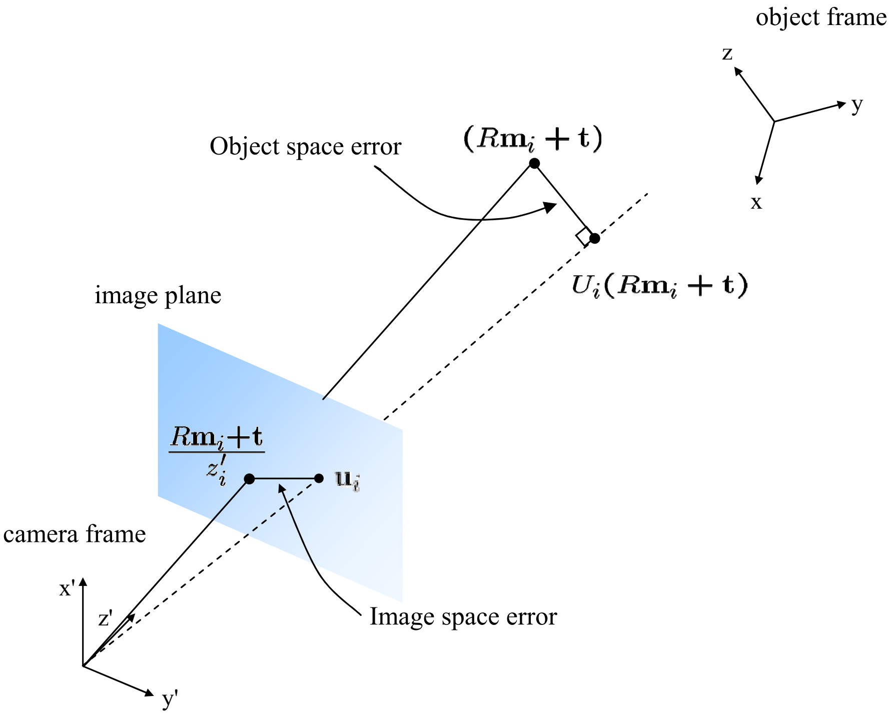

object frame
z
y
x
Object space error
$(R\mathbf{m}_i + \mathbf{t})$
$U_i(R\mathbf{m}_i + \mathbf{t})$
image plane
$\frac{R\mathbf{m}_i+\mathbf{t}}{z'_i}$
$\mathbf{u}_i$
camera frame
x'
z'
y'
Image space error

Figure 2.3: The image space error and the object space error.

Instead of recovering the pose $\{R, t\}$ using a least squares cost function penalizing the classical image space collinearity error via (2.1), we adopt the object space collinearity error introduced in [55], as depicted in Fig. 2.3. Let U_i be the projection operator,

$$U_i = U_i^\top := \frac{u_i u_i^\top}{u_i^\top u_i}, \quad U_i^2 = U_i. \tag{2.2}$$

For object space collinearity, the orthogonal projection of $m_i' = Rm_i + t$ on to the line-of-sight of the corresponding image point u_i should be equal to m_i' itself, as described by the following equation,

$$Rm_i + t = U_i(Rm_i + t), \quad i = 1, \cdots, n. \tag{2.3}$$

In the presence of pixel noise, the cost function penalizing the object space collinearity error is given as,

$$\begin{aligned} \phi : SO_3 \times \mathbb{R}^3 &\rightarrow \mathbb{R}, \\ \phi(R, t) &= \frac{1}{2}\sum_{i=1}^{n} \|(I - U_i)(Rm_i + t)\|^2. \end{aligned} \tag{2.4}$$

Observe that the cost function is quadratic in terms of errors, which are linear in the unknown pose parameters R, t. It is zero if and only if (2.3) is satisfied. The optimization is made nontrivial by the constraints that R is an element of the manifold SO_3 and the presence of a cone constraint $\mathcal{K}$ resulting from the requirement that the object must be in front of the camera. Actually optimizing quadratic cost functions on SO_3 is well studied, as is optimization on cones. What is interesting here from an optimization point of view is to tackle the much harder problem of optimizing a cost function on the intersection of SO_3 and a cone.

We have also applied the proposed algorithm to more conventional image space cost functions, but our simulations suggest that the pose estimates are not as good as those obtained using the object space cost function (2.4). It is of interest to ask: why does this cost function yield estimates of R, denoted $\hat{R}$, which are better estimates than obtained using other indices? Our conjecture is that it is as close as we can reasonable work with an index penalizing the term $\|R - \hat{R}\|^2$ or equivalently maximizing the term $\mathrm{tr}(R\hat{R}^\top)$.

2.1.3 Optimal Translation

We eliminate the translation vector t from the cost function (2.4) via least squares optimization to reduce the number of parameters for optimization. Note that this is not a noise-free elimination. It might appear that we formulate the problem as two-stage optimization, first optimize R then optimize t. However, by eliminating t via least squares, this two stage optimization actually gives optimal solutions as if we have solve it in one stage. Exploiting the fact that $(I-U_i)^\top(I-U_i)=(I-U_i)$, cost function (2.4) can be re-expressed as,

$$\phi=\frac{1}{2}\sum_{i=1}^{n}m_i^\top R^\top(I-U_i)Rm_i+t^\top\sum_{i=1}^{n}(I-U_i)(m_i^\top\otimes I)\mathrm{vec}(R)+\frac{1}{2}t^\top\sum_{i=1}^{n}(I-U_i)t. \quad (2.5)$$

Denoting

$$\widetilde{U}:=\sum_{i=1}^{n}(I-U_i),\ \ \mathcal{U}=\widetilde{U}^{-1}\sum_{i=1}^{n}(I-U_i)(m_i^\top\otimes I), \quad (2.6)$$

with R fixed, an optimal t that minimizes (2.5) is given by

$$t=-\mathcal{U}\mathrm{vec}(R), \quad (2.7)$$

Substituting (2.7) into (2.1), the depth parameters z_i' can be reformulated as,

$$z_i'=\mathcal{B}_i\,\mathrm{vec}(R),\ \ \mathcal{B}_i:=e_3^\top((m_i^\top\otimes I)-\mathcal{U}). \quad (2.8)$$

The cone constraint $\mathcal{K}$ can now be expressed in terms of $R\in SO_3$ as

$$\mathcal{K}:=\{R\mid\{\mathcal{B}_i\mathrm{vec}(R)>0\}_{i=1,2,\cdots,n}\}. \quad (2.9)$$

2.1.4 Cost Function Independent of Translation

Substituting (2.7) into (2.4), the cost function can be reformulated as,

$$f:SO_3\to\mathbb{R},\quad f(R)=\frac{1}{2}\|\mathcal{D}\mathrm{vec}(R)\|^2, \quad (2.10)$$

where

$$\mathcal{D}=\begin{bmatrix}D_1^\top & D_2^\top & \cdots & D_n^\top\end{bmatrix}^\top,\quad D_i=(I-U_i)\big((m_i^\top\otimes I)-\mathcal{U}\big). \quad (2.11)$$

Remark 2.1.1 *Note that the function*

$$g:\mathbb{R}^{3\times3}\to\mathbb{R},\quad g(X)=\frac{1}{2}\|\mathcal{D}\mathrm{vec}(X)\|^2, \quad (2.12)$$

is quadratic in the Euclidean space $\mathbb{R}^{3\times3}$. However, the cost function $f=g|_{SO_3}$, which is the restriction of g to manifold SO_3 is no longer a quadratic function. Also, we do not linearize this cost by using earlier estimates of R and t in the optimization process as in [55].

Shifted Centroid It turns out to be useful to refer all measurements to the centroids. Thus, denoting $\bar{m} := \frac{1}{n}\sum_{i=1}^{n} m_i$, (2.3) can be rewritten as follows,

$$(I - U_i)(R(m_i - \bar{m}) + (t + R\bar{m})) = 0, \ i = 1, \cdots, n.$$

The shifted model $\{m_i - \bar{m}\}$ has created a new object frame at its centroid and a new translation vector $(t + R\bar{m})$. Once the translation vector $(t + R\bar{m})$ is estimated from the algorithm, it needs to be transformed back into the original object frame. Without loss of generality, we assume subsequently that $\bar{m} = 0$, and the desired translation parameter is t.

2.2 Optimization on the Special Orthogonal Group

Having mathematically formulated the problem as minimizing a smooth function over the intersection of the manifold SO_3 and a cone $\mathcal{K}$, we turn attention to developing numerical algorithm that exploits the geometric structure of this constraint manifold.

2.2.1 Geometry of the Special Orthogonal Group

Rotational motion in $\mathbb{R}^3$ can be represented by the special orthogonal group SO_3, which consists of 3×3 orthogonal matrices with determinant $+1$. It is a Lie group and its associated Lie algebra $\mathfrak{so}_3$ is the set of 3×3 skew symmetric matrices. There is a well known isomorphism from the Lie algebra $(\mathbb{R}^3, \times)$ to the Lie algebra $(\mathfrak{so}_3, [.,.])$, where $\times$ denotes the cross product and $[.,.]$ denotes the matrix commutator. This allows one to identify $\mathfrak{so}_3$ with $\mathbb{R}^3$ using the following mapping,

$$\Omega : \mathbb{R}^3 \to \mathfrak{so}_3, \quad \omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \mapsto \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}. \tag{2.13}$$

Notice that Ω can be written as,

$$\Omega(\omega) = Q_x\omega_x + Q_y\omega_y + Q_z\omega_z, \tag{2.14}$$

where

$$Q_x := \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}, \quad Q_y := \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}, \quad Q_z := \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{2.15}$$

Tangent Space of SO_3

Consider the tangent space of SO_3 at point R,

$$\mathrm{T}_R SO_3 = \{R\Omega \mid \Omega \in \mathfrak{so}_3\}, \tag{2.16}$$

and the affine tangent space of SO_3 at the point R is given as,

$$\mathrm{T}_R^{\mathrm{aff}} SO_3 = \{R + R\Omega \mid \Omega \in \mathfrak{so}_3\}. \tag{2.17}$$

Local Parameterization of SO_3

Recall that a manifold is as a collection of local coordinate charts. Computations on a manifold are often conveniently carried out in these local parameter spaces. Let $R \in SO_3$, there exist a smooth exponential map

$$\mu_R : \mathbb{R}^3 \to SO_3, \;\; \omega \mapsto Re^{\Omega(\omega)}, \tag{2.18}$$

which is a local diffeomorphism around the origin in $\mathbb{R}^3$.

2.2.2 Cost Function

Cost Function on the Manifold SO_3

Recall $\mathcal{D}$ in (2.11) and the smooth function from (2.10),

$$f : SO_3 \to \mathbb{R}, \quad f(R) = \frac{1}{2}\|\mathcal{D}\mathrm{vec}(R)\|^2.$$

In the noise free case, the value of this function is zero if and only if there is a rotation matrix which aligns all object points with the line-of-sight of the corresponding image points exactly. In the presence of noise, the value of the cost function is no longer zero. Thus, we are seeking for the minima of this cost function, which can be interpreted as the least squares approximation to the true rotation.

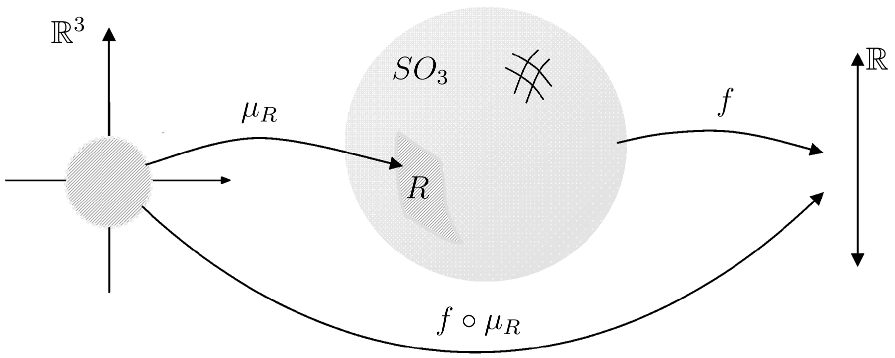


Figure 2.4: The mapping μ_R is a local parameterization of SO_3 around the point R such that $R = \mu_R(0)$, f is the smooth function defined on SO_3 and $f \circ \mu_R$ is the function f expressed in local parameter space $\mathbb{R}^3$.

Local Cost Function

Consider the mappings as in Fig. 2.4. The cost function f at $R \in SO_3$ expressed in local parameter space using the smooth local parameterization μ_R defined in (2.18) is given by,

$$f \circ \mu_R : \mathbb{R}^3 \to \mathbb{R}, \quad f \circ \mu_R(\omega) = \frac{1}{2}\|\mathcal{D}\mathrm{vec}(Re^{\Omega(\omega)})\|^2. \tag{2.19}$$

Quadratic Model of Local Cost Function

The second order Taylor approximation of $f \circ \mu_R$ about $0 \in \mathbb{R}^3$ in direction ω is

$$j_0^{(2)}(f \circ \mu_R) : \mathbb{R}^3 \to \mathbb{R},$$
$$\omega \mapsto \left((f \circ \mu_R)(t\omega) + \frac{\mathrm{d}}{\mathrm{d}\,t}(f \circ \mu_R)(t\omega) + \frac{1}{2}\frac{\mathrm{d}^2}{\mathrm{d}\,t^2}(f \circ \mu_R)(t\omega)\right)\bigg|_{t=0}.$$

This expansion contains three terms:

(i) A constant

$$(f \circ \mu_R)(t\omega)|_{t=0} = \frac{1}{2}\|\mathcal{D}\mathrm{vec}(R)\|^2,$$

(ii) A term linear in ω

$$\frac{\mathrm{d}}{\mathrm{d}\,t}(f\circ\mu_R)(t\omega)\Big|_{t=0} = \mathrm{vec}^\top(R\Omega(\omega))\mathcal{D}^\top\mathcal{D}\mathrm{vec}(R) = \omega^\top\nabla_{f\circ\mu_R}(0),$$

recall Ω from (2.13), and let $\mathrm{vec}(\Omega) := Q\omega$, $Q := [\mathrm{vec}(Q_x)\ \mathrm{vec}(Q_y)\ \mathrm{vec}(Q_z)]$, then we have the Euclidean gradient of the local cost function evaluated at zero,

$$\nabla_{f\circ\mu_R}(0) = Q^\top(I\otimes R^\top)\mathcal{D}^\top\mathcal{D}\mathrm{vec}(R), \tag{2.20}$$

(iii) A quadratic term consists of a sum of two terms. The first term is given as,

$$\mathrm{vec}^\top(R\Omega(\omega))\mathcal{D}^\top\mathcal{D}\mathrm{vec}(R\Omega(\omega)) = \omega^\top\widehat{\mathrm{H}}_{f\circ\mu_R}(0)\omega. \tag{2.21}$$

Denoting $\mathrm{vec}(C) := \mathcal{D}^\top\mathcal{D}\mathrm{vec}(R)$, the second term is,

$$\begin{aligned}\mathrm{vec}^\top(R)\mathcal{D}^\top\mathcal{D}\mathrm{vec}(R\Omega^2(\omega)) &= \mathrm{vec}^\top(R)\mathcal{D}^\top\mathcal{D}(\Omega^\top(\omega)\otimes R)\mathrm{vec}(\Omega(\omega)),\\ &= \mathrm{vec}^\top(R^\top C\Omega^\top(\omega))\mathrm{vec}(\Omega(\omega)),\\ &= \omega^\top\widetilde{\mathrm{H}}_{f\circ\mu_R}(0)\omega.\end{aligned}$$

Thus, the Hessian matrix of $f\circ\mu_R$ evaluated at $0\in\mathbb{R}^3$ is,

$$\mathrm{H}_{f\circ\mu_R}(0) = \widehat{\mathrm{H}}_{f\circ\mu_R)}(0) + \widetilde{\mathrm{H}}_{f\circ\mu_R}(0), \tag{2.22}$$

where

$$\begin{aligned}\widehat{\mathrm{H}}_{f\circ\mu_R}(0) &= Q^\top(I\otimes R^\top)\mathcal{D}^\top\mathcal{D}(I\otimes R)Q \quad \geq 0,\\ \widetilde{\mathrm{H}}_{f\circ\mu_R}(0) &= -Q^\top(I\otimes R^\top C),\end{aligned} \tag{2.23}$$

and since $\widetilde{\mathrm{H}}_{f\circ\mu_R}(0)$ is always symmetric, we have

$$\widetilde{\mathrm{H}}_{f\circ\mu_R}(0) = -\frac{1}{2}Q^\top\left((I\otimes R^\top C) + (I\otimes C^\top R)\right)Q. \tag{2.24}$$

Remark 2.2.1 *Recall f from (2.10), the element $R = \mu_R(0)$ is a critical point of f if and only if the following holds,*

$$Q^\top(I\otimes R^\top)\mathcal{D}^\top\mathcal{D}\mathrm{vec}(R) = 0.$$

A positive definite Hessian, $\mathrm{H}_{f\circ\mu_R}(0) > 0$ indicates that R is a local minimum. In the noise free case, $\widetilde{\mathrm{H}}_{f\circ\mu_R}(0) = 0$ and hence $\mathrm{H}_{f\circ\mu_R}(0) = \widehat{\mathrm{H}}_{f\circ\mu_R}(0) \geq 0$. For generic objects we expect that the number of critical points is finite, and that each is isolated.

2.2.3 Newton Decrement

The Newton decrement δ is defined in terms of the gradient $\nabla_{f\circ\mu_R}(0)$ and the Hessian $\mathrm{H}_{f\circ\mu_R}(0)$, as

$$\delta := \sqrt{[\nabla_{f\circ\mu_R}(0)]^\top[\mathrm{H}_{f\circ\mu_R}(0)]^{-1}\nabla_{f\circ\mu_R}(0)}. \tag{2.25}$$

This decrement δ approaches zero as the algorithm converges to a local or global minimum. It features in the work of Nesterov [67] for optimizing convex self-concordant functions in Euclidean space. Recall that self concordant functions are those where the second derivative terms to the power $\frac{3}{2}$ dominate third derivatives. A key result is that there is a *domain of attraction* for the Newton step using a unity step size if $\delta < \frac{3-\sqrt{5}}{2}$, a global constant.

Although the theory of [67] does not apply immediately for optimization on a manifold, yet since manifolds are locally Euclidean, it can be used as a guideline. In [42], the notion of convex self-concordant functions is explored in a manifold setting. Here after some manipulations, it can be shown that on SO_3, in the neighbourhood of a minimum, the cost function f is locally convex and self-concordant. Thus here we conservatively estimate a *domain of attraction* as $\delta \ll \frac{3-\sqrt{5}}{2}$. Here as curvature of the manifold increases, it makes sense to use more conservative values. We use Newton decrement as an indicator on selecting the appropriate direction of geodesic search as,

$$\begin{aligned} \text{Gradient direction} &\ : \ \delta \geq \epsilon_1 \ \text{ (eg. } 10^{-1}\text{)}, \\ \text{Gauss direction} &\ : \ \epsilon_1 > \delta > \epsilon_2 \ \text{ (eg. } 10^{-2}\text{)}, \\ \text{Newton direction} &\ : \ \delta \leq \epsilon_2. \end{aligned} \tag{2.26}$$

It is also used to assist in the decision for using a Newton step size instead of carrying out geodesic search and for algorithm termination, as

$$\begin{aligned} \text{Newton step size}=\|\omega_{\text{Newton}}\| &\ : \ \delta < \epsilon_3 \ \text{ (eg. } 10^{-3}\text{)}, \\ \text{Terminate} &\ : \ \delta < \epsilon_4 \ \text{ (eg. } 10^{-6}\text{)}. \end{aligned} \tag{2.27}$$

2.2.4 Algorithm

The proposed algorithm is iterative in nature. Each iteration consists of three mappings as,

$$s = \pi_3 \circ \pi_2 \circ \pi_1 : (SO_3 \cap \mathcal{K}) \to (SO_3 \cap \mathcal{K}). \tag{2.28}$$

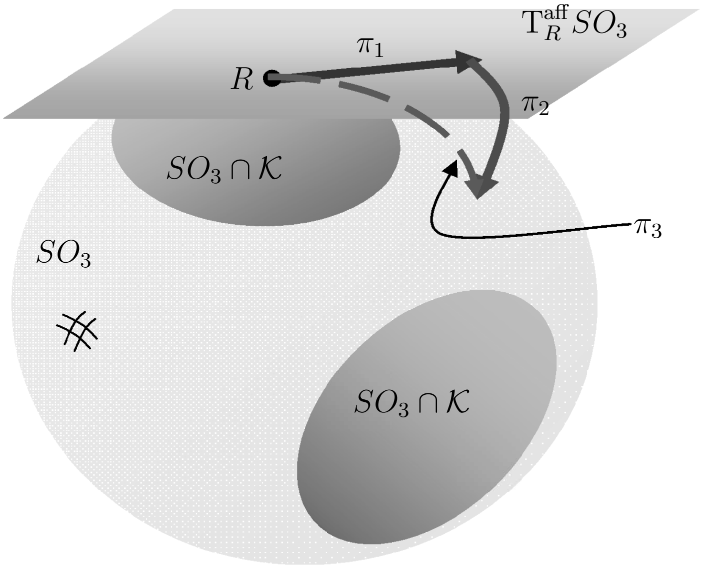


Figure 2.5: Each iteration of the proposed algorithm consists of three mapings, namely π_1 maps a point $R \in (SO_3 \cap \mathcal{K})$ to an element of the affine tangent space $\mathrm{T}_R^{\mathrm{aff}} SO_3$, followed by π_2 which projects that vector back to the manifold and π_3 which carries out geodesic search on SO_3 in the direction of the projected vector.

At each iteration, a local parameterization μ_R of the manifold around $R \in (SO_3 \cap \mathcal{K})$ is constructed. The point R is pulled back to the Euclidean space via μ_R. The optimal vector that minimizes the quadratic model of the local cost function $f \circ \mu_R$, achieved by the operation π_1, is then pushed forward to the manifold via the mapping π_2. Finally, in operation π_3, a one dimensional search along the geodesic on SO_3 in the direction of this projected vector is carried out to ensure cone constraint is satisfied. By appropriately identifying the local parameter space $\mathbb{R}^3$ with the affine tangent space $\mathrm{T}_R^{\mathrm{aff}} SO_3$, the first two steps of the algorithm can also be interpreted geometrically as carrying out an optimization procedures defined on $\mathrm{T}_R^{\mathrm{aff}}SO_3$, followed by a nonlinear projection back to the manifold to give a geodesic search direction, as illustrated in Fig. 2.5.

Optimization in Local Parameter Space

Consider the optimization step,

$$\begin{aligned} \pi_1 : (SO_3 \cap \mathcal{K}) &\rightarrow \big(SO_3 \cap \mathcal{K},\ \mathrm{T}^{\mathrm{aff}}SO_3\big) \\ R &\mapsto (R,\ R + R\Omega(\omega_{\mathrm{opt}}(R)))\,, \end{aligned} \tag{2.29}$$

where ω_{opt} as a function of $R = \mu_R(0)$ is a suitable descent direction of f expressed in local parameter space. Actually, three possibilities for the ω_{opt} calculation are of interest to our algorithm.

1. *Newton direction.* First, we have the Newton direction, which minimizes the quadratic model of the local cost function,

$$\begin{aligned} \omega_{\mathrm{opt}}^{\mathrm{Newton}}(R) &= \arg\min_{y\in\mathbb{R}^3}\ j_0^{(2)}(f \circ \mu_R)(y) \\ &= -[\mathrm{H}_{f\circ\mu_R}(0)]^{-1}\nabla_{f\circ\mu_R}(0). \end{aligned} \tag{2.30}$$

2. *Gauss direction.* Consider the restriction of the function g defined in (2.12) to the affine tangent bundle $\mathrm{T}^{\mathrm{aff}}SO_3$,

$$h : \mathrm{T}^{\mathrm{aff}}SO_3 \rightarrow \mathbb{R}, \quad h(\xi) = g|_{\mathrm{T}^{\mathrm{aff}}SO_3} = \frac{1}{2}\|\mathcal{D}\mathrm{vec}(\xi)\|^2,$$

   and the function h expressed in the local parameter space,

$$\nu_R : \mathbb{R}^3 \rightarrow \mathrm{T}^{\mathrm{aff}}SO_3, \quad y \mapsto R + R\Omega(y),$$

$$h \circ \nu_R : \mathbb{R}^3 \rightarrow \mathbb{R}, \quad h \circ \nu_R(y) = \frac{1}{2}\|\mathcal{D}\mathrm{vec}(R + R\Omega(y)\|^2.$$

   The Gauss direction is the minimizer of the function $h \circ \nu_R$,

$$\begin{aligned} \omega_{\mathrm{opt}}^{\mathrm{Gauss}}(R) &= \arg\min_{y\in\mathbb{R}^3}(h \circ \nu_R)(y), \\ &= -[\mathrm{H}_{h\circ\nu_R}(0)]^{-1}\nabla_{h\circ\nu_R}(0), \\ &= -[\widehat{\mathrm{H}}_{f\circ\mu_R}(0)]^{-1}\nabla_{f\circ\mu_R}(0). \end{aligned} \tag{2.31}$$

3. *Gradient direction.* The third descent direction of interest to us is in the negative gradient direction,
$$\omega_{\text{opt}}^{\text{gradient}}(R) = -\nabla_{f\circ\mu_R}(0). \tag{2.32}$$

Remark 2.2.2 *When* $\mathrm{H}_{f\circ\mu_R}(0)$ *or* $\widehat{\mathrm{H}}_{f\circ\mu_R}(0)$ *is singular, pseudo inverse replaces inverse in the above equations.*

Projecting Back via Local Parameterization

The mapping π_2 projects the optimal affine tangent vector back to the manifold by means of the local parameterization μ_R,

$$\begin{aligned} \pi_2 : \big((SO_3 \cap \mathcal{K}),\ \mathrm{T}^{\text{aff}} SO_3\big) \quad &\rightarrow \quad SO_3 \\ (R,\ R + R\Omega(\omega_{\text{opt}}(R))) \quad &\mapsto \quad Re^{\Omega(\omega_{\text{opt}}(R))}. \end{aligned} \tag{2.33}$$

Analytic Geodesic Search on SO_3

The mapping π_3 involves a one dimensional search along the geodesic curve,

$$\begin{aligned} \pi_3 : SO_3 \quad &\rightarrow \quad SO_3 \cap \mathcal{K}, \\ Re^{\Omega(\omega_{\text{opt}}(R))} \quad &\mapsto \quad Re^{\theta_{\text{opt}}\Omega(\omega_{\text{opt}}(R))}, \end{aligned} \tag{2.34}$$

where θ_{opt} is the optimum step size which minimizes cost function along the geodesic, as well as satisfying the cone constraint.

Certain Newton-type algorithms use a heuristic line search in a particular direction to ensure that the cost function decreases at every step. When the optimization is on a manifold, the line search translates to a geodesic search, and in the case of manifold SO_3, a finite range search. In ill-conditioned problems, as arise in high noise, the number of trial step sizes can be very large for many iterations until the optimization path steers clear of the boundary or saddle point or other sources of ill-conditioning. This is one motivation for us to use an analytic geodesic search, since this is possible for the manifold SO_3. The other motivations are to avoid violating the cone constraint $\mathcal{K}$ and to assist in achieving the global minimum, rather than some other local minimum. The proposed analytic geodesic search is described on an *arbitrary geodesic* on SO_3. It involves the solution of a quartic equation. An important result for us, which does not generalize to matrices higher order than 3×3, is as follows.

Lemma 2.2.1 *Given a vector* $\omega \in \mathbb{R}^3$*, a* 3×3 *skew symmetric matrix* $\Omega \in \mathfrak{so}_3$*, and a step size* $\theta \in [0, 2\pi)$*, with* $\bar{\omega} := \frac{\omega}{\|\omega\|}$*,* $\|\omega\| := \sqrt{(\omega^\top \omega)}$*, then*

$$\mathrm{vec}(e^{\theta\Omega(\bar{\omega})}) = G \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \\ 1 \end{bmatrix}, \; G := \begin{bmatrix} -\mathrm{vec}(\Omega(\bar{\omega})^2) & \mathrm{vec}(\Omega(\bar{\omega}) & \mathrm{vec}(I_3 + \Omega(\bar{\omega})^2) \end{bmatrix}. \tag{2.35}$$

Moreover, the function (2.35) is 2π *periodic in* θ*, so that for any integer* k*,*

$$e^{2\pi k \Omega(\bar{\omega})} = I.$$

Proof 2.2.1 *The proof follows from application of Rodrigues' rotation formula [66],*

$$e^{\theta\Omega(\bar{\omega})} = I + \Omega(\bar{\omega}) \sin(\theta) + \Omega^2(\bar{\omega})(1 - \cos(\theta)).$$

To apply the results of Lemma 2.2.1, consider the cost function f restricted to the geodesic $Re^{\theta\Omega(\bar{\omega}_{\mathrm{opt}}(R))}$ given as,

$$\begin{aligned} \varphi(\theta) &= f|_{Re^{\theta\Omega(\bar{\omega}_{\mathrm{opt}}(R))}} = \frac{1}{2}\|\mathcal{D}\,\mathrm{vec}(Re^{\theta\Omega(\bar{\omega}_{\mathrm{opt}}(R))})\|^2, \\ &= \frac{1}{2}\|\mathcal{D}(I \otimes R) G \begin{bmatrix} \cos(\theta) & \sin(\theta) & 1 \end{bmatrix}^\top \|^2. \end{aligned} \tag{2.36}$$

Now, the task is to 'walk' along the geodesic on the manifold SO_3 and search for a step size θ that minimizes the cost function $\varphi(\theta)$ and satisfies the cone constraint. To achieve this, we find all critical step sizes $\{\theta_*\}$ by setting the first derivative of the cost function (2.36) to zero. Denoting $G^\top(I \otimes R^\top)\mathcal{D}^\top\mathcal{D}(I \otimes R)G$ as (a_{ij}),

$$\frac{\mathrm{d}}{\mathrm{d}\,\theta}\varphi(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 1 \end{bmatrix} (a_{ij}) \begin{bmatrix} \sin(\theta) \\ \cos(\theta) \\ 0 \end{bmatrix} = 0,$$

by eliminating $\sin(\theta)$ from the above equation using trigonometric formula $\cos^2(\theta) = 1 - \sin^2(\theta)$, we obtain a quartic equation,

$$\sum_{j=0}^{4} b_j(\cos(\theta))^j = 0, \tag{2.37}$$

$$\begin{aligned}
b_0 &:= {a_{12}}^2 - {a_{13}}^2, \quad b_1 := -2\left(a_{12}a_{23} + a_{13}(a_{11} - a_{22})\right), \\
b_2 &:= {a_{23}}^2 + {a_{13}}^2 - (a_{11} - a_{22})^2 - 4{a_{12}}^2, \\
b_3 &:= 2\left(2a_{12}a_{23} + a_{13}(a_{11} - a_{22})\right), \quad b_4 := 4{a_{12}}^2 + (a_{11} - a_{22})^2.
\end{aligned}$$

Next, the critical step size that gives minimum cost and fulfills the cone constraint $\mathcal{K}$, denoted the optimum step size θ_{opt} is given as,

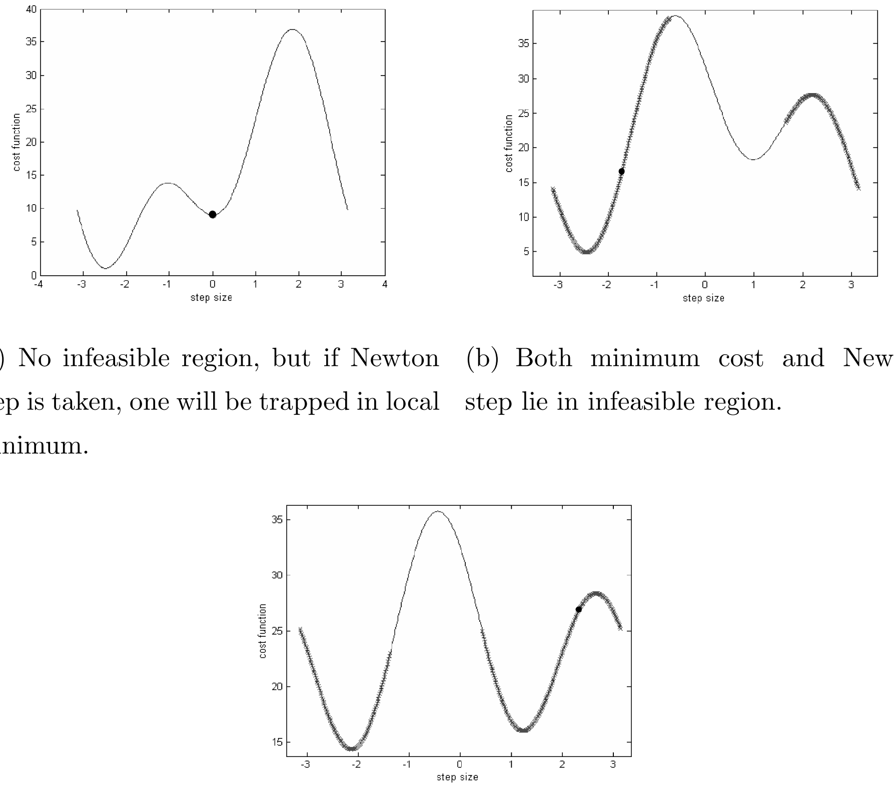
$$\theta_{\text{opt}} = \arg\min_{\theta} \ \varphi(\theta), \quad \text{where } \theta \in \{\theta_* \mid Re^{\theta_*\Omega(\bar{\omega}_{\text{opt}}(R))} \in (SO_3 \cap \mathcal{K})\}. \tag{2.38}$$

Figure 2.6 shows the plots of cost function $\varphi(\theta)$ vs. step size θ in the range $[-\pi, \pi]$. It illustrates the idea of an analytic geodesic search. The black dot in each plot indicates a Newton step, $\theta = \|\omega_{\text{opt}}\|$ in the search direction. The darkened portion of the curves represents the infeasible region where the cone constraint fails. In Fig. 2.6(a), there is no infeasible region. If a Newton step is taken, one will be trapped in the local minimum. However, the analytic geodesic search will select the θ corresponding to the minimum cost in the search direction and hence escapes from the local minimum and heads towards the global minimum. In Fig. 2.6(b), the minimum cost lies in the infeasible region, so the geodesic search will select the θ corresponding to the second minimum cost value that is in the feasible region. Also, the search directions might not yield any feasible local minimum, as depicted in Fig. 2.6(c). In this case, no parameter update is made. Nevertheless, by carrying out a search in a random direction periodically, the probability of achieving a feasible local minimum, or indeed a global minimum, is increased.

2.2.5 Convergence Analysis of Algorithm

Global Convergence

The algorithm is such that the non-negatively valued cost function f decreases monotonically at each iteration, and thus converges. Consequently, point iterate R_k con-



(a) No infeasible region, but if Newton step is taken, one will be trapped in local minimum.

(b) Both minimum cost and Newton step lie in infeasible region.

(c) All minima lie in infeasible region.

Figure 2.6: Analytic geodesic search: plots of cost function f restricted to the geodesic on SO_3 for various step size in the range $[-\pi, \pi]$, the black dot indicates Newton step, darkened portion of the curves represents the infeasible region, i.e., object behind camera.

verges to $\{R \in SO_3 \,|\, f(R) = c\}$ for non-negative scalar c. However, since f is smooth on SO_3, the algorithm step given by the gradient or Gauss or Newton direction is downhill on f and zero when the gradient is zero. Thus R_k converges to a critical point of f, when its gradient is zero.

Of course, critical points of f where the gradient is zero are fixed points of the algorithm. However, critical points other than local minima are unstable fixed points. That is, small random perturbations of the algorithm from these points will result in further iterations which give cost reductions.

The algorithm is designed to escape local minima that are not the global minima by virtue of the geodesic searches including periodic geodesic searches in a random direction. Even so for practical purposes, since the algorithm is only implemented for say 5–10 iterations, there may well be a low probability that there is ‘convergence’ to a local minima which is not a global minimum. Given this background, we provide only a rigorous quadratic convergence analysis in the vicinity of a global minimum, and for simplicity assume that the global minimum is unique and isolated, as we expect in the case of generic objects located in front of the camera.

Local Quadratic Convergence at Global Minimum

Theorem 2.2.1 *Let $R_* \in SO_3$ be the unique and nondegenerate global minimum of the smooth function $f : SO_3 \to \mathbb{R}$ defined in* (2.10)*. Let R_k be a point in an open neighbourhood of R_*. Consider the proposed iteration on SO_3,*

$$R_{k+1} = s(R_k), \quad s = \pi_3 \circ \pi_2 \circ \pi_1, \tag{2.39}$$

where π_1 is given by the Newton direction defined in (2.30)*, π_2 involves projection back to SO_3 via the smooth exponential map of* (2.33)*, and π_3 is an analytic geodesic search described in* (2.34)*. Then the point sequence generated by s converges quadratically to R_*.*

Proof 2.2.2 *See Appendix A.1.*

2.3 Algorithm Initialization

Noise Free Solution

In the noise-free case, the optimal value of the cost function (2.10) is zero. That is there exists an $R \neq 0$ such that

$$\mathcal{D}^\top \mathcal{D} \operatorname{vec}(R) = 0. \tag{2.40}$$

This clearly indicates that $\mathcal{D}^\top \mathcal{D} \geq 0$ cannot be full rank. Subsequently, the rank of $\mathcal{D}$ is of interest. For the generic case,

$$\operatorname{rank}(\mathcal{D}) \leq \min\{(2n-3), 9\}. \tag{2.41}$$

Derivation of this inequality can be found in Appendix A.2.

Let Y be the set of right singular vectors that span the null space of $\mathcal{D}$, with $Y^\top Y = I_q$ with $q \geq 1$, we express $\mathrm{vec}(R)$ linearly in Y as follows,

$$\mathrm{vec}(R) = Y\alpha, \quad R = \begin{bmatrix} Y_1\alpha & Y_2\alpha & Y_3\alpha \end{bmatrix}, \tag{2.42}$$

where Y is a $9 \times q$ matrix,

$$Y := \begin{bmatrix} Y_1^\top & Y_2^\top & Y_3^\top \end{bmatrix}^\top, \quad \alpha := \begin{bmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_q \end{bmatrix}^\top.$$

Here Y_i is a $3 \times q$ matrix. Based on R from (2.42) and the fact that $R^\top R = I$, then

$$\alpha^\top Y_k^\top Y_l \, \alpha = \begin{cases} 1, & \text{if } k = l, \\ 0, & \text{otherwise.} \end{cases} \tag{2.43}$$

Hence, we can solve for α from (2.43). A lower bound for q, the dimension of α, is calculated using (2.41) as

$$q := (9 - \mathrm{rank}(\mathcal{D}) \geq \max(1, 9 - (2n - 3)), \tag{2.44}$$

with equality holding in the generic model case. This is consistent with the known result that a unique solution (2.40) in the generic noise-free case for $n = 6$, since then $q = 1$.

In the generic case for $n \geq 6$ then $q = 1$. There are six possible α solutions from (2.43) as follows,

$$\alpha = \pm\frac{1}{\sqrt{\mathcal{Y}_{11}}} = \pm\frac{1}{\sqrt{\mathcal{Y}_{22}}} = \pm\frac{1}{\sqrt{\mathcal{Y}_{33}}}. \tag{2.45}$$

Each solution is tested to achieve a feasible parameter estimate which yields a zero cost. For $n = 5$ in generic case, $q = 2$. In this case, we can obtain the two elements of α by solving a pair of quadratic equations. Based on (2.43), there are 9 pairs of quadratic equations to be solved. To illustrate, consider the following pair,

$$\alpha^\top \mathcal{Y}_{33}\alpha = 1, \tag{2.46}$$

$$\alpha^\top \mathcal{Y}_{12}\alpha = 0. \tag{2.47}$$

We proceed as follow,

Step 1: Let $\alpha = S_\alpha[\alpha_1 \;\; 1]^\top$, S_α is a scaling factor, solve for α_1 from (2.47).

Step 2: Solve for S_α by substituting α_1 into $\alpha^\top \mathcal{Y}_{33}\alpha = 1$ and finally solve for α.

Here we can obtain up to 4 possible solutions for α. Among all possible solutions, half of them are numerically equivalent but associated with opposite signs.

For $n = 4$ and $n = 3$, the details of solutions for R and t (not unique for $n = 3$) can be found in [73] and [23] respectively. The cases $n = 1$ and $n = 2$ yield an infinite set of solutions. Once α is known, we can solve for $R(\alpha)$, z_i', t from (2.42), (2.8) and (2.7) respectively. There may be multiple solutions, including possibly complex solutions for α. Only the real α that results in $z_i' > 0$ and $R(\alpha)$ associated with minimum cost will be accepted.

Low Noise Initialization

We solve for α as in the noise-free case. However, now Y is chosen from the set of right singular vectors associated with the q smallest singular values, which may not be zero. The singular vectors might not result in a special orthogonal matrix R based on (2.42). Hence, a reasonable approach is to look for an optimum $R_{\text{opt}}(\alpha)$ as

$$\begin{aligned} R_{\text{opt}}(\alpha) &= \arg\min_{R \in SO_3} \|\text{vec}(R) - Y\alpha\|, \\ &= \arg\min_{R \in SO_3} \|R - [Y_1\alpha \;\; Y_2\alpha \;\; Y_3\alpha]\|, \\ &= \arg\max_{R \in SO_3} \text{tr}(R^\top G(\alpha)), \end{aligned} \tag{2.48}$$

where $G(\alpha) := [Y_1\alpha \;\; Y_2\alpha \;\; Y_3\alpha]$. By applying an SVD on $G(\alpha)$, we obtain

$$\begin{aligned} G(\alpha) &= U_R S_R V_R^\top, \\ R_{\text{opt}}(\alpha) &:= U_R \begin{bmatrix} I_2 & 0 \\ 0 & \det(U_R V_R^\top) \end{bmatrix} V_R^\top. \end{aligned} \tag{2.49}$$

Observe that $\det(R_{\text{opt}}(\alpha)) = 1$. Subsequently, t can be obtained from (2.7). Only $R_{\text{opt}}(\alpha)$ associated with minimum cost and fulfilling the cone constraint $\mathcal{K}$ will be accepted. If there is no estimates $R_{\text{opt}}(\alpha)$ such that the cone constraint is satisfied or there is no real solution for α at all, then we refer to this situation as a high noise case, the following high noise initialization is adopted. Such an optimization as in (2.48) is the basis of the Orthogonal Iteration (OI) proposed in [55], and indeed the SVD based solutions of (2.49) which ensures that $\det(R_{\text{opt}}) = +1$, could be used to strengthen the robustness of the OI algorithm.

High Noise Initialization

Select any random $R \in SO_3$ which also satisfies the cone constraint $\mathcal{K}$ of (2.9). Alternatively, one can start with an arbitrary initialization on SO_3, perhaps not feasible in that the cone constraint is not satisfied, and achieve feasibility using the geodesic search.

2.4 Implementation of Algorithm

Start with $R = R_0$ using initialization procedures described in Section 2.3.

Step 1: Carry out the optimization step,

- Compute the gradient $\nabla_{f\circ\mu_R}(0)$ and the Hessian matrix, $\mathrm{H}_{f\circ\mu_R}(0)$ via (2.20) and (2.22) respectively.
- Calculate the Newton decrement δ from (2.25). If $\delta < \epsilon_4$, go to Step 5,
- Compute the optimal direction in the local parameter space,

$$\omega_{\text{opt}}(R) = \begin{cases} \omega_{\text{opt}}^{\text{gradient}} = -\nabla_{f\circ\mu_R}(0), \;\; \text{if } \delta \geq \epsilon_1 \\ \\ \omega_{\text{opt}}^{\text{Gauss}} = -[\widehat{\mathrm{H}}_{f\circ\mu_R}(0)]^{\dagger}\nabla_{f\circ\mu_R}(0), \;\; \text{if} \;\; \epsilon_2 < \delta < \epsilon_1 \\ \\ \omega_{\text{opt}}^{\text{Newton}} = -[\mathrm{H}_{(f\circ\mu_R}(0)]^{\dagger}\nabla_{f\circ\mu_R}(0), \;\; \text{if} \;\; \delta < \epsilon_2 \\ \\ \omega_{\text{opt}}^{\text{rand}}, \;\; \text{periodically, or if any of the above direction} \\ \qquad\quad \text{ends up in an infeasible region} \end{cases}$$

Here, $\omega_{\text{opt}}^{\text{rand}} \in \mathbb{R}^3$ is a random vector with elements, in the range $[0, 1]$,

- Form the normalized direction $\bar{\omega}_{\text{opt}}(R) = \frac{\omega_{\text{opt}}(R)}{\|\omega_{\text{opt}}(R)\|}$.

Step 2: Projecting back to the manifold SO_3 via $\pi_2^{\mu_R}$ or π_2^{svd},

Step 3: Carry out a one dimensional search along the geodesic $Re^{\theta\Omega(\bar{\omega}_{\text{opt}}(R))}$,

- Solve the quartic equation in (2.37) to obtain all critical step sizes $\{\theta_*\}$,
- For each $\theta \in \{\theta_*\}$, calculate the corresponding rotation $Re^{\theta\Omega(\bar{\omega}_{\text{opt}}(R))}$, its cost via (2.36) and the depth parameter estimates via (2.8).
- Obtain the critical rotation angle θ_{opt} that gives minimum cost and fulfills the cone constraint $\mathcal{K}$ via (2.38).
- Compute $\widehat{R} = Re^{\theta_{\text{opt}}\Omega(\bar{\omega}_{\text{opt}}(R))}$,

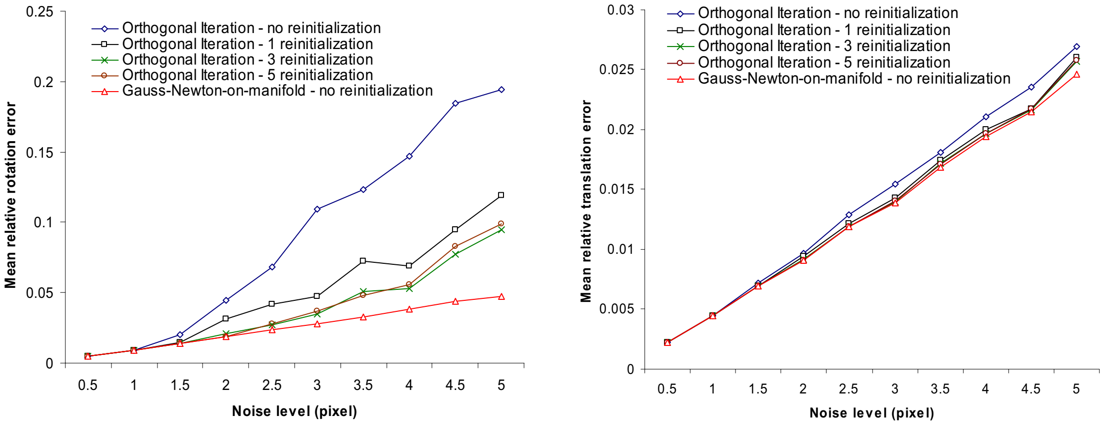
Step 4: Set $R = \widehat{R}$, go back to Step 1,

Step 5: The pose estimates are R and $t = -\mathcal{U}\text{vec}(R)$ respectively.

2.5 Simulations

A set of 3D points that are generated uniformly within a cube of size [-5,5] are chosen as the model. The model is rotated and translated randomly within the field of view of the camera and then projected onto an image plane. Gaussian noise of mean zero and varying standard deviations σ is added to both image plane coordinates independently. The different standard deviations in the Gaussian noise corresponds to various noise levels. All simulations are repeated for 1000 times. A CCD camera with focal length of 600, aspect ratio of 1.0 and principal point of (256,256) is used. All simulations are carried out in MATLAB. Throughout the simulation, we define the relative pose error as the normalized distance between the actual pose and its estimate. That is, $E_A := 2\frac{\|A-\hat{A}\|}{\|A\|+\|\hat{A}\|}$, $A \in \{R, t\}$. The performance of the proposed algorithm, denoted Gauss-Newton-on-manifold is compared against the Orthogonal Iteration (OI) algorithm of [55].

Relative Pose Error vs. Noise level Points and poses are generated as described earlier. With the number of matched points fixed at 12, we vary the noise level between 0.5 to 5.0 pixels. It is clear from Fig. 2.7 that the proposed algorithm (without reinitialization) performs significantly better than the OI algorithm in terms of relative pose error, even after the OI algorithm has been reinitialized 5 times. This is particularly the case for noise level greater than 1.5 pixel.



(a) Relative rotation error vs. noise level.

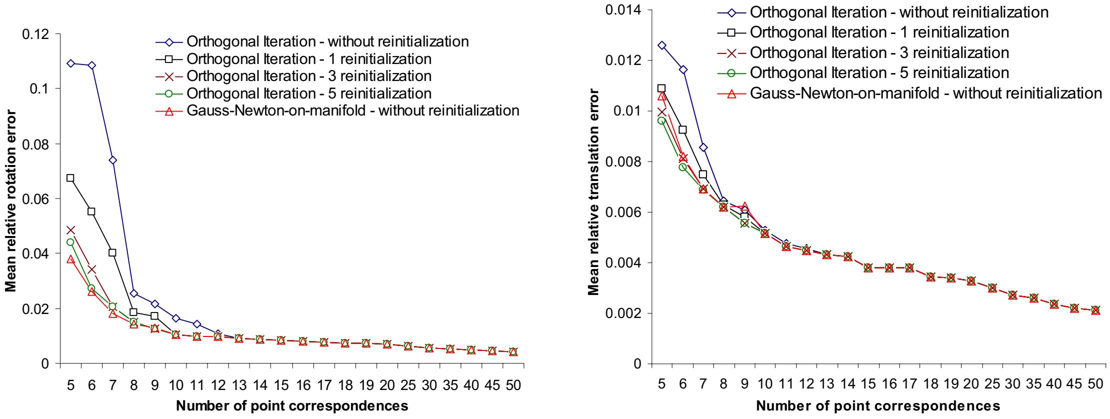
(b) Relative translation error vs. noise level.

Figure 2.7: Relative pose error vs. noise level.

Relative Pose Error vs. Number of Points Here, we vary the number of point correspondences from 5 to 50 and add 1.0 x 1.0 pixel Gaussian noise to the images. Figure 2.8 indicates that the performance of both algorithms improve as the number of point matches increase. However, the proposed recursions (without reinitialization) gives better pose estimates than the OI algorithm, even after 5 reinitialization, for point correspondences less than 15.

Number of Iterations Figure 2.9 shows the average number of iterations (without reinitialization) required by each method to achieve the performance shown in Fig. 2.7 and Fig. 2.8. It is clear that our approach always converges in 5–10 iterations as opposed to the OI algorithm that requires at least 10 iterations.

Rate of Local Convergence Figure 2.10 shows that the proposed algorithm con-



(a) Relative rotation error vs. number of point correspondences.

(b) Relative translation error vs. number of point correspondences.

Figure 2.8: Relative pose error vs. number of point correspondences.

verges to a fixed point R_* at a quadratic rate as opposed to the OI algorithm which converges at a fast linear rate.

Remark 2.5.1 *The cost per iteration of the proposed algorithm is small since implementing the Newton step with* 3×3 *Hessian matrices in not onerous, nor is the solution of a quartic equation for the optimal step size selection. We observe the computational cost per iteration of OI algorithm is much less than ours when implemented using Matlab. But we expect that when implemented in C/C++, the relative computational cost of our algorithm will improve significantly.*

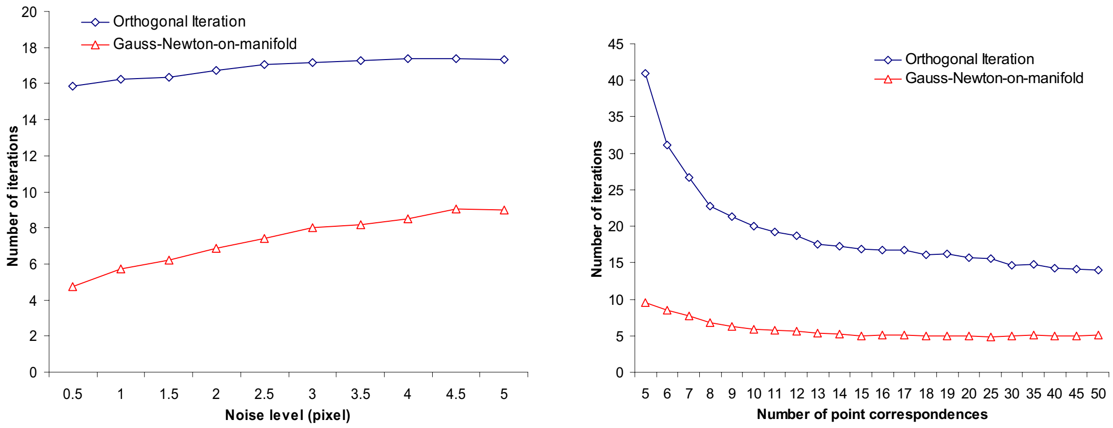
2.6 Algorithm Robust to Outliers

Recall that the proposed algorithm minimizes the sum of squared residuals,

$$\frac{1}{2}\sum_{i=1}^{n} r_i^2, \quad r_i = \|D_i \text{vec}(R)\|,$$

and gives accurate fitting of data when the underlying noise in the data is Gaussian but is very vulnerable to outliers.

To avoid this, a robust version of the algorithm which adapt the M-estimator technique proposed by Huber [40] is presented. The M-estimator technique is based



(a) (b)

Figure 2.9: Number of iterations (mean) vs. noise level and number of corresponding points.

on the idea of reducing the effect of outliers by replacing the squared residuals, r_i^2, with another function of the residuals, yielding the cost function

$$\frac{1}{2}\sum_{i=1}^{n}\rho(r_i), \tag{2.50}$$

where ρ is a symmetric, positive function with a unique minimum at zero and is chosen to be growing slower than the squared function. The solution to this problem can be found by setting its derivative with respect to the unknown parameter vector θ to zero,

$$\sum_{i=1}^{n}\frac{\mathrm{d}\rho}{\mathrm{d}r_i}\frac{\partial r_i}{\partial\theta_j}=0 \tag{2.51}$$

Defining a weight function

$$w_i = r_i\frac{\mathrm{d}\rho}{\mathrm{d}r_i},$$

then (2.51) becomes

$$\sum_{i=1}^{n}w_i r_i\frac{\partial r_i}{\partial\theta_j}=0.$$

Observe that this is the solution to the following problem

$$\min\sum_{i=1}^{n}w_i r_i^2.$$

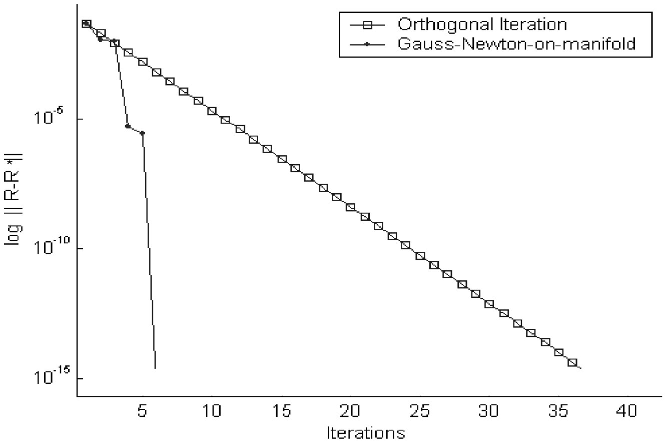


Figure 2.10: Rate of local convergence.

In practice, instead of solving (2.50), it is more efficient to solve a sequence of reweighted least squares problem. Thus, to increase the robustness of the proposed algorithm against outlier, we proceed to minimize the following weighted cost function

$$\begin{aligned} f_w : SO_3 &\to \mathbb{R}, \\ f_w(R) &= \frac{1}{2}\sum_{i=1}^{n} w_i \|D_i \text{vec}(R)\|^2 = \frac{1}{2}\|\mathcal{D}\text{vec}(R)\|_W^2, \end{aligned}$$

where $W = \text{diag}(w_1, \cdots, w_n) \otimes I_3$.

Among the weighting function proposed in the statistics literature, Huber's weight function is given as

$$w_i = \begin{cases} 1 & \text{if } |r_i| \le cs \\ \frac{cs}{|r_i|} & \text{otherwise} \end{cases}, \quad (c = 1.345),$$

and Tukey's biweight function is

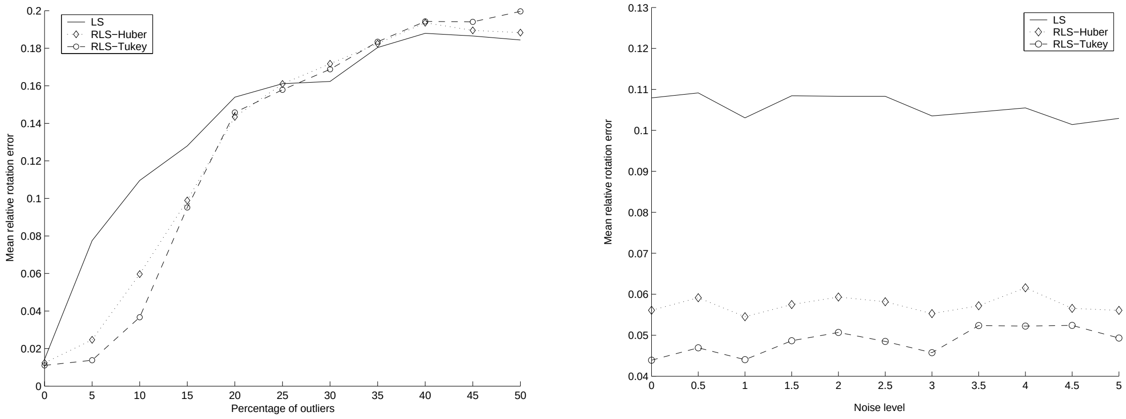
$$w_i = \begin{cases} \left(1 - \left(\frac{r_i}{cs}\right)^2\right)^2 & \text{if } |r_i| \le cs \\ 0 & \text{otherwise} \end{cases}, \quad (c = 4.6851).$$

Here the scale

$$s = \frac{\text{median}_i |r_i|}{0.6745}$$

is introduced to give a scale invariant version of an M-estimator and the value 0.6745 is one half of the interquartile range of the Gaussian normal distribution $N(0, 1)$.

To evaluate the robustness against outliers, the algorithm which minimizes a least squares cost function (LS) is compared against the algorithm which minimizes the reweighted least squares cost function (RLS) using Huber and Tukey weights. Data are generated as in earlier simulation, but now the number of point correspondences is fixed at 20. The outliers are generated by shifting certain image points at least 50 pixels from their actual location. In addition, Gaussian noise of mean zero and σ pixel standard deviation is added to the rest of the point correspondences. 1000 trials were carried out.



(a) Mean relative rotation error vs. percentage of outliers with $\sigma = 1$ pixel.

(b) Mean relative rotation error vs. noise level with 5% outliers.

Figure 2.11(a) shows the plot of mean relative rotation error vs. percentage of outliers with $\sigma = 1$ pixel. In the absence of outliers, both LS and RLS give similar performance. In the presence of less than 20% of outliers, RLS gives significantly better performance than LS but when the percentage of outliers is greater than 25%, we observe no advantage of using RLS against LS. Figure 2.11(b) shows the plot of mean relative rotation error vs. noise level with 5% outliers. This plot indicates that even in the presence of a small relative number of outliers, RLS outperforms LS significantly regardless of the noise level. Also among RLS methods, Tukey's biweight functions seems to give better performance than Huber's weight function.

2.7 Summary

In this chapter, the task of determining position and orientation of a 3D object from single 2D image have been formulated as an optimization problem cast on the intersection of a rotation group and a cone constraint. Newton-like algorithms based on the proposed parameterization-based geometrical framework has been developed. The techniques take into account the underlying geometry of the constraint manifold to achieve a quadratic convergence rate. Unlike most existing numerical procedures, the proposed algorithms evolve on the constraint manifold and thus preserve constraint sets at every iteration. The key differentiating features, each adding measurable value to the algorithm, concern

- A new approach of obtaining noise-free solutions for $n \geq 5$, which can be used for low noise initializations,

- The use of 3×3 Hessian inversion, as opposed to the conventional Newton approach that works with a 6×6 Hessian matrix [53],

- The analytic geodesic search for step size selection, requiring a solution of a quartic, and facilitating escape from a local minimum but not from the global minimum,

- Achieving local quadratic convergence by means of simulations and mathematical proof,

- No need to reinitialize the algorithm to achieve global minimum within the feasible region,

- Introduction of the Newton decrement as an indicator for selection of gradient, Gauss, or Newton directions and for algorithm termination,

- For a prescribed number of iterations, the proposed algorithm achieves significantly lower pose estimation errors than earlier methods and it converges to a global minimum in typically 5–10 iterations.

- To increase robustness against outliers, the proposed algorithm adapting the M-estimator technique is also presented. It is not easy to derive analytical results on this robustness, but simulations appear convincing.

These features suggest that as digital signal processing hardware becomes more powerful, the algorithm can be conveniently applied to on-line pose recovery tasks, and can be a basic tool for more sophisticated tasks involving adaptive object recognition and pose tracking.

Chapter 3

Estimation of Essential Matrices

Chapter 2 demonstrates that the task of estimating the pose of an object from a single intensity image is equivalent to solving an optimization problem cast on a manifold. In this chapter, we focus on determining the relative pose information between a pair of calibrated images, also known as 2D-2D pose estimation. Part of the material in this chapter has been published in [34].

The seminal work of Longuet-Higgin [51] in 1981, showed that the pose information between a pair of calibrated images is algebraically captured by an essential matrix. Many techniques have been developed for estimation of this matrix.

Existing Work Techniques for estimating essential matrices can be categorized into three main groups: linear, iterative and robust methods. Linear methods are mainly based on least-squares and eigenvalue minimization. Iterative methods are based on optimizing linear methods by means of iteration. Robust methods are based on accurate geometry detection and the removal of false matchings. Excellent reviews on the estimation of the essential matrix are presented in [39, 38].

Our Work In this chapter, the task of determining the essential matrix is addressed as minimizing a smooth function over the manifold of all essential matrices, called the essential manifold. Locally quadratically convergent Newton-like techniques exploiting the geometric structure of the constraint manifold were presented. The proposed algorithm is compared with the recent geometric optimization techniques presented by Ma, Sastry and Košecká [58], which are adapted from [20]. The efficiency of the proposed techniques are demonstrated by means of computer simulations, where convergence typically occurs in 5–20 iterations, with complexity per iteration much less than in the work of [58].

Chapter Outline We begin with the problem formulation in Section 3.1. This is followed by presentation of our iterative algorithm in Section 3.2. Algorithm implementation steps are outlined in Section 3.3 and simulation results are presented in Section 3.4. A chapter summary can be found in Section 3.5.

3.1 Problem Formulation

Two images of the same scene are related by epipolar geometry as illustrated in Figure 3.1. The images can be taken by two cameras or the images can be taken by a mobile camera at two different positions. Given an object point M and its two dimensional projections m_1 and m_2 on both image planes with camera centers C_1 and C_2, the five points define an epipolar plane Π, which intersects both image planes I_1 and I_2 at the epipolar lines l_1 and l_2. The image of the camera centers C_1 ,C_2 are captured on the image planes I_2, I_1 at epipoles e_2, e_1, respectively.

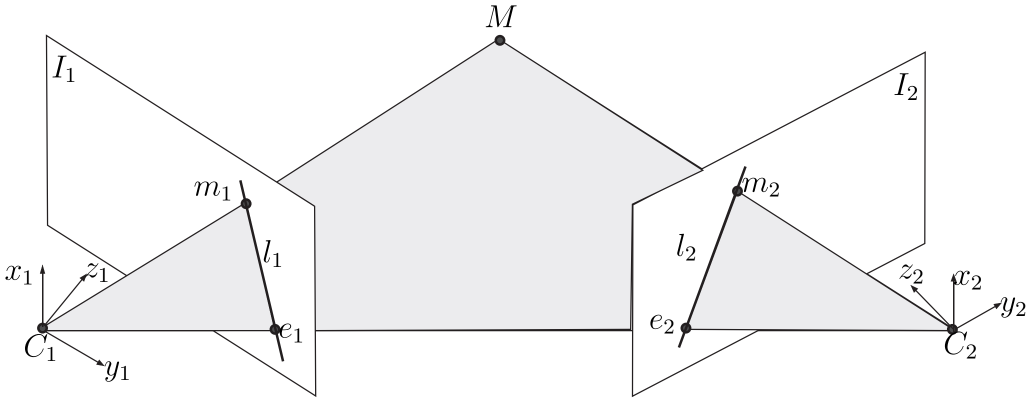


Figure 3.1: The epipolar geometry

If the internal camera parameters are known, the epipolar constraint can be represented algebraically in terms of a 3 × 3 matrix, called the essential matrix E, as

$$m_2^\top E m_1 = 0. \tag{3.1}$$

The essential matrix encapsulates both the rotation and translation parameters, thus it plays a crucial role in motion analysis.

It is known that with 8 or more generic point correspondences in the noise-free case, the essential matrix is uniquely determined. For 5 generic point correspondences, there are up to 20 such solutions [37], with a certain symmetry so that they are all determined from 10 solutions. We are not aware of references for the case of 6 or 7 point correspondences.

Thus, given a set of point correspondences between a pair of calibrated intensity images, the task is to estimate the essential matrix which encapsulates the relative pose between the images.

3.2 Optimization on the Essential Manifold

3.2.1 Geometry of the Essential Manifold

Recall, that an essential matrix is a real (3 × 3)-matrix in factored form

$$E = \Omega R, \tag{3.2}$$

where $R \in SO_3$ is a rotation matrix and $\Omega \in \mathfrak{so}_3$, as defined in (2.13) represents the translation vector between two views. Once E is known, then the factorization (3.2) can be made explicit as shown below. It is well-known, that the essential matrices are characterized by the property that they have exactly one positive singular value of multiplicity two, consequently E must be rank 2 [31]. In particular, normalized essential matrices are those of Frobenius norm equal to $\sqrt{2}$ and which are therefore characterized by having the set of singular values $\{1, 1, 0\}$.

Characterization of Normalized Essential Manifold

The normalized essential manifold is defined as

$$\mathcal{E} := \left\{ \Omega R \;\middle|\; \Omega \in \mathfrak{so}_3, R \in SO_3,\ \|\Omega\|^2 = \operatorname{tr}(\Omega\Omega^\top) = 2 \right\}. \tag{3.3}$$

This is the basic nonlinear constraint set on which the proposed algorithms are defined.

First, we show that a non-zero (3 × 3)-matrix E is essential if and only if there exists a decomposition

$$E = U\Sigma V^\top, \quad \Sigma = \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 0 \end{bmatrix},\ s > 0,\ U, V \in SO_3. \tag{3.4}$$

Note that, E is a normalized essential matrix when $s = 1$. Assuming $E = \Omega R$ with $\Omega \in \mathfrak{so}_3$ and $R \in SO_3$, the equality $E = \Omega R$ implies

$$EE^\top = \Omega R R^\top \Omega^\top = -\Omega^2,$$

with a corresponding set of eigenvalues

$$\lambda(EE^\top) = \left\{ s^2, s^2, 0 \right\}, \text{ where } s := \sqrt{\sum_{i=1}^{3} \omega_i^2}.$$

The set of singular values of E is then $\sigma(E) = \{s, s, 0\}$. For the converse, consider

$$\Psi := \begin{bmatrix} 0 & -s & 0 \\ s & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \in \mathfrak{so}_3 \text{ and } \Gamma := \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \in SO_3.$$

One has

$$\Psi^\top = \Gamma \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Hence for the singular value decomposition of E,

$$\begin{aligned} E &= U \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 0 \end{bmatrix} V^\top = U\Gamma^\top\Gamma \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 0 \end{bmatrix} \Gamma U^\top U\Gamma^\top V^\top, \\ &= \underbrace{(U\Gamma^\top\Psi^\top\Gamma U^\top)}_{\Omega}\underbrace{(U\Gamma^\top V^\top)}_{R} = \Omega R. \end{aligned} \tag{3.5}$$

as required.

The next result characterizes the normalized essential manifold $\mathcal{E}$ as the smooth manifold of (3×3)-matrices with a fixed set of singular values $\{1, 1, 0\}$.

Proposition 3.2.1 *Let us denote*

$$E_0 := \begin{bmatrix} I_2 & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathcal{E} := \left\{ UE_0V^\top \mid U, V \in SO_3 \right\}. \tag{3.6}$$

Then, $\mathcal{E}$ is a smooth five-dimensional manifold diffeomorphic to $\mathbb{RP}^2 \times SO_3$.

Proof 3.2.1 *Recall that the real projective plane $\mathbb{RP}^2$ can be identified with the isospectral manifold*

$$\left\{ U \begin{bmatrix} I_2 & 0 \\ 0 & 0 \end{bmatrix} U^\top \middle| U \in \mathcal{SO}_3 \right\}.$$

Therefore the map

$$\phi : \mathcal{E} \to \mathbb{RP}^2 \times \mathcal{SO}_3, \quad U \begin{bmatrix} I_2 & 0 \\ 0 & 0 \end{bmatrix} V^\top \mapsto \left(U \begin{bmatrix} I_2 & 0 \\ 0 & 0 \end{bmatrix} U^\top, UV^\top \right),$$

is smooth and bijective. Moreover, the inverse

$$\phi^{-1}(X, Y) = XY,$$

is smooth as well. Therefore, ϕ is a diffeomorphism. Since $\mathbb{RP}^2$ is of dimension 2, and $\mathcal{SO}_3$ is of dimension 3, then $\mathbb{RP}^2 \times \mathcal{SO}_3$ is of dimension 5.

The orthogonal matrices appearing in the above SVD of a given essential matrix are not uniquely determined. However, the possible choices are easily described, leading to an explicit description of factorizations.

Remark 3.2.1 *Let* $\mathbb{S}^2 := \left\{\Omega \in \mathfrak{so}_3 \middle| \|\Omega\| = \sqrt{2}\right\}$ *denote the two-sphere in* $\mathfrak{so}_3$ *with radius equal to* $\sqrt{2}$. *The map*

$$\pi : \mathbb{S}^2 \times \mathcal{SO}_3 \to \mathcal{E}, \quad (\Omega, R) \mapsto \Omega R,$$

is a two-to-one covering map with preimages

$$\pi^{-1}\left(U \begin{bmatrix} I_2 & 0 \\ 0 & 0 \end{bmatrix} V^\top\right) = \left(U \begin{bmatrix} 0 & \varepsilon & 0 \\ -\varepsilon & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} U^\top, U \begin{bmatrix} 0 & -\varepsilon & 0 \\ \varepsilon & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} V^\top \right),$$

with $U, V \in \mathcal{SO}_3$ *and* $\varepsilon \in \{+1, -1\}$.

Tangent Space of Essential Manifold

Theorem 3.2.1 *The tangent space at the normalized essential matrix* $E = UE_0V^\top$ *is*

$$\begin{aligned} T_E\mathcal{E} &= \left\{U\,(\Omega E_0 - E_0\Psi)V^\top \middle| \,\Omega, \Psi \in \mathfrak{so}_3\right\}, \\ &= \left\{ U \begin{bmatrix} 0 & \omega_{12} - \psi_{12} & -\psi_{13} \\ -(\omega_{12} - \psi_{12}) & 0 & -\psi_{23} \\ -\omega_{13} & -\omega_{23} & 0 \end{bmatrix} V^\top \middle| \omega_{ij}, \psi_{ij} \in \mathbb{R}, i, j \in \{1,2,3\} \right\}, \end{aligned} \tag{3.7}$$

with usual notation $\Omega = (\omega_{ij})$ *and* $\Psi = (\psi_{ij})$.

Proof 3.2.2 *For any* $E = UE_0V^\top \in \mathcal{E}$. *Let* $\alpha_E : SO_3 \times SO_3 \to \mathcal{E}$ *be the smooth map defined by* $\alpha_E(\widehat{U}, \widehat{V}) = \widehat{U}E\widehat{V}^\top$. *The tangent space* $T_E\mathcal{E}$ *is the image of the linear map*

$$\mathrm{D}\,\alpha_E(I_3, I_3) : \mathfrak{so}_3 \times \mathfrak{so}_3 \to \mathbb{R}^{3\times3}, (\widehat{\Omega}, \widehat{\Psi}) \mapsto \widehat{\Omega}E - E\widehat{\Psi},$$

i.e., the image of the derivative of α_{E_0} *evaluated at the identity* $(I_3, I_3) \in SO_3 \times SO_3$. *By setting* $\Omega := U^\top\widehat{\Omega}U$ *and* $\Psi := V^\top\widehat{\Psi}V$ *the result follows, see [32], pp 89 for details.*

Corollary 3.2.1 *The kernel of the mapping* $\mathrm{D}\,\alpha_{E_0}(I_3, I_3) : \mathfrak{so}_3 \times \mathfrak{so}_3 \to \mathbb{R}^{3\times3}$ *is the set of matrix pairs* $(\Omega, \Psi) \in \mathfrak{so}_3 \times \mathfrak{so}_3$ *with*

$$\Omega = \Psi = \begin{bmatrix} 0 & x & 0 \\ -x & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad x \in \mathbb{R}. \tag{3.8}$$

Proof 3.2.3 *Let* $\Omega = (\omega_{ij}), \Psi = (\psi_{ij})$ *with* $\Omega, \Psi \in \mathfrak{so}_3$ *and* E_0 *as defined in* (3.6)*, then*

$$\begin{aligned}
& \Omega E_0 - E_0 \Psi = 0 \\
\iff & \begin{bmatrix} 0 & \omega_{12} - \psi_{12} & -\psi_{13} \\ -(\omega_{12} - \psi_{12}) & 0 & -\psi_{23} \\ -\omega_{13} & -\omega_{23} & 0 \end{bmatrix} = 0_3 \\
\iff & \omega_{12} = \psi_{12} \text{ and } \omega_{13} = \psi_{13} = \omega_{23} = \psi_{23} = 0.
\end{aligned}$$

Setting $\omega_{12} = \psi_{12} = x$ *gives the result.*

The affine tangent space at $E = UE_0V^\top \in \mathcal{E}$ can be identified with tangent space $T_E\mathcal{E}$ via translating $T_E\mathcal{E}$ by E, as

$$T_E^{\text{aff}}\mathcal{E} = \left\{ U \begin{bmatrix} 1 & -x_3 & -x_5 \\ x_3 & 1 & x_4 \\ -x_2 & x_1 & 0 \end{bmatrix} V^\top \middle| x_1, \ldots, x_5 \in \mathbb{R} \right\}. \tag{3.9}$$

Parameterization of Essential Manifold

Computations on a manifold are often conveniently carried out in terms of a local parameterisation. For our later convergence analysis we therefore need a local parameterisation of the essential manifold, as depicted in Figure 3.2.

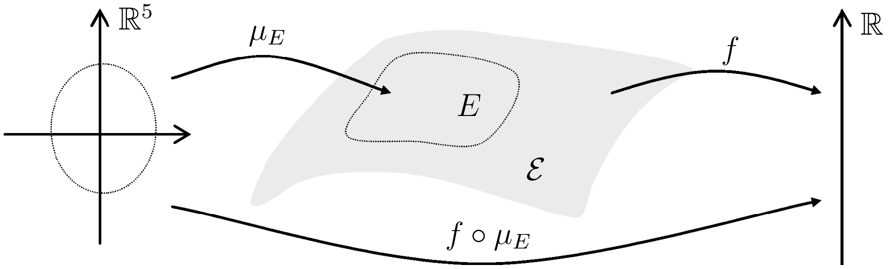


Figure 3.2: The mapping μ_E is the local parameterization of essential manifold $\mathcal{E}$ around point E such that $E = \mu_E(0)$, f is the smooth function defined on $\mathcal{E}$ and $f \circ \mu_E$ is f expressed in local parameter space $\mathbb{R}^5$.

Lemma 3.2.1 *Let $U, V \in SO_3$ be arbitrary, let $x^\top = \begin{bmatrix} x_1, \dots, x_5 \end{bmatrix} \in \mathbb{R}^5$, and let E_0 be defined as in* (3.6)*. Consider the mappings*

$$\Omega_1 : \mathbb{R}^5 \to \mathfrak{so}_3, \begin{bmatrix} x_1, \dots, x_5 \end{bmatrix}^\top \mapsto \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & -\frac{x_3}{\sqrt{2}} & x_2 \\ \frac{x_3}{\sqrt{2}} & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \tag{3.10}$$

and

$$\Omega_2 : \mathbb{R}^5 \to \mathfrak{so}_3, \begin{bmatrix} x_1, \dots, x_5 \end{bmatrix}^\top \mapsto \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & \frac{x_3}{\sqrt{2}} & x_5 \\ -\frac{x_3}{\sqrt{2}} & 0 & -x_4 \\ -x_5 & x_4 & 0 \end{bmatrix}. \tag{3.11}$$

Consider also

$$\mu_E : \mathbb{R}^5 \to \mathcal{E}, \quad x \mapsto U \, \mathrm{e}^{\Omega_1(x)} \, E_0 \, \mathrm{e}^{-\Omega_2(x)} \, V^\top. \tag{3.12}$$

Then the mapping μ_E is a local diffeomorphism around $0 \in \mathbb{R}^5$.

Proof 3.2.4 *Smoothness of μ_E is obvious. We will show that μ_E is an immersion at* 0*. To see that μ_E is an immersion it is sufficient to show that the derivative*

$$\mathrm{D}\mu_E(0) : \mathbb{R}^5 \to T_{\mu_E(0)}\mathcal{E},$$

is injective. For arbitrary $h^\top = \begin{bmatrix} h_1, \dots, h_5 \end{bmatrix} \in \mathbb{R}^5$ we get

$$\mathrm{D}\mu_E(0)h = U \left(\frac{1}{\sqrt{2}} \begin{bmatrix} 0 & -\frac{h_3}{\sqrt{2}} & h_2 \\ \frac{h_3}{\sqrt{2}} & 0 & -h_1 \\ -h_2 & h_1 & 0 \end{bmatrix} E_0 - E_0 \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & \frac{h_3}{\sqrt{2}} & h_5 \\ -\frac{h_3}{\sqrt{2}} & 0 & -h_4 \\ -h_5 & h_4 & 0 \end{bmatrix} \right) V^\top,$$

$$= \frac{1}{\sqrt{2}} U \begin{bmatrix} 0 & -\sqrt{2}h_3 & -h_5 \\ \sqrt{2}h_3 & 0 & h_4 \\ -h_2 & h_1 & 0 \end{bmatrix} V^\top,$$

which implies injectivity in an obvious manner. The result follows..

Remark 3.2.2 *In this chapter we consider the essential manifold as an orbit of the group $SO_3 \times SO_3$ acting on E_0 by equivalence. By the differential of this group action* *the usual canonical Riemannian metric on $SO_3 \times SO_3$ induces a Riemannian metric on the essential manifold which is called the normal Riemannian metric on $\mathcal{E}$, see e.g. [32] for details about this construction. Moreover, by exploiting Corollary 3.2.1 one can show that by this group action geodesics on $SO_3 \times SO_3$, namely one parameter subgroups, are mapped to geodesics on $\mathcal{E}$. We refer to [59], Theorem 5.9.2, for a proof of this fact in a more general context. It turns out that the curves on $\mathcal{E}$ we will use in the sequel, i.e.*

$$\gamma : t \mapsto U \, \mathrm{e}^{t\Omega_1(x)} \, E_0 \, \mathrm{e}^{-t\Omega_2(x)} \, V^\top, \quad x \in \mathbb{R}^5 \tag{3.13}$$

are indeed geodesics on $\mathcal{E}$ with respect to the normal Riemannian metric. In addition, the inverse μ_E^{-1} defines a so-called normal Riemannian coordinate chart. Such a chart has the feature that the Riemannian metric expressed in this chart evaluated at zero is represented by the identity.

3.2.2 Cost Function

Cost Function on the Essential Manifold

Let $M^{(i)} := m_1^{(i)} m_2^{(i)\top}$, where $m_1^{(i)}, m_2^{(i)} \in \mathbb{R}^3$ correspond to the normalized i^{th} point image pair in the left and in the right camera, respectively, for which the correspondence is assumed to be known. Consider the smooth function

$$f : \mathcal{E} \to \mathbb{R}, \quad f(E) = \frac{1}{2} \sum_{i=1}^{n} \left(m_2^{(i)\top} E m_1^{(i)} \right)^2 = \frac{1}{2} \sum_{i=1}^{n} \operatorname{tr}^2(M^{(i)} E). \tag{3.14}$$

The value of this cost function attains zero if and only if there is an essential matrix which fulfills the epipolar constraint for each image point pair. That is, in the noise free case the global minimum value is zero. In the noisy case the zero value will in general not be attained. It nevertheless makes sense to search for mimima of this cost function even in the presence of noise. The minima then can be interpreted as least squares approximations to the true essential matrix, ignoring for the moment any statistical interpretations or refinements.

Local Cost Function

The cost function f at $E \in \mathcal{E}$ expressed in local parameter space using the smooth local parameterization μ_E defined in (3.12) is given by,

$$f \circ \mu_E : \mathbb{R}^5 \to \mathbb{R}, \quad f \circ \mu_E(x) = \frac{1}{2}\sum_{i=1}^{n} \operatorname{tr}^2(M^{(i)} U e^{\Omega_1(x)} E_0 e^{\Omega_2(x)^\top} V^\top). \tag{3.15}$$

Quadratic Model of the Local Cost Function

The second order Taylor expansion of $f \circ \mu_E$ around the point $E = UE_0V^\top \in \mathcal{E}$ is defined as

$$\begin{aligned} j_0^{(2)}(f \circ \mu_E) : \mathbb{R}^5 &\to \mathbb{R}, \\ x &\mapsto \left(f \circ \mu_E(tx) + \frac{\mathrm{d}}{\mathrm{d}t} f \circ \mu_E(tx) + \frac{1}{2}\frac{\mathrm{d}^2}{\mathrm{d}t^2} f \circ \mu_E(tx) \right)\Bigg|_{t=0} \end{aligned} \tag{3.16}$$

As expected the 2-jet contains three terms:

(i) A constant term is,

$$(f \circ \mu_E)(tx)|_{t=0} = \frac{1}{2}\sum_{i=1}^{n} \operatorname{tr}^2 M^{(i)} E, \tag{3.17}$$

(ii) The linear term is,

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\,t} f \circ \mu_E(tx)\Bigg|_{t=0} &= \sum_{i=1}^{n} \left(\operatorname{tr} M^{(i)} E\right) \left(\operatorname{tr} M^{(i)} U \Big(\Omega_1(x) E_0 - E_0 \Omega_2(x)\Big) V^\top\right), \\ &= (\nabla_{f \circ \mu_E}(tx))|_{t=0}^\top \cdot x, \\ &= \left\langle \operatorname{grad} f(UE_0V^\top), U(\Omega_1(x)E_0 - E_0\Omega_2(x))V^\top \right\rangle_{\text{n.RM}}, \end{aligned}$$

which can be interpreted as either (I) the transposed Euclidean gradient of $f \circ \mu_E : \mathbb{R}^5 \to \mathbb{R}$ evaluated at zero acting on $x \in \mathbb{R}^5$, or (II) as the Riemannian gradient of $f : \mathcal{E} \to \mathbb{R}$ evaluated at $E \in \mathcal{E}$ paired by the normal Riemannian metric with the tangent element $U(\Omega_1(x)E_0 - E_0\Omega_2(x))V^\top \in T_E\mathcal{E}$. Let $\operatorname{vec}(\Omega_1(x)) := Q_1 x$, $\operatorname{vec}(\Omega_2(x)) := Q_2 x$, where

$$\begin{aligned} Q_1 &:= \begin{bmatrix} \operatorname{vec}(\tfrac{1}{\sqrt{2}}Q_x) & \operatorname{vec}(\tfrac{1}{\sqrt{2}}Q_y) & \operatorname{vec}(\tfrac{1}{2}Q_z) & 0 & 0 \end{bmatrix} \in \mathbb{R}^{9\times 5}, \\ Q_2 &:= \begin{bmatrix} 0 & 0 & \operatorname{vec}(\tfrac{1}{2}Q_z) & \operatorname{vec}(\tfrac{1}{\sqrt{2}}Q_x) & \operatorname{vec}(\tfrac{1}{2}Q_y) \end{bmatrix} \in \mathbb{R}^{9\times 5}, \end{aligned} \tag{3.18}$$

and Q_x, Q_y, Q_z from (2.15). Then, the explicit formula for the Euclidean gradient of the local cost function evaluated at zero is,

$$\nabla_{f\circ\mu_E}(0) = J^\top \mathcal{M}\operatorname{vec}(E), \tag{3.19}$$

where

$$J := \mathcal{M}(V\otimes U)\begin{bmatrix}(E_0\otimes I) & -(I\otimes E_0)\end{bmatrix}\begin{bmatrix}Q_1\\ Q_2\end{bmatrix}, \quad \mathcal{M} := \begin{bmatrix}\operatorname{vec}^\top(M^{(1)^\top})\\ \vdots \\ \operatorname{vec}^\top(M^{(n)^\top})\end{bmatrix}. \tag{3.20}$$

(iii) A quadratic term in x, which consists of a sum of two terms. The first one,

$$\sum_{i=1}^{n} \operatorname{tr}^2 M^{(i)} U\Big(\Omega_1(x)E_0 - E_0\Omega_2(x)\Big)V^\top = x^\top \cdot \widehat{\mathrm{H}}_{f\circ\mu_E}(0)\cdot x,$$

is a quadratic form on $\mathbb{R}^5$ with the corresponding matrix $\widehat{\mathrm{H}}_{f\circ\mu_E}(0)$ being positive (semi)definite for all $U, V \in \mathcal{SO}_3$. The second term is given as,

$$\begin{aligned}&\sum_{i=1}^{n}\Big(\operatorname{tr} M^{(i)}E\Big)\Big(\operatorname{tr} M^{(i)}U\Big(\Omega_1^2(x)E_0 + E_0\Omega_2^2(x) - 2\Omega_1(x)E_0\Omega_2(x)\Big)V^\top\Big),\\ =\;& x^\top\cdot\widetilde{\mathrm{H}}_{f\circ\mu_E}(0)\cdot x.\end{aligned}$$

Hence, the Hessian matrix of the local cost function evaluated at zero is,

$$\mathrm{H}_{f\circ\mu_E}(0) = \widehat{\mathrm{H}}_{f\circ\mu_E}(0) + \widetilde{\mathrm{H}}_{f\circ\mu_E}(0), \tag{3.21}$$

where

$$\widehat{\mathrm{H}}_{f\circ\mu_E}(0) = J^\top J \;\geq 0, \tag{3.22}$$

and by denoting $\operatorname{vec}(D) := (V^\top\otimes U^\top)\mathcal{M}^\top\mathcal{M}\operatorname{vec}(E)$, we have

$$\widetilde{\mathrm{H}}_{f\circ\mu_E}(0) = \begin{bmatrix}Q_1^\top & Q_2^\top\end{bmatrix}\begin{bmatrix}-(DE_0\otimes I) & (D\otimes E_0)\\ (D^\top\otimes E_0) & -(E_0D\otimes I)\end{bmatrix}\begin{bmatrix}Q_1\\ Q_2\end{bmatrix}. \tag{3.23}$$

For the quadratic term in (3.16) there is then a further interpretation:

$$\frac{\mathrm{d}^2}{\mathrm{d}\,t^2} f\circ\mu_E(tx)\bigg|_{t=0} = \mathcal{H}_{f(\gamma(t))}(\dot\gamma(t),\dot\gamma(t))\big|_{t=0}, \tag{3.24}$$

i.e., $\mathcal{H}_{f(\gamma(t))}$ is the Hessian operator of $f:\mathcal{E}\to\mathbb{R}$ represented along geodesics $\gamma:\mathbb{R}\to\mathcal{E}$, $\gamma(t) = \mu_E(tx)$, $\gamma(0) = E$, see e.g. [46], pp 342.

Remark 3.2.3 *Recall $f : \mathcal{E} \to \mathbb{R}$ defined by* (3.14)*, the element $E \in \mathcal{E}$ is a critical point of f if and only if*

$$J^\top \mathcal{M} \mathrm{vec}(E) = 0.$$

3.2.3 Algorithm

We consider the algorithm as the self map

$$s = \pi_2 \circ \pi_1 : \mathcal{E} \to \mathcal{E} \tag{3.25}$$

consisting of an optimization step followed by a projection. As explained in more detail below our algorithms are iterative in nature. Each algorithmic step consists of two partial steps, the first one being an optimization procedure defined on an appropriate affine tangent space to $\mathcal{E}$ (by identifying $\mathbb{R}^5$ with $T_E^{\mathrm{aff}}\mathcal{E}$ appropriately), and the second one is a nonlinear projection back to the manifold.

Optimization in Local Parameter Space

The mapping π_1 constitutes the optimization step. Depending on the type of projection in the subsequent step, there are two versions of this mapping, namely

- Mapping π_1^{svd}:

$$\begin{aligned} \pi_1^{\mathrm{svd}} : \mathcal{E} &\to \mathrm{T}^{\mathrm{aff}}\mathcal{E} \subset \mathbb{R}^{3\times 3}, \\ E = UE_0V^\top &\mapsto U\big(E_0 + \Omega_1(x_{\mathrm{opt}}(E))E_0 - E_0\Omega_2(x_{\mathrm{opt}}(E))\big)V^\top, \end{aligned} \tag{3.26}$$

- Mapping $\pi_1^{\mu_E}$ or π_1^{cay}:

$$\begin{aligned} \pi_1^{\mu_E} \text{ or } \pi_1^{\mathrm{cay}} : \mathcal{E} &\to (\mathcal{E},\ \mathrm{T}^{\mathrm{aff}}\mathcal{E}), \\ E = UE_0V^\top &\mapsto \big(E,\, U\big(E_0 + \Omega_1(x_{\mathrm{opt}}(E))E_0 - E_0\Omega_2(x_{\mathrm{opt}}(E))\big)V^\top\big)\,, \end{aligned} \tag{3.27}$$

where $x_{\mathrm{opt}} \in \mathbb{R}^5$ as a function of $E = \mu_E(0)$ can be given by the Newton direction when $\mathrm{H}_{f\circ\mu_E}(0) > 0$,

$$x_{\mathrm{opt}}^{\mathrm{Newton}}(E) = \arg\min_{x\in\mathbb{R}^5} j_0^{(2)}(f\circ\mu_E)(x) = -[\mathrm{H}_{f\circ\mu_E}(0)]^{-1}\nabla_{f\circ\mu_E}(0), \tag{3.28}$$

or a Gauss direction otherwise,

$$x_{\text{opt}}^{\text{Gauss}}(E) = -[\widehat{\mathrm{H}}_{f\circ\mu_E}(0)]^{-1}\nabla_{f\circ\mu_E}(0), \tag{3.29}$$

Projecting onto the Essential Manifold

The mapping π_2 involves one of the projections discussed below,

- **Orthogonal Projection.** The orthogonal projection with respect to the Frobenius norm is achieved via singular value decomposition,

  $$\begin{aligned} \pi_2^{\text{svd}} : \mathrm{T}^{\text{aff}}\mathcal{E} \subset \mathbb{R}^{3\times 3} &\rightarrow \mathcal{E}, \\ X = U\Sigma V^\top &\mapsto UE_0V^\top, \end{aligned} \tag{3.30}$$

  where $U, V \in O_3$, $\Sigma = \operatorname{diag}(\sigma_1, \sigma_2, \sigma_3)$ such that the singular values $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq 0$ and σ_3 is simple. Readers who are interested in the mathematical details of this projection can refer to [32].

- **Projecting back via local parameterization.** Let Ω_1 and Ω_2 be defined as in (3.10) and (3.11), respectively. For fixed $x_{\text{opt}} \in \mathbb{R}^5$ consider the smooth mapping

  $$\begin{aligned} \pi_2^{\mu_E} : \big(\mathcal{E}, T^{\text{aff}}\mathcal{E}\big) &\rightarrow \mathcal{E}, \\ \big(E, U\big(E_0 + \Omega_1(x_{\text{opt}}(E))E_0 - E_0\Omega_2(x_{\text{opt}}(E))\big)V^\top\big)& \\ &\mapsto U\,\mathrm{e}^{\Omega_1(x_{\text{opt}}(E))}\,E_0\,\mathrm{e}^{-\Omega_2(x_{\text{opt}}(E))}\,V^\top. \end{aligned} \tag{3.31}$$

  Obviously, for fixed x_{opt} the mapping $\pi_2^{\mu_E}$ maps straight lines in $T_E\mathcal{E}$ going through $E = UE_0V^\top$ such as

  $$l_E : t \mapsto U\big(E_0 + t\Omega_1(x_{\text{opt}})E_0 - E_0t\Omega_2(x_{\text{opt}})\big)V^\top, \tag{3.32}$$

  to smooth curves $\pi_2^{\mu_E}\big(l_{UE_0V^\top}(t)\big) \subset \mathcal{E}$. As mentioned above the resulting curves on $\mathcal{E}$, namely

  $$\pi_2^{\mu_E}\big(l_E(t)\big) = U\,\mathrm{e}^{t\Omega_1(x_{\text{opt}})}\,E_0\,\mathrm{e}^{-t\Omega_2(x_{\text{opt}})}\,V^\top = U\,\mathrm{e}^{\Omega_1(tx_{\text{opt}})}\,E_0\,\mathrm{e}^{-\Omega_2(tx_{\text{opt}})}\,V^\top, \tag{3.33}$$

are geodesics on $\mathcal{E}$ with respect to the so-called normal Riemannian metric on $\mathcal{E}$. One therefore can think of the projection $\pi_2^{\mu_E}$ defined by (3.31) as a Riemannian one. Moreover, the parameterisation μ_E given by (3.12) defines a so-called Riemannian normal coordinate chart μ_E^{-1} sending a suitably chosen open neighborhood of $E \in \mathcal{E}$ diffeomorphically to an open neighborhood of the origin of $\mathbb{R}^5$.

- **Cayley-like projection.** As a further alternative one might approximate the matrix exponential of skew-symmetric matrices by its first order diagonal Padé approximant, or more commonly called the Cayley transformation:

$$\begin{aligned} \operatorname{cay}: \mathfrak{so}_3 &\to SO_3, \\ \Omega &\mapsto \left(I + \frac{1}{2}\Omega\right)\left(I - \frac{1}{2}\Omega\right)^{-1}. \end{aligned} \tag{3.34}$$

The Cayley mapping on $\mathfrak{so}_3$ is well known to be a local diffeomorphism around $0 \in \mathfrak{so}_3$. Moreover, it approximates the exponential mapping $\exp : \mathfrak{so}_3 \to SO_3$ defined by $\Omega \mapsto \exp(\Omega) = \mathrm{e}^{\Omega}$ up to second order. We therefore consider in the sequel the smooth projection mapping

$$\begin{aligned} &\pi_2^{\mathrm{cay}} : \left(\mathcal{E}, T^{\mathrm{aff}}\mathcal{E}\right) \to \mathcal{E}, \qquad (3.35) \\ &\left(E, U\left(E_0 + \Omega_1(x_{\mathrm{opt}}(E))E_0 - E_0\Omega_2(x_{\mathrm{opt}}(E))\right)V^\top\right) \\ &\qquad \mapsto U \operatorname{cay}(\Omega_1(x_{\mathrm{opt}}(E)))E_0 \operatorname{cay}(-\Omega_2(x_{\mathrm{opt}}(E)))V^\top. \end{aligned}$$

In summary, one algorithmic step of s consists of two partial steps, namely π_1 sending a point E on the essential manifold $\mathcal{E}$ to an element of the affine tangent space $T_E^{\mathrm{aff}}\mathcal{E}$, followed by π_2 projecting that element back to $\mathcal{E}$.

3.2.4 Convergence Analysis of Algorithm

Local Quadratic Convergence

Theorem 3.2.2 *Let $E_* \in \mathcal{E}$ be a nondegerate minimum of the smooth function $f : \mathcal{E} \to \mathbb{R}$ defined in* (3.14)*. Let E_k be a point in an open neighbourhood of E_*. Consider*

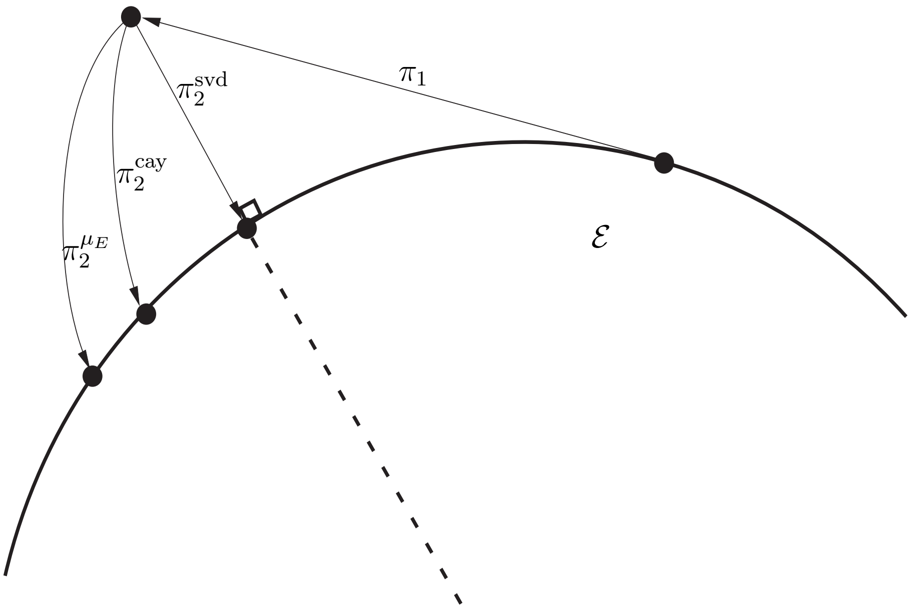


Figure 3.3: Three different projection steps.

the proposed iteration on $\mathcal{E}$,

$$E_{k+1} = s(E_k), \quad s = \pi_2 \circ \pi_1$$

where π_1 is given by the Newton step, defined in (3.28)*, and π_2 involves projection back to the manifold. Then the point sequence generated by s converges quadratically to E_*.*

Proof 3.2.5 *See Appendix A.3.*

3.2.5 Discussions

If $\pi_2^{\mu_E}$ is used for the second algorithmic step π_2 then one can show that the overall algorithm is nothing other than a Riemannian manifold version of Newton's method, the Riemannian metric being the so-called normal one. Despite the well-known fact that under mild assumptions, the Riemannian manifold version of Newton's method is locally quadratically convergent, see [78], Theorem 3.4, p. 57, our results are apparently more than just an application of this nice result. We would like to mention that the latter version of our algorithm is also different from the approach taken in [58]. The Riemannian metric those authors use is different, therefore also their geodesics are not in accordance with ours. Whereas in [58] the local structure of the essential manifold being a product of Stiefel manifolds is exploited we here prefer to think of this manifold as an orbit of $SO_3 \times SO_3$ acting on $\mathbb{R}^{3\times 3}$ by equivalence, i.e., the manifold of all (3×3)-matrices having the set of singular values equal to $\{1, 1, 0\}$. Some features about these different approaches are summarised as follows.

Manifold Structure

- Ma *et al.:* The essential manifold $\mathcal{E}$ is *locally* diffeomorphic to the product of two Stiefel manifolds

$$\mathcal{E} \cong_{\text{local}} \mathbb{S}^2 \times SO_3 \tag{3.36}$$

- Our approach: We exploit the *global* diffeomorphism of $\mathcal{E}$ to the set of matrices having singular values $\{1, 1, 0\}$

$$\mathcal{E} \cong SO_3 \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot SO_3 \tag{3.37}$$

Geodesics Emanating from $E = \Omega R = U E_0 V^\top \in \mathcal{E}$

- Ma *et al.:*

$$t \mapsto \left(\mathrm{e}^{\Delta t}\, \Omega\, \mathrm{e}^{-\Delta t}, R\, \mathrm{e}^{\Gamma t}\right) \tag{3.38}$$

  where $\Delta, \Gamma \in \mathfrak{so}_3$ and $[\Delta, [\Delta, \Omega]] = -\frac{1}{2}\|\Delta\|^2 \Omega$.

- Our approach:

$$t \mapsto U\, \mathrm{e}^{\Delta t}\, E_0\, \mathrm{e}^{-\Gamma t}\, V^\top \tag{3.39}$$

  where $\Delta = \left[\begin{smallmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{smallmatrix}\right]$ and $\Gamma = \left[\begin{smallmatrix} 0 & x_3 & x_5 \\ -x_3 & 0 & -x_4 \\ -x_5 & x_4 & 0 \end{smallmatrix}\right]$ and $x_1, \ldots, x_5 \in \mathbb{R}$.

Riemannian Metric $g : T_E\mathcal{E} \times T_E\mathcal{E} \to \mathbb{R}$

- Ma *et al.:* The Euclidean Riemannian metric induced by the canonical submanifold structure of each factor

  $$\mathbb{S} \subset \mathbb{R}^3 \quad \text{and} \quad SO_3 \subset \mathbb{R}^{3\times 3}, \tag{3.40}$$

  or equivalently, the normal one induced by the similarity group action on the first factor

  $$SO_3 \times \mathbb{S} \to \mathbb{S}, \quad (U, \Omega) \mapsto U\Omega U^\top \tag{3.41}$$

  and right translation on the second factor

  $$SO_3 \times SO_3 \to SO_3, \quad (V, R) \mapsto RV^\top. \tag{3.42}$$

  Explicitly, for two elements of the tangent space $\xi_1, \xi_2 \in T_{(\Omega,R)}\mathcal{E}$ with $\xi_i = ([\Delta_i, \Omega], R\Gamma_i)$

  $$g\Big(\big([\Delta_1, \Omega], R\Gamma_1\big), \big([\Delta_2, \Omega], R\Gamma_2\big)\Big) = \operatorname{tr}\big({\Delta_1}^\top\Delta_2\big) + \operatorname{tr}\big({\Gamma_1}^\top\Gamma_2\big) \tag{3.43}$$

  with $\Delta_i, \Gamma_i \in \mathfrak{so}_3$, $[\Delta_i, [\Delta_i, \Omega]] = -\frac{1}{2}\|\Delta_i\|^2\Omega$ for $i = 1, 2$.

- Our approach: The normal Riemannian metric induced by the equivalence group action

  $$SO_3 \times SO_3 \times \mathbb{R}^{3\times 3} \to \mathbb{R}^{3\times 3}, \quad ((U, V), E) \mapsto UEV^\top. \tag{3.44}$$

  Explicitly, for two elements of the tangent space $\xi_1, \xi_2 \in T_{UE_0V^\top}\mathcal{E}$ with $\xi_i = U(\Delta_iE_0 - E_0\Gamma_i)V^\top$

  $$g\Big(U(\Delta_1E_0 - E_0\Gamma_1)V^\top, U(\Delta_2E_0 - E_0\Gamma_2)V^\top\Big) = \operatorname{tr}\big({\Delta_1}^\top\Delta_2\big) + \operatorname{tr}\big({\Gamma_1}^\top\Gamma_2\big) \tag{3.45}$$

  where for $i = 1, 2$: $\Delta_i = \begin{bmatrix} 0 & -x_3^{(i)} & x_2^{(i)} \\ x_3^{(i)} & 0 & -x_1^{(i)} \\ -x_2^{(i)} & x_1^{(i)} & 0 \end{bmatrix}$ and $\Gamma = \begin{bmatrix} 0 & x_3^{(i)} & x_5^{(i)} \\ -x_3^{(i)} & 0 & -x_4^{(i)} \\ -x_5^{(i)} & x_4^{(i)} & 0 \end{bmatrix}$ and $x_1^{(i)}, \ldots, x_5^{(i)} \in \mathbb{R}$. In fact, the tangent map of μ_E defined by (3.12) maps frames

$\{e_1, \ldots, e_5\}$ in $\mathbb{R}^5$, orthonormal with respect to the Euclidean metric, into frames of $T_E\mathcal{E}$, orthonormal with respect to the normal Riemannian metric:

$$e_i \mapsto \mathrm{D}\,\mu(0) \cdot e_i = U \left(\underbrace{\Omega_1(e_i)E_0 - E_0\Omega_2(e_i)}_{=:\xi_i} \right) V^\top, \tag{3.46}$$

with

$$\left\langle U\xi_i V^\top, U\xi_j V^\top \right\rangle_{\text{n.RM}} = \operatorname{tr}\Omega_1^\top(e_i)\Omega_1(e_j) + \operatorname{tr}\Omega_2^\top(e_i)\Omega_2(e_j) = e_i^\top e_j = \delta_{ij}. \tag{3.47}$$

One might argue that the Riemannian metric we use is induced by a restriction from another Riemannian metric defined on the embedding $\mathbb{R}^{3\times 3}$. This is actually not the case, moreover, one can show that such a metric on $\mathbb{R}^{3\times 3}$ does not exist.

3.3 Implementation of Algorithm

Start with an initial estimate of Essential matrix $E = UE_0V^\top$ obtained from the standard 8-point algorithm.

Step 1. Carry out the optimization step π_1,

- Compute the gradient $\nabla_{f\circ\mu_E}(0)$ and the Hessian matrix $\mathrm{H}_{f\circ\mu_E}(0)$ via (3.19), (3.21) respectively.
- If $\mathrm{H}_{f\circ\mu_E}(0) > 0$, compute the Newton step $x_{\text{opt}} = -[\mathrm{H}_{f\circ\mu_E}(0)]^{-1}\nabla_{f\circ\mu_E}(0)$, otherwise compute the Gauss step $x_{\text{opt}} = -[\widehat{\mathrm{H}}_{f\circ\mu_E}(0)]^{-1}\nabla_{f\circ\mu_E}(0)$.

Step 2. Carry out the projection step π_2. There are three alternative projections,

- π_2^{svd} : Let $x_{\text{opt}} = [x_1\ x_2\ \cdots\ x_5]$, form the optimal affine tangent vector, $\xi_{\text{opt}} \in T_E^{\text{aff}}\mathcal{E}$,

$$\xi_{\text{opt}} = U \begin{bmatrix} 1 & -x_3 & -x_5/\sqrt{2} \\ x_3 & 1 & x_4/\sqrt{2} \\ -x_2/\sqrt{2} & x_1/\sqrt{2} & 0 \end{bmatrix} V^\top = \widehat{U} \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} \widehat{V}^\top,$$

and compute the projected estimate of the essential matrix $\widehat{E} = \widehat{U}E_0\widehat{V}^\top$.

- $\pi_2^{\mu_E} : \widehat{U} = U\,\mathrm{e}^{\Omega_1(x_{\mathrm{opt}})}, \quad \widehat{V} = V\,\mathrm{e}^{\Omega_2(x_{\mathrm{opt}})}, \quad \widehat{E} = \widehat{U} E_0 \widehat{V}^\top.$
- $\pi_2^{\mathrm{cay}} : \ \widehat{U} = U\,\mathrm{cay}(\Omega_1(x_{\mathrm{opt}})), \quad \widehat{V} = V\,\mathrm{cay}(\Omega_2(x_{\mathrm{opt}})), \quad \widehat{E} = \widehat{U} E_0 \widehat{V}^\top.$

Step 3. Set $E = \widehat{E}$, $U = \widehat{U}$, $V = \widehat{V}$, go back to Step 1 if $\|\nabla_{f\circ\mu_E}(0)\| > \varepsilon$, a prescribed accuracy.

3.4 Simulations

All simulations are carried out in Matlab. Each trial

- Generates 3D points randomly in a field of view (FOV) 60° and depth varying from 100 to 400 units of focal length,
- Computes the first projection matrix $P = K[I_3 \ \mathbf{0}]$, where K is the camera calibration matrix,
- Randomly generates a rotation R and a translation $\mathbf{t}$,
- Computes the second projection matrix $P' = K[R\ \mathbf{t}]$,
- Projects the 3D points onto the two 512×512 image planes using P and P'. Only visible points will be used in the algorithm,
- Adds Gaussian noise of mean zero and standard deviation σ to the image points,
- Normalizes the image points with K and solves for essential matrix E,
- Stops when $\|\nabla_{f\circ\mu}(0)\| < 10^{-12}$,
- Computes the mean Euclidean distance between image points and the corresponding epipolar lines on both images.

In the following simulations, ‘alg’ denotes the algorithm which minimizes the cost function (3.14), ‘dist’ denotes the algorithm which minimizes the Euclidean distance between image points and the corresponding epipolar lines,

$$f_D(E) = \sum_{i=1}^{n} w_i^2 \,\mathrm{tr}^2(M^{(i)}E), \tag{3.48}$$

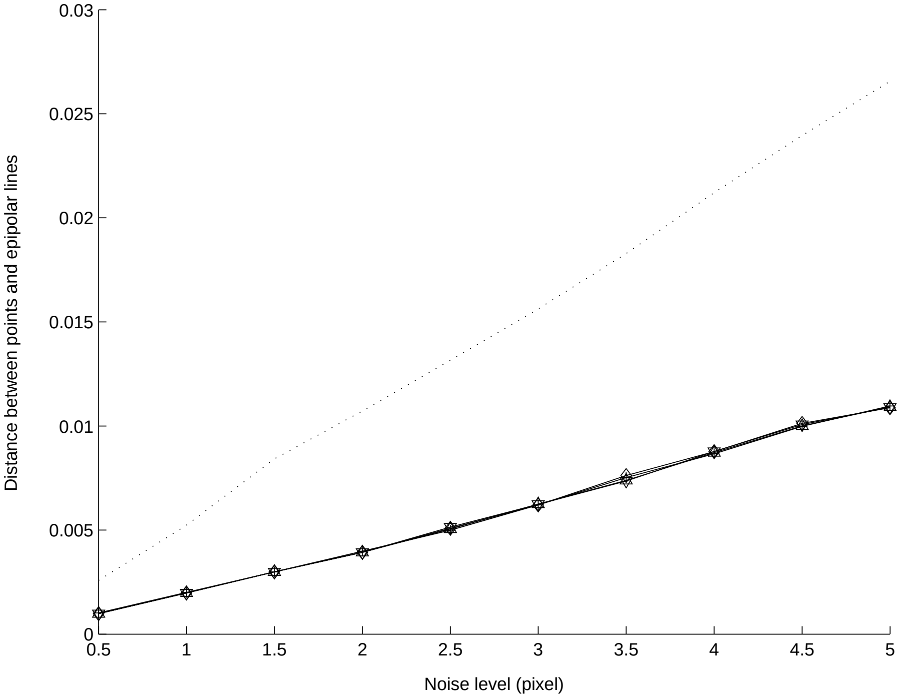


Figure 3.4: Distance between points and epipolar lines vs. noise level for 8p algorithm (dotted line) and other algorithms (PNsvd,PNexp,PNcay,MKS).

where, denoting previous estimate of E as $\widehat{E}$ and e_i as the i^{th} column of the identity matrix,

$$w_i = \left(\frac{1}{\operatorname{tr}^2(m^{(i)} e_1^\top \widehat{E}) + \operatorname{tr}^2(m^{(i)} e_2^\top \widehat{E})} + \frac{1}{\operatorname{tr}^2(e_1 m^{(i)} \widehat{E}) + \operatorname{tr}^2(e_2 m^{(i)} \widehat{E})} \right)^{\frac{1}{2}}, \tag{3.49}$$

and 'grad' denotes the algorithm which minimizes a weighted version of the gradient based nonlinear cost function,

$$f_G(E) = \sum_{i=1}^{n} \frac{\operatorname{tr}^2(M^{(i)} E)}{g_i^2} \tag{3.50}$$

where $g_i = \sqrt{\operatorname{tr}^2(m^{(i)} e_1^\top \widehat{E}) + \operatorname{tr}^2(m^{(i)} e_2^\top \widehat{E}) + \operatorname{tr}^2(e_1 m^{(i)} \widehat{E}) + \operatorname{tr}^2(e_2 m^{(i)} \widehat{E})}$.

Also, '8p' denotes the standard 8-point algorithm, which is a least squares method followed by an orthogonal projection onto the essential manifold via singular value decomposition, 'MKS' stands for the Riemannian-based Newton method published in [58], and 'PNsvd', 'PNexp', 'PNcay' stand for the proposed parameterization-based Newton-like algorithms with π_2^{svd}, $\pi_2^{\mu_E}$, π_2^{cay} projection respectively.

Sensitivity Analysis The number of point correspondences is fixed at 20, and Gaussian noise of varying standard deviation $\sigma \in [0.5, 5]$ is introduced and the simulation is repeated 500 times. In Fig. 3.4, we observe that the standard 8 point method (dotted line) is relatively sensitive to noise compared to other methods in minimizing the cost function (3.14). The figure shows that all three proposed algorithms and MKS have similar sensitivity properties.

Rate of Local Convergence Figure 3.5 illustrates the typical convergence rate of various algorithms. It is clear that all three proposed techniques, together with the MKS method converge at a local quadratic rate. This is parallel with the mathematical proof presented in the appendix and as claimed by [58].

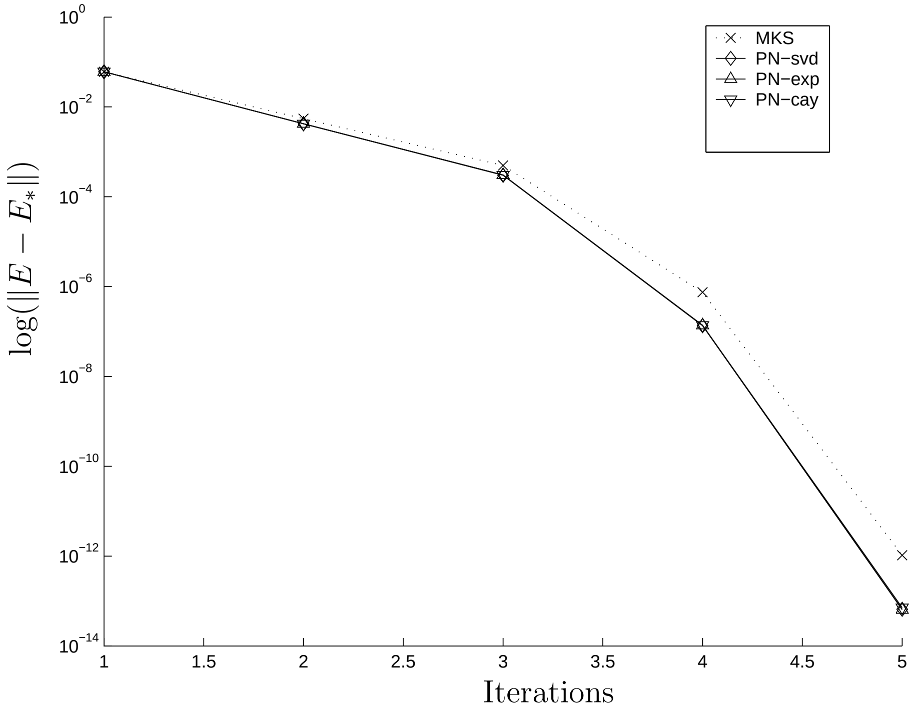


Figure 3.5: Log($\|E - E_*\|$) vs. iterations

Computing Time Figure 3.6 shows the average computing time using different algorithms and minimizing different cost function. For cost function 'alg', the standard 8-point method is the quickest technique. Although all three proposed methods are slightly slower than the standard 8-point method, they are at least 25 times faster than MKS proposed in [58]. Both gradient based and distance based criteria solved using the proposed methods are at least 3 times faster than MKS. Note that although the MKS method is locally quadratically convergent as is ours, its computational effort per iteration is huge, because each iteration is data dependent. The plot also suggests that π_2^{svd} projection is slightly faster than the other two projections.

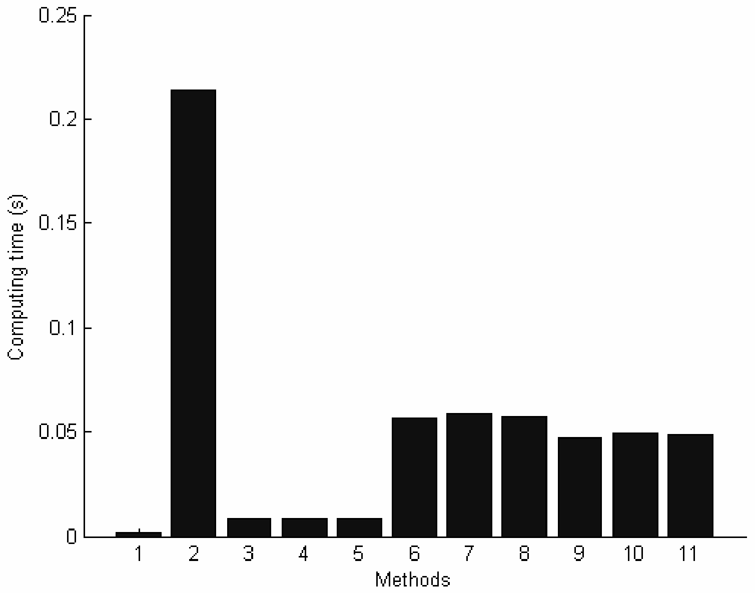


Figure 3.6: Computing time for various methods:(1) alg-8p, (2) alg-MKS, (3) alg-PNsvd, (4) alg-PNexp, (5) alg-PNcay,(6) grad-PNsvd, (7) grad-PNexp, (8) grad-PNcay, (9) dist-PNsvd, (10) dist-PNexp, (11) dist-PNcay

3.5 Summary

This chapter put forward a new geometric optimization framework to compute the essential matrix between two calibrated intensity images, which subsequently allows the pose information to be determined. The approach is based on successive local parameterization of the constraint manifold. The following summarizes research contributions,

- The algorithm requires relatively little computational effort per iteration, and is independent of the number of point correspondences used.

- Our algorithm achieves the same rate of local convergence as Ma *et al.* [58] but is computationally simpler with effort per iteration being significantly lower.

- Our proof of local quadratic convergence is independent of the chosen cost function.

- We have proposed algorithms with three different projections: the exponential map, the Cayley transformation and the orthogonal projection.

Chapter 4

3D Localization of Quadratic Surfaces

In prior chapters, geometric optimization methods have been adopted to recover the pose information from 2D intensity images. Now, we are interested in applying the same framework to locate 3D quadratic surfaces using range images. Quadratic surfaces, also known as quadrics, are commonly occurring shapes in man made objects. Accurate and fast localization of quadrics or surfaces consists of quadric patches from measurement data is important in many industrial robotics and machine vision tasks as well as a step towards developing a flexible manufacturing system. Material in this chapter has appeared in [49].

Existing Work Two main approaches to solving this problem in the literatures are the feature based method and the model based method. The feature based method is based on the geometric relation between a set of 3D feature correspondences extracted from the actual surface data and the surface model stored in the database. This approach has been well studied in the literature, [5, 30, 36]. The model based approach, which we adopt in this chapter, minimizes the error between the data measured on an actual surface and the CAD model of that surface, see [9, 22, 29, 65, 6, 10, 28]. As opposed to the feature based method, this approach does not require 3D data preprocessing such as feature extraction and explicit correspondences. The only information required from the database are the equations of the surface.

Our Work In this chapter, the 3D localization of quadratic surfaces is addressed as an unconstrained optimization problem cast on the special Euclidean group. The optimization is based on locally quadratically convergent Newton-type iterations evolving on this constraint manifold. Simulation results suggest the proposed algorithms are relatively robust against additive Gaussian noise and occlusions.

Chapter Outline In Section 4.1, the task of locating quadrics from single range image is formulated as finding an element on the special Euclidean group that minimizes a certain smooth cost function. In Section 4.2, we present the proposed Newton-like iterative algorithms evolving on the constraint manifold. This is followed by algorithm initialization in Section 4.3 and an outline of the algorithm implementation in Section 4.4. Simulation studies are described in Section 4.5 and a summary of the main results and research contributions is presented in Section 4.6.

4.1 Problem Formulation

4.1.1 Quadric

A quadric, also known as quadratic surface, is defined by the zero set of degree 2 polynomials in 3 variables, as $\{\tilde{m} \in \mathbb{R}^3 \mid \tilde{m}^\top Q_{11}\tilde{m}+2\tilde{m}^\top Q_{12}+Q_{22} = 0\}$. Equivalently, using homogeneous coordinates, a quadric is given by

$$m^\top Q m = 0, \quad m := \begin{bmatrix} \tilde{m} \\ 1 \end{bmatrix}, \quad Q := \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{12}^\top & Q_{22} \end{bmatrix}, \tag{4.1}$$

where Q is the symmetric surface coefficient matrix. Without loss of generality, we take $\operatorname{tr}(Q_{11}^\top Q_{11}) = 1$. Examples of quadrics are illustrated in Fig. 4.1.

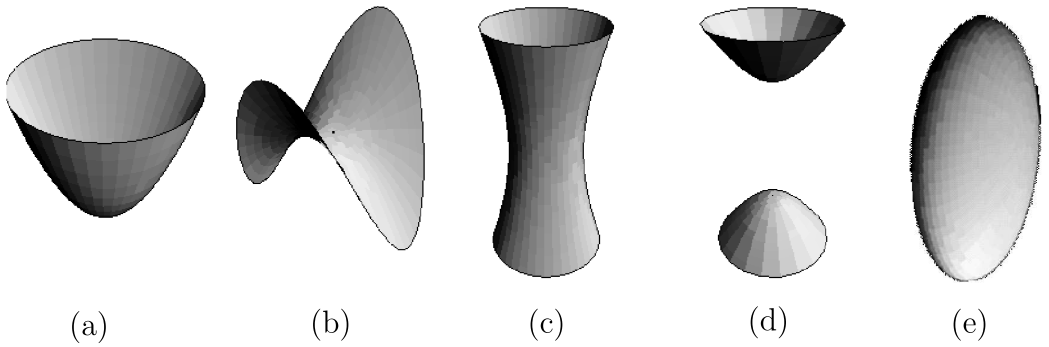


Figure 4.1: Examples of quadratic surface: (a) Elliptic Paraboloid, (b) Hyperbolic Paraboloid, (c) Elliptic Hyperboloid of One Sheet, (d) Elliptic Hyperboloid of Two Sheets, (e) Ellipsoid.

4.1.2 Transformed Quadric

Now, consider the quadric being rotated by a matrix $R \in SO_3$ and translated by a vector $t \in \mathbb{R}^3$. Each point on the transformed quadric $\tilde{p} \in \mathbb{R}^3$ is given by,

$$\left\{ p = \begin{bmatrix} \tilde{p} \\ 1 \end{bmatrix} \in \mathbb{R}^4 \;\middle|\; p^\top A(R,t) p = 0 \right\}. \tag{4.2}$$

Here $A(R,t) := T(R,t)^\top Q T(R,t)$ is the surface coefficient of the transformed quadric, and

$$T(R,t) := \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \in SE_3. \tag{4.3}$$

which represents 3D rigid body transformation is an element of special Euclidean group, SE_3. Thus, given surface measurement data $p_i \in \mathbb{R}^4$ and known surface coefficient Q, the task is to find the transformation matrix $T(R,t) \in SE_3$ satisfying (4.2).

4.2 Optimization on the Special Euclidean Group

To address the task as an unconstrained optimization problem on SE_3, we begin by reviewing some of the relevant geometrical properties of SE_3. We then identify the cost function, propose an iterative algorithm evolving on this manifold and study the convergence analysis of the proposed algorithm.

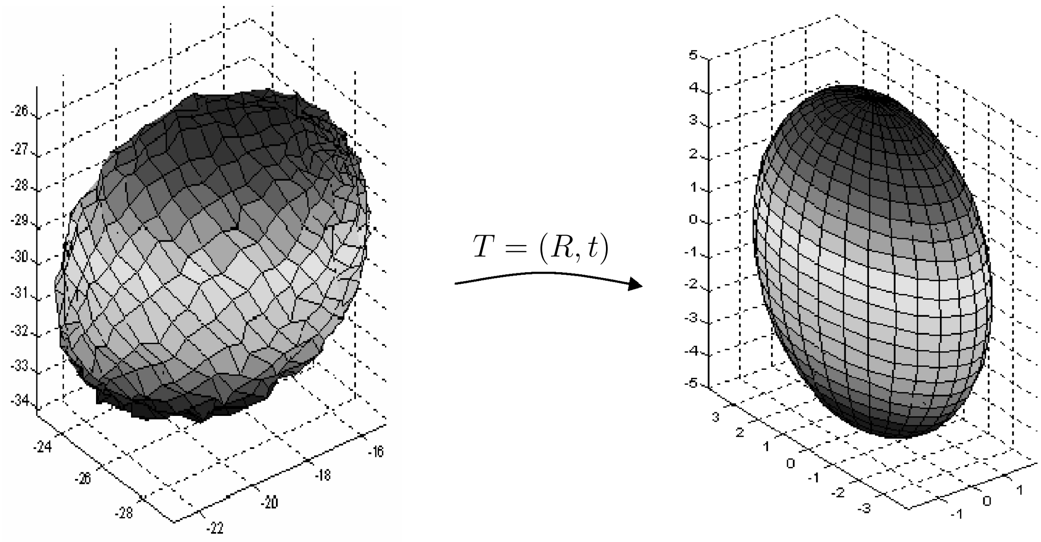


Figure 4.2: Given noisy measurement data of a quadric (ellipsoid) and its CAD model, find the position and orientation of the transformed quadric.

4.2.1 Geometry of the Special Euclidean Group

Rigid body motions in $\mathbb{R}^3$ can be represented by the special Euclidean group, SE_3,

$$SE_3 := \left\{(R,t)\,\middle|\,R \in SO_3,\;\; t \in \mathbb{R}^3\right\} = SO_3 \times \mathbb{R}^3,$$

where SO_3 is the group of 3×3 rotation matrices, as introduced in Section 2.2.1, and t is the translation vector. The special Euclidean group is a Lie group and its associated Lie algebra is denoted $\mathfrak{se}_3$. Due to isomorphism, one can identify $\mathfrak{se}_3$ with $\mathbb{R}^6$ using the following mapping,

$$\zeta : \mathbb{R}^6 \to \mathfrak{se}_3, \quad \zeta(x) = \begin{bmatrix} \Omega(\omega) & v \\ 0 & 0 \end{bmatrix}, \;\; x := \begin{bmatrix} \omega \\ v \end{bmatrix}, \tag{4.4}$$

where $\Omega(\omega)$ as defined in (2.13) and $v \in \mathbb{R}^3$.

Tangent Space of SE_3

The tangent space of SE_3 at T is

$$\mathrm{T}_T SE_3 = \{T\zeta \mid \zeta \in \mathfrak{se}_3\}, \tag{4.5}$$

and the affine tangent space of SE_3 at T is

$$\mathrm{T}_T^{\mathrm{aff}} SE_3 = \{T + T\zeta \mid \zeta \in \mathfrak{se}_3\}. \tag{4.6}$$

Local Parameterization of SE_3

For every point $T \in SE_3$, there exists a smooth exponential map

$$\mu_T : \mathbb{R}^6 \to SE_3, \quad x \mapsto Te^{\zeta(x)}, \tag{4.7}$$

which is a local diffeomorphism around origin in $\mathbb{R}^6$. Such a map will be known as a local parameterization of SE_3 around T.

4.2.2 Cost Function

Cost Function on the Manifold SE_3

We work with the cost function that penalizes the algebraic distance of the measurement data to the quadric,

$$\begin{gathered} f : SE_3 \to \mathbb{R}, \\ f(T) = \frac{1}{n}\sum_{i=1}^{n}(p_i^\top T^\top Q T p_i)^2 = \frac{1}{n}\sum_{i=1}^{n} \mathrm{tr}(p_i p_i^\top T^\top Q T)^2. \end{gathered} \tag{4.8}$$

Exploiting the relationship between the trace and vec operators, (4.8) can be re-expressed as,

$$f(T) = \|B\mathrm{vec}(T^\top Q T)\|^2, \; B := \frac{1}{\sqrt{n}} \begin{bmatrix} \mathrm{vec}^\top(p_1 p_1^\top) \\ \vdots \\ \mathrm{vec}^\top(p_n p_n^\top) \end{bmatrix}. \tag{4.9}$$

Local Cost Function

Consider the mappings as in Fig. 4.3. The cost function f at $T \in SE_3$ expressed in local parameter space using the smooth local parameterization μ_T defined in (4.7) is given by,

$$f \circ \mu_T : \mathbb{R}^6 \to \mathbb{R}, \quad f \circ \mu_T(x) = \|B\mathrm{vec}(e^{\zeta(x)^\top} T^\top Q T e^{\zeta(x)})\|^2. \tag{4.10}$$

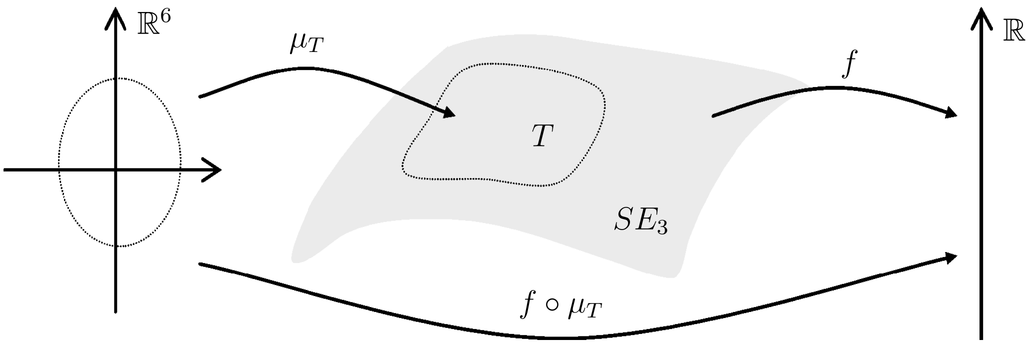


Figure 4.3: The mapping μ_T is the local parameterization of SE_3 around point T such that $\mu_T(0) = T$, f is the smooth function defined on SE_3 and $f \circ \mu_T$ is f expressed in local parameter space $\mathbb{R}^6$.

Quadratic Model of the Local Cost Function

The second order Taylor approximation of $f \circ \mu_T$ about $0 \in \mathbb{R}^6$ in direction x is

$$\begin{aligned} &j_0^{(2)}(f \circ \mu_T) : \mathbb{R}^6 \to \mathbb{R}, \\ x \mapsto &\left((f \circ \mu_T)(tx) + \frac{\mathrm{d}}{\mathrm{d}t}(f \circ \mu_T)(tx) + \frac{1}{2}\frac{\mathrm{d}^2}{\mathrm{d}t^2}(f \circ \mu_T)(tx) \right)\Bigg|_{t=0}. \end{aligned} \tag{4.11}$$

The expansion consists of three terms,

(i) a constant term,

$$(f \circ \mu_T)(tx)|_{t=0} = \|B\mathrm{vec}(A)\|^2,$$

(ii) a linear term,

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}(f \circ \mu_T)(tx)\Bigg|_{t=0} &= 2\mathrm{vec}^\top(A\zeta(x) + \zeta^\top(x)A)B^\top B\mathrm{vec}(A), \\ &= 2x^\top \nabla_{f\circ\mu_T}(0). \end{aligned}$$

Denoting $\mathrm{vec}(\zeta(x)) := Gx$, $\mathrm{vec}(\zeta^\top(x)) := Jx$, then G, J are 16×6 matrices consisting of 1, -1, 0, the explicit formula for the gradient of $f \circ \mu_T$ evaluated at $0 \in \mathbb{R}^6$ is given as,

$$\nabla_{f\circ\mu_T}(0) = C^\top B\mathrm{vec}(A), \quad C := B[(I \otimes A)\ (A \otimes I)] \begin{bmatrix} G \\ J \end{bmatrix}. \tag{4.12}$$

(iii) a quadratic term which itself consists of the sum of two terms. The first term is given as

$$\|B\text{vec}\left(A\zeta(x) + \zeta^\top(x)A\right)\|^2 = x^\top \widehat{\text{H}}_{f\circ\mu_T}(0)x,$$

and the second term is

$$\text{vec}^\top(A)B^\top B\text{vec}\left(A\zeta^2(x) + 2\zeta^\top(x)A\zeta(x) + \zeta^{\top^2}(x)A\right) = x^\top \widetilde{\text{H}}_{f\circ\mu_T}(0)x.$$

Thus, the Hessian of $f \circ \mu_T$ evaluated at $0 \in \mathbb{R}^6$ is

$$\text{H}_{f\circ\mu_T}(0) = \widehat{\text{H}}_{f\circ\mu_T}(0) + \widetilde{\text{H}}_{f\circ\mu_T}(0), \tag{4.13}$$

and denoting $\text{vec}(D) := B^\top B\text{vec}(A)$, we have

$$\begin{aligned} \widehat{\text{H}}_{f\circ\mu_T}(0) &= C^\top C \geq 0, \\ \widetilde{\text{H}}_{f\circ\mu_T}(0) &= \begin{bmatrix} G^\top & J^\top \end{bmatrix} \begin{bmatrix} (D^\top \otimes A) & (D^\top A \otimes I) \\ (AD \otimes I) & (A \otimes D^\top) \end{bmatrix} \begin{bmatrix} G \\ J \end{bmatrix}. \end{aligned} \tag{4.14}$$

4.2.3 Algorithm

The proposed algorithm consists of the iteration,

$$s = \pi_2 \circ \pi_1 : SE_3 \to SE_3, \tag{4.15}$$

where π_1 maps a point $T \in SE_3$ to an element in the affine tangent space (by identifying $\mathbb{R}^6$ with $\text{T}_T^{\text{aff}} SE_3$ appropriately) and π_2 projects that element back to SE_3 by means of the local parametrization μ_T, as illustrated in Fig. 4.4.

Optimization in Local Parameter Space, π_1

The current iteration point $T \in SE_3$ is pulled back to local coordinates $\mathbb{R}^6$ via the local parameterization μ_T. An optimization step is performed in the local parameter space, which involves two partial steps,

$$\pi_1 = \pi_1^b \circ \pi_1^a : SE_3 \to \left(SE_3, \text{T}^{\text{aff}} SE_3\right). \tag{4.16}$$

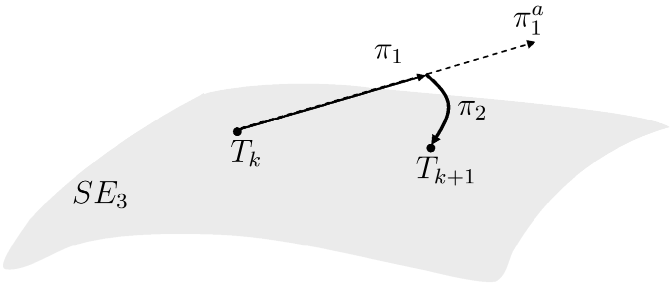


Figure 4.4: The proposed algorithm first maps a point $T \in SE_3$ to an element of the affine tangent space via $\pi_1 = \pi_1^b \circ \pi_1^a$, followed by step π_2 to project that vector back to manifold.

The first partial step π_1^a is used to compute a suitable descent direction for a quadratic model of the local cost function defined in (4.10)

$$\begin{aligned} \pi_1^a : SE_3 &\;\to\; \left(SE_3, \mathrm{T}^{\mathrm{aff}} SE_3\right), \\ T &\;\mapsto\; (T,\ T + \zeta(x_{\mathrm{opt}}(T))), \end{aligned} \tag{4.17}$$

where $x_{\mathrm{opt}} \in \mathbb{R}^6$ as a function of $T = \mu_T(0)$ can be given by the Newton direction when $\mathrm{H}_{f\circ\mu_T}(0) > 0$, as

$$\begin{aligned} x_{\mathrm{opt}}^{\mathrm{Newton}}(T) &= \arg\min_{y \in \mathbb{R}^6} j_0^{(2)}(f \circ \mu_T)(y) \\ &= -[\mathrm{H}_{f\circ\mu_T}(0)]^{-1} \nabla_{f\circ\mu_T}(0), \end{aligned} \tag{4.18}$$

or a Gauss direction otherwise, as

$$x_{\mathrm{opt}}^{\mathrm{Gauss}}(T) = -[\widehat{\mathrm{H}}_{f\circ\mu_T}(0)]^{-1} \nabla_{f\circ\mu_T}(0). \tag{4.19}$$

Once an optimal direction is computed, the second partial step π_1^b is used to carry out an approximate one dimensional line search along this descent direction. We proceed with a search on the scalar $\lambda > 0$ which ensures that the cost function $f \circ \mu_T(\lambda x_{\mathrm{opt}})$ is reduced at every step, giving rise to the mappings,

$$\begin{aligned} \pi_1^b : \left(SE_3, \mathrm{T}^{\mathrm{aff}} SE_3\right) &\;\to\; \left(SE_3, \mathrm{T}^{\mathrm{aff}} SE_3\right), \\ (T,\ T + T\zeta(x_{\mathrm{opt}}(T))) &\;\mapsto\; (T, T + T\zeta(\lambda_{\mathrm{opt}} x_{\mathrm{opt}}(T))), \end{aligned} \tag{4.20}$$

where λ_{opt} is the step length that reduces the cost function in direction x_{opt}, and is found using the simple backtracking line search. Since we are using a descent direction, for sufficiently small step size, the cost function will go downhill.

Remark 4.2.1 *Given a descent direction x_{opt} for function $f \circ \mu_T$ at $0 \in \mathbb{R}^6$,*

- *An exact line search is,*

$$\lambda_{\text{opt}} = \arg\min_{\lambda > 0} f \circ \mu_T(\lambda x_{\text{opt}}), \tag{4.21}$$

- *A backtracking line search with parameters $\alpha \in (0, 0.5), \beta \in (0, 1)$, then starting at $\lambda := 1$, we proceed as follows,*

$$\begin{aligned} \text{While} \quad & f \circ \mu_T(\lambda x_{\text{opt}}) > f \circ \mu_T(0) + \alpha\lambda[\nabla_{f\circ\mu_T}(0)]^\top x_{\text{opt}}, \\ \text{do} \quad & \lambda := \beta\lambda, \end{aligned}$$

and $\lambda_{\text{opt}} := \lambda$.

Projecting Back via Local Parametrization, π_2

Once the descent direction and downhill step size has been obtained, it is projected back to the manifold via the local parametrization μ_T,

$$\begin{aligned} \pi_2 : (SE_3, \mathrm{T}^{\text{aff}} SE_3) & \rightarrow SE_3, \\ T + T\zeta(\lambda_{\text{opt}} x_{\text{opt}}(T)) & \mapsto Te^{\zeta(\lambda_{\text{opt}} x_{\text{opt}}(T))}. \end{aligned} \tag{4.22}$$

4.2.4 Convergence Analysis of the Algorithm

Local Quadratic Convergence

Theorem 4.2.1 *Let $T_* \in SE_3$ be a nondegerate minimum of the smooth function $f : SE_3 \rightarrow \mathbb{R}$ defined in (4.9). Let T_k be a point in an open neighbourhood of T_*. Consider the proposed iteration on SE_3,*

$$T_{k+1} = s(T_k), \quad s = \pi_2 \circ \pi_1$$

where π_1 is given by the Newton step, defined in (4.18), and π_2 involves projection back to the manifold via the smooth local parameterization of (4.22). Then the point sequence generated by s converges quadratically to T_*.

Proof 4.2.1 *See Appendix A.4.*

Global Convergence

The proposed algorithm achieves local quadratic convergence rates. However, the algorithm does not address the issue of escaping local minima. From implementation of the algorithm, convergence to local minima is particularly frequent for elliptic hyperboloid, hyperbolic paraboloid and hyperboloid of two sheets. Simulations suggest that the simplest approach is to initialize the algorithm randomly at different points on the manifold, and select the one with lowest cost.

4.3 Algorithm Initialization

For initialization of the algorithm, we use a two steps least squares approach that gives closed form solution. In this approach, the pose estimation problem is split up into two subproblems, namely the surface fitting problem to recover the surface coefficient followed by pose estimation. Details of this strategies can be found in [6]. We summarize the steps used to estimate the pose in closed form and add in few new results that help in pose computation.

Recovering the Surface Coefficient

Consider the transformed surface coefficient $A := T^\top QT$. This belongs to the class of symmetric matrices $\mathbb{S}^{4\times 4}$. Consider the function

$$\phi : \mathbb{S}^{4\times 4} \to \mathbb{R}, \quad S \mapsto \|B\text{vec}(S)\|^2. \tag{4.23}$$

and the mapping,

$$\nu : \mathbb{R}^{10} \to \mathbb{S}^{4\times 4}, \qquad a = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{10} \end{bmatrix} \mapsto \begin{bmatrix} a_1 & a_2 & a_3 & a_7 \\ a_2 & a_4 & a_5 & a_8 \\ a_3 & a_5 & a_6 & a_9 \\ a_7 & a_8 & a_9 & a_{10} \end{bmatrix}. \tag{4.24}$$

It is clear that this mapping is bijective. Now,

$$\phi \circ \nu : \mathbb{R}^{10} \to \mathbb{R}, \quad a \mapsto \|B\text{vec}(\nu(a))\|^2 = \|BKa\|^2, \tag{4.25}$$

where K is a matrix consisting of elements 1 and 0 such that $Ka = \nu(a)$. The vector a^* that minimizes the function (4.25) subject to constraint $\|a\| = 1$ is the right singular vector corresponding to the zero singular value of the matrix BK. Of course, when the data has noise, then a^* is the right singular vector associated with the minimum singular value of matrix BK. However, to achieve Euclidean invariance, it is usual to constrain solution such that $\operatorname{tr}(A_{11}^\top A_{11}) = \operatorname{tr}(Q_{11}^\top Q_{11}) = 1$, i.e

$$\bar{a}^\top C \bar{a} = 1, \quad \bar{a} = [a_1 \cdots a_6], \quad C := \operatorname{diag}(1, 2, 2, 1, 2, 1). \tag{4.26}$$

Once a^* is obtained, and since the map ν is bijective, the optimum $A \in \mathcal{S}$ is $A^* = \nu(a^*)$.

Recovering Motion Parameters

Once the surface coefficient of the transformed quadric A is determined, we can obtain $R \in SO_3$ and $t \in \mathbb{R}^3$ as follows. We know that

$$\kappa T(R,t)^\top Q T(R,t) = A, \quad \text{for some scalar } \kappa, \tag{4.27}$$

and since Q_{11}, A_{11} are symmetric, singular value decomposition of both matrices will give,

$$Q_{11} = V_Q \Sigma_Q V_Q^\top, \quad A_{11} = V_A \Sigma_A V_A^\top, \tag{4.28}$$

where $V_Q, V_A \in \mathcal{O}_3$ and Σ_Q, Σ_A are diagonal matrices with diagonal elements decreasing in magnitude, then we have

$$\{R^i = V_Q \Gamma^i V_A^\top \in SO_3\}, \tag{4.29}$$

where Γ^i is a diagonal matrix with diagonal elements ± 1. There are $2^3 = 8$ possible Γ^i matrices. We claim here that the optimal rotation is the one associated with minimum cost and minimum distance from the original position, i.e.,

$$i^* = \operatorname{argmin}_i \|R^i - I\|^2 = \operatorname{argmax}_i \ \operatorname{tr}(R^i). \tag{4.30}$$

Once optimal $R^* := R^{i^*}$ is found, we can compute an optimal t^* from,

$$t^* = Q_{11}^{-1}(\kappa^{-1} R A_{12} - Q_{12}), \quad \kappa = \frac{1}{3}\operatorname{tr}(\Sigma_A \Sigma_Q^{-1}). \tag{4.31}$$

Note that when Q_{11} is singular, such as in the case of paraboloid surfaces, the solution is not unique, so a pseudo inverse is used to recover one optimal solution t^*.

4.4 Implementation of Algorithm

Start with $T = T_0 \in SE_3$ using the initialization procedure described in Section 4.3.

Step 1: Carry out the optimization step,

- Compute the gradient vector $\nabla_{f\circ\mu_T}(0)$ and the Hessian matrix $\mathrm{H}_{f\circ\mu_T}(0)$ via (4.12), (4.13) respectively,
- If $\mathrm{H}_{f\circ\mu_T}(0) > 0$,
  compute the Newton step, $x_{\text{opt}}(T) = -[\mathrm{H}_{f\circ\mu_T}(0)]^{-1}\nabla_{f\circ\mu_T}(0)$,
  otherwise compute the Gauss step $x_{\text{opt}}(T) = -[\widehat{\mathrm{H}}_{f\circ\mu_T}(0)]^{-1}\nabla_{f\circ\mu_T}(0)$,
- Compute the optimum step size λ_{opt} in direction $x_{\text{opt}}(T)$ using backtracking line search, as described in *Remark 4.2.1.*

Step 2: Carry out the projection step, $\widehat{T} = Te^{\zeta(\lambda_{\text{opt}}x_{\text{opt}}(T))}$,

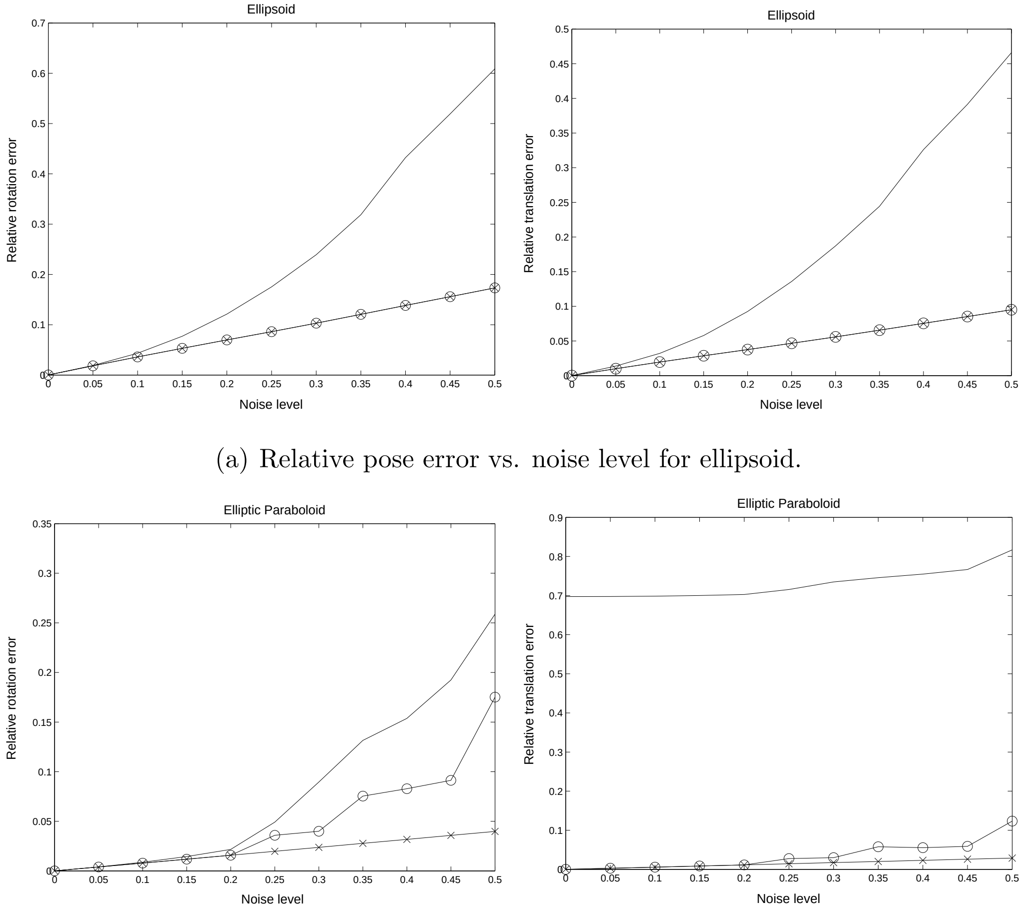
Step 3: Set $T = \widehat{T}$, go back to Step 1 if $\|\nabla_{f\circ\mu_T}(0)\| > \epsilon$, a prescribed accuracy.

4.5 Simulations

A series of simulations were performed on artificially generated uniformly distributed points on quadric surface. Our proposed geometric approach (GA) has been compared with the 2 steps closed form least squares solutions presented under algorithm initialization (LS) and the cyclic coordinate descent method (CCD).

Cyclic coordinate descent is a variant of our proposed approach. It exploits the property that SE_3 is the product manifold $SO_3 \times \mathbb{R}^3$. At each iteration, we first freeze the rotation and optimize only the translation, then next freeze the translation and optimize only the rotation. The motivation for this is that we can carry out analytic geodesic searches in SO_3, requiring the solution of an 8^{th} order polynomial. Likewise, in a line search for $t \in \mathbb{R}^3$ a 3^{rd} order polynomial is solved. The advantage of this approach is that it can potentially escape from a local minimum, not the global minimum in a few iterations, without any random reinitializations. The algorithm is not useful after the first few iterations since it converges only linearly and the analytic line search has not then add value.

Performances of the different techniques are evaluated by comparing the relative Euclidean distance between the pose parameters. All simulations are implemented using Matlab.



(a) Relative pose error vs. noise level for ellipsoid.

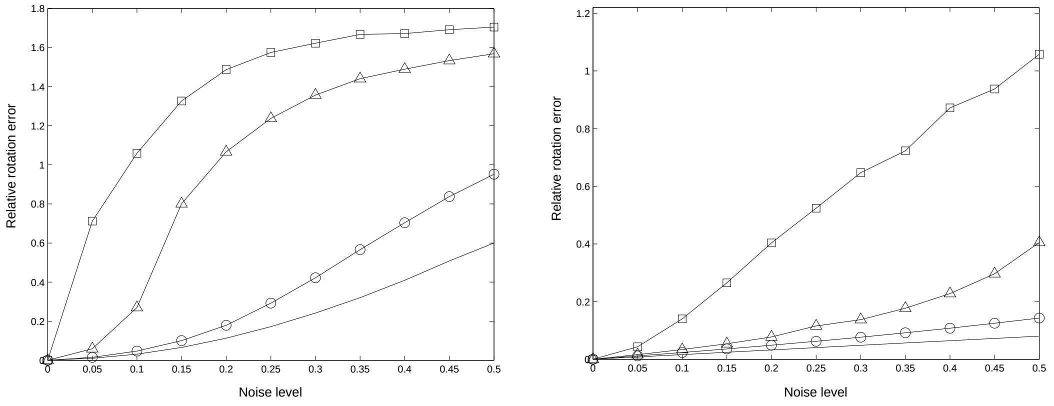
(b) Relative pose error vs. noise level for elliptic paraboloid.

Figure 4.5: Robustness against additive Gaussian noise for closed form least squares approach (solid line), geometric approach without restart (○), geometric approach with 5 random restart (×).

Robustness against Additive Gaussian Noise We investigate the robustness of the algorithms as the measured data are corrupted with increasing amount of additive Gaussian noise and when only partial views of the surface are available.

Simulations show both CCD and GA have the same performance accuracy when the noise level and size of the surface are varied. Thus, only results for GA and LS are plotted. Figure 4.5 indicates that GA is far less sensitive to additive Gaussian noise and than LS. Interestingly, for an ellipsoid, the initialization achieves the global minimum in the presence of high noise level (Fig. 4.5(a)) since random restart converges to the same minimum. Similar results are also observed for an elliptic hyperboloid of one sheet. However, for elliptic paraboloid (Fig. 4.5(b)), hyperboloid of two sheets and hyperbolic paraboloid, we observe the presence of many local minima, thus random reinitializations of the algorithm or initial CCD iterations are required to achieve the 'global' minimum.

Robustness against Occlusion Figure 4.6 shows the performance of GA and LS when part of the surface are occluded. It clearly indicates that GA is far more robust to occlusion than LS. Likewise, for relative translation error.



(a) Least squares approach

(b) Geometric approach

Figure 4.6: Robustness against occlusion: whole surface (solid line), half surface (○), quarter surface (△), small patch (□).

Rate of Local Convergence Figure 4.7 illustrates that GA converges at a local quadratic rate and CCD always converges very quickly at its first few iterations, but then converges linearly. It may make sense to use CCD for the first few iterations then switch to the Newton method. Also, the local quadratic convergence rate GA is also better than the approach presented in [65] which claims to converge at an exponential rate.

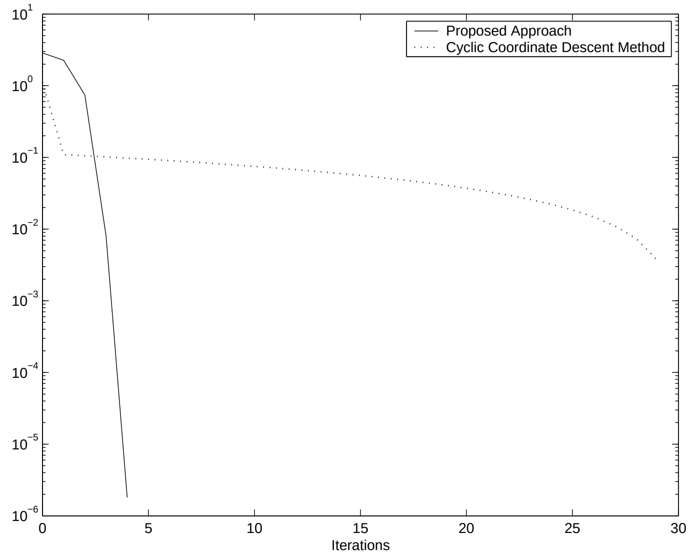


Figure 4.7: Rate of local convergence.

Remark 4.5.1 *For symmetric quadrics such as sphere and cylinder, the estimated pose is no longer belonging to SE_3 of 6 dimension, rather it belongs to the reduced space (sphere – 3 dimension, cylinder – 4 dimension). A similar algorithmic approach applies with a minor modification. For a cylinder, we have*

$$\Omega : \mathbb{R}^2 \longrightarrow \mathfrak{so}_3, \quad \Omega(\omega) = \begin{bmatrix} 0 & 0 & \omega_2 \\ 0 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}, \tag{4.32}$$

and the Lie algebra of SE_3, denoted $\mathfrak{se}_3$ is the set of 4×4 matrix of the form

$$\zeta : \mathbb{R}^4 \longrightarrow \mathfrak{se}_3 \quad \zeta(x) = \begin{bmatrix} \Omega(\omega) & v \\ 0 & 0 \end{bmatrix}, \quad x := \begin{bmatrix} \omega, \\ v \end{bmatrix} \tag{4.33}$$

where $\Omega(w) \in \mathfrak{so}_3$ and $v \in \mathbb{R}^3$, results in a 4×4 Hessian.

For a sphere, due to symmetry, we only need to find the relative translation, thus *reduce the dimension from 6 to 3.*

$$\zeta(v) = \begin{bmatrix} 0 & v \\ 0 & 1 \end{bmatrix} \tag{4.34}$$

Thus, each Newton step requires only inversion of a 3×3 *Hessian matrix. Similarly for other symmetric quadrics.*

4.6 Summary

This chapter presents a novel geometric algorithm capable of locating quadratic surfaces quickly and accurately, even in the presence of measurement noise and occlusions. The main contributions of this chapter can be summarized as follows,

- We derive a novel parameterization-based framework for minimizing a smooth function on the special Euclidean group,
- We devise efficient algorithms that are data independent at each iteration and converge at a local quadratic rate,
- We have algorithms that explicitly preserve constraints at every iteration,
- We compare the proposed algorithms with existing two steps least squares methods,
- We introduce the notion of parallel random initializations to assist in escaping local minima,
- We prove the local quadratic convergence rate of the devised Newton-like algorithms.

Chapter 5

Global Registration of Multiple 3D Point Sets

In earlier chapters our focus has been the estimation of pose information from single or stereo images. In this part of the thesis, we extract pose information from multiple range images. Most of the material in this chapter has been published in [45].

Constructing a 3D computer model of a real object from 3D surface measurement data has various applications in computer graphics, virtual reality, computer vision and reverse engineering. To construct such a model, a single view of the object is often not sufficient due to self occlusion, presence of shadow and the limited field of view of the 3D scanner. Thus multiple partial views of the object from different viewpoints are needed to describe the entire object. Typically the views are obtained from multiple scanners or a single scanner being stationed at different locations and orientation, or a fixed scanner taking time-sampled images of an object on the moving turntable. The images are often simplified as a set of features, such as points and the relative position and orientation (pose) between views are unknown. Thus, these partially overlapping views need to be registered within a common reference frame to determine the unknown relative pose.

Two-view (pairwise) registration has been well studied in the literature. It is usually performed using feature matching [22], *Iterative Closest Point* (ICP) [9] or its variant [13]. Several analytical methods for registration of two 3D point sets with known correspondences have been proposed. An overview of these techniques can be found in [44] and a comparison of these analytical methods have been presented in [52].

Multiview registration is a more difficult problem. There are two strategies towards solving the problem, local (sequential) registration and global (simultaneous) registration. The sequential registration involves the alignment of pairs of overlapping views followed by an integration step to ensure all views are combined [64, 13]. This widely used approach does not give an optimal solution because errors are accumulated and propagated as pointed out in [7, 8]. On the other hand, global registration aligns all scans at the same time by distributing the registration error evenly over all overlapping views, see [21, 72, 75, 77].

We restrict attention to the registration of multiview 3D point sets with known correspondences between overlapping images. Relatively little has been published on global registration of multiple point sets. Some work in this area can be found in [7, 81, 70]. These techniques have been compared in [17]. Recently, Williams and Bennamoun [84] have proposed an algorithm that generalizes the well known pairwise solution to global registration by iteratively optimizing individual rotation matrices using singular value decomposition.

The particular problem of multiview registration is that the function to be minimized is a nonconvex function of a set of rotations. By elementary operations, translations can be eliminated to simplify the optimization. Any algorithm that minimizes the resulting function on rotation matrices must also maintain the rotation matrix constraints during the course of an iterative procedure. Other approaches have been proposed; Pottmann *et al.* [71] suggest using the underlying affine space, applying the rigidity constraints only towards the end. Thus, standard optimization approaches either use Lagrange constraints or have to perform projection steps after each iteration to ensure that the (nonlinear)rotation matrix constraints are maintained.

Our Work In this paper, we consider the simultaneous registration of multiview 3D point sets with known correspondences between overlapping scans. We address the global registration task as an unconstrained optimization problem on a constraint manifold. Our novel algorithm involves iterative cost function reduction on the smooth manifold formed by the N-fold product of special orthogonal groups. The optimization is based on locally quadratically convergent Newton-type iterations on this constraint manifold. The proposed algorithm is fast, converges at a local quadratic rate, computation per iteration is low since the iteration cost is independent of the number of data points in each view. The Newton or Gauss steps only involve the inverse of 3×3 Hessian matrices, one for each view. In addition, we present a new closed form solution based on a singular value decomposition to achieve simultaneous registration of multiple point sets. In the noise free case, it gives correct registrations in a single step. In the presence of noise additional 3×3 singular value decompositions for projection to the constraint manifold are required. This analytical solution is a useful initial estimate for any iterative algorithm. Part of this chapter has appeared in a conference paper [45].

Chapter Outline We start with a review of prior art in the area in Section 5.1. The global point registration problem is formulated as an unconstrained optimization on a constraint manifold in Section 5.2. This is followed by a presentation of our iterative scheme in Section 5.3. Next we describe our analytic noise-free solution and our noisy initialization steps in Section 5.4. A summary of the implementation is presented in Section 5.5. Experimental evaluation and conclusions follow in Section 5.6 and Section 5.7.

5.1 Related Work

The first work on pairwise scan alignment is that of Faugeras and Herbert [22], Horn [36], and Arun *et al.* [5]. In all cases, the authors obtained simple closed form expressions for the single transformation minimizing the least squares error between the registered scans. Such pairwise schemes are used as modules in general multiview approaches such as in the Iterative Closest Point (ICP) [9] and the work of Chen and Medioni [13]. Simultaneous multiview registration schemes algorithm is considered by numerous researchers [13], [8], [21], [72], [75], [77]; among the more recent are papers by Benjemaa and Schmitt [7] and Williams and Bennamoun [84], the former group formulating the optimization in quaternion space, and the latter deriving a similar approach using matrix representations. A comparative study of simultaneous multiview registration schemes is carried out by Cunnington and Stoddart [17]; however this comparison predates the work of Williams and Bennamoun.

The ICP algorithm has become the most common method for aligning three-dimensional models based purely on the geometry. The algorithm is widely used for registering the outputs of 3D scanners, which typically only scan an object from one direction at a time. ICP starts with two meshes and an initial guess for their relative rigid-body transform, and iteratively refines the transform by repeatedly generating pairs of corresponding points on the meshes and minimizing an error metric. Generating the initial alignment can be done by a variety of heuristics, such as tracking scanner position, identification and indexing of surface features [22, 80], ”spin-image” surface signatures [43], computing principal axes of scans [25], exhaustive search for corresponding points [14, 15], or via user input.

Registration of corresponding points is not the only approach to solving multiview registration in general. ICP itself uses other heuristics to align surfaces, and in many cases matching a point to a surface can provide a better fit than simple point-point matching [74]. Due to space limitations, we will not discuss these approaches further.

The most directly relevant prior art is a paper by Adler *et al.* [3] that considers the problem of spine realignment. There, the problem is to determine correct poses for individual vertebrae on the spinal cord such that misalignment between adjacent vertebrae is minimized and a balance criterion (expressed as an affine condition over the poses) is maintained. They demonstrate that a good solution to this problem closely resembles a healthy spinal alignment. Their approach, like ours, is to view the problem as a minimization over a product manifold of SO_3, and use a Newton-type method to solve it. The specifics of their approach are different in that they derive an iterative scheme from first principles by using the covariant derivative ∇_X on the manifold; our approach uses the Lie-algebraic representation of the tangent space to yield an more direct approach.

It may be viewed that our requirement for *apriori* knowledge of point correspondences from overlapping scans is a major limitation, since this is usually not the case in practice. However, our algorithm is meant to work in conjunction with methods like ICP which provide a general framework for model registration. The crucial inner step of the ICP algorithm is to refine the transform such that it minimizes an error metric. It is this step that we consider in this paper.

5.2 Problem Formulation

Given possibly noisy surface measurements from multiple 3D images and point correspondences among overlapped images, the registration process is to find the rigid body transformations between each image coordinate frame in order to align sets of surface measurements into a reference frame.

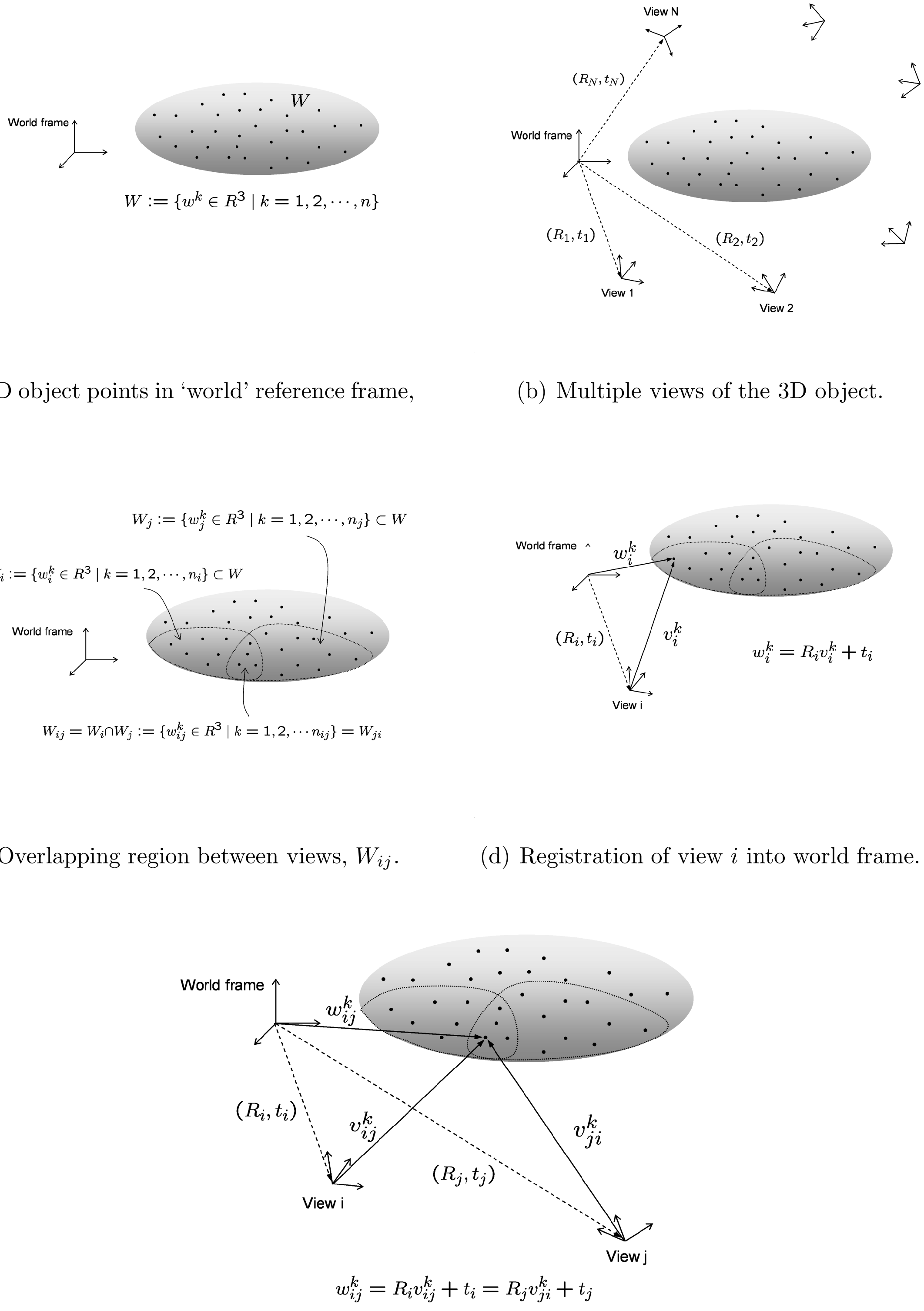
5.2.1 3D Object Points and Multiple Views

Figure 5.1 illustrates the global multiview registration problem. Consider a 3D object as a set of 3D points $W := \{w^k \in \mathbb{R}^3 \mid k = 1, 2, \cdots, n\}$ in a 'world' reference frame (Fig. 5.1(a)). Throughout the paper we indicate the k^{th} point in a set by a superscript k.

Now consider *multiple views* of the object as depicted in Fig. 5.1(b). Each view being from a different vantage point and viewing direction and each viewing being of possibly only a subset of the n 3D points. For N views, let us denote the relative rotations and translations as $(R_1, t_1), \cdots, (R_N, t_N)$, that is, relative to the 'world' reference frame, where R_i is a 3×3 rotation matrix, satisfying $R_i^\top R_i = I_3$, $\det(R_i) = +1$, and $t_i \in \mathbb{R}^3$ is a translation vector.

As illustrated in Fig. 5.1(c) and Fig. 5.1(d), the i^{th} view is limited to n_i points $W_i = \{w_i^k \in \mathbb{R}^3 \mid k = 1, 2, \cdots, n_i\} \subset W$ and is denoted $V_i = \{v_i^k \in \mathbb{R}^3 \mid k = 1, 2, \cdots, n_i\}$ and consists of the images of the n_i points in W_i with relative rotation matrices and translation vectors given by (R_i, t_i). Thus in the noise free case,

$$w_i^k = R_i v_i^k + t_i, \quad k = 1, 2, \cdots, n_i.$$


World frame
W
$W := \{w^k \in R^3 \mid k = 1, 2, \cdots, n\}$
View N
(R_N, t_N)
World frame
(R_1, t_1)
(R_2, t_2)
View 1
View 2
$W_j := \{w_j^k \in R^3 \mid k = 1, 2, \cdots, n_j\} \subset W$
$W_i := \{w_i^k \in R^3 \mid k = 1, 2, \cdots, n_i\} \subset W$
World frame
$W_{ij} = W_i \cap W_j := \{w_{ij}^k \in R^3 \mid k = 1, 2, \cdots n_{ij}\} = W_{ji}$
World frame
w_i^k
(R_i, t_i)
v_i^k
$w_i^k = R_i v_i^k + t_i$
View i
World frame
w_{ij}^k
(R_i, t_i)
v_{ij}^k
(R_j, t_j)
v_{ji}^k
View i
View j
$w_{ij}^k = R_i v_{ij}^k + t_i = R_j v_{ji}^k + t_j$

(a) 3D object points in 'world' reference frame, W.

(b) Multiple views of the 3D object.

(c) Overlapping region between views, W_{ij}.

(d) Registration of view i into world frame.

(e) Registration of view i and view j into world frame.

Figure 5.1: Global registration of multiple 3D point sets: given multiple 3D point sets, each represents a partial view of a 3D object. Assuming that point correspondences between overlapping views are known, the task is to register all views into a single reference frame simultaneously.

Let $W_{ij} = W_i \cap W_j$ be the set of n_{ij} points in W_i for which there are corresponding points in W_j, for $i, j = 1, \cdots, N$. That is, $W_{ij} = W_{ji}$ consists of $n_{ij} = n_{ji}$ points $w_{ij}^k = w_{ji}^k \in \mathbb{R}^3$, $k = 1, \cdots, n_{ij}$. In view V_i the set of images of these points is denoted $V_{ij} := \{v_{ij}^k \in \mathbb{R}^3 \mid k = 1, 2, \cdots, n_{ij}\} \subset V_i$ and of course for view V_j it is denoted $V_{ji} := \{v_{ji}^k \in \mathbb{R}^3 \mid k = 1, 2, \cdots, n_{ij}\} \subset V_j$. In the noise free case, it is immediate that

$$w_{ij}^k = R_i v_{ij}^k + t_i = R_j v_{ji}^k + t_j, \quad \forall\, i, j = 1, 2, \cdots, N,\ k = 1, 2, \cdots, n_{ij}, \tag{5.1}$$

as depicted in Fig. 5.1(e).

5.2.2 Global Registration Error

When there is measurement noise, it makes sense to work with a cost function that penalizes the error $(R_i v_{ij}^k + t_i) - (R_j v_{ji}^k + t_j)$ for all $i, j = 1, 2, \cdots, N$ and $k = 1, 2, \cdots, n_{ij}$. Trivially the error is zero for $i = j$. The cost index for all the registrations which first comes to mind is given by the sum of the squared Euclidean distances between the corresponding points in all overlaps,

$$\begin{aligned} g &= \sum_{i=1}^{N} \sum_{j=i+1}^{N} \sum_{k=1}^{n_{ij}} \|(R_i v_{ij}^k + t_i) - (R_j v_{ji}^k + t_j)\|^2, \\ &= \sum_{i=1}^{N} \sum_{j=i+1}^{N} \sum_{k=1}^{n_{ij}} (\|R_i v_{ij}^k - R_j v_{ji}^k\|^2 + 2(t_i - t_j)^\top (R_i v_{ij}^k - R_j v_{ji}^k) + \|t_i - t_j\|^2). \end{aligned} \tag{5.2}$$

Before optimizing this index, we first reformulate the index in a more convenient notation. Let $e_i :=$ i^{th} column of $N \times N$ identity matrix, I_N, $e_{ij} := e_i - e_j$,

$$\mathcal{R} := \begin{bmatrix} R_1 & R_2 & \cdots & R_N \end{bmatrix} \in \mathbb{R}^{3 \times 3N}, \quad \mathcal{T} := \begin{bmatrix} t_1 & t_2 & \cdots & t_N \end{bmatrix} \in \mathbb{R}^{3 \times N}, \tag{5.3}$$

then we have

$$R_i = \mathcal{R}(e_i^\top \otimes I_3), \quad t_i = \mathcal{T} e_i, \quad t_i - t_j = \mathcal{T} e_{ij}.$$

Also define,

$$\begin{bmatrix} A & B \\ B^\top & C \end{bmatrix} = \sum_{i=1}^{N} \sum_{j=i+1}^{N} \sum_{k=1}^{n_{ij}} \begin{bmatrix} a_{ij}^k \\ e_{ij} \end{bmatrix} \begin{bmatrix} a_{ij}^{k\top} & e_{ij}^\top \end{bmatrix} \geq 0, \quad a_{ij}^k := (e_i \otimes I_3) v_{ij}^k - (e_j \otimes I_3) v_{ji}^k.$$

Now, simple manipulations show that the registration error cost function (5.2) can be rewritten in convenient matrix notation as

$$g(\mathcal{R}, \mathcal{T}) = \operatorname{tr}(\mathcal{R}A\mathcal{R}^\top + 2\mathcal{R}B\mathcal{T}^\top + \mathcal{T}C\mathcal{T}^\top) = \operatorname{tr}\left(\begin{bmatrix} \mathcal{R} & \mathcal{T} \end{bmatrix} \begin{bmatrix} A & B \\ B^\top & C \end{bmatrix} \begin{bmatrix} \mathcal{R}^\top \\ \mathcal{T}^\top \end{bmatrix}\right) \geq 0, \tag{5.4}$$

or equivalently, as

$$g(\mathcal{R}, \mathcal{T}) = \operatorname{tr}(\mathcal{R}A\mathcal{R}^\top) + 2\operatorname{vec}^\top(\mathcal{T})\operatorname{vec}(\mathcal{R}B) + \operatorname{vec}^\top(\mathcal{T})(C \otimes I_3)\operatorname{vec}(\mathcal{T}), \tag{5.5}$$

since $\operatorname{tr}(X^\top Y) = \operatorname{vec}^\top(X)\operatorname{vec}(Y)$.

5.2.3 A More Compact Reformulation

Optimal Translation

Observe that (5.5) is readily minimized over all $\operatorname{vec}(\mathcal{T}) \in \mathbb{R}^{3N}$, or $\mathcal{T} \in \mathbb{R}^{3\times N}$. Thus,

$$\mathcal{T}^*(\mathcal{R}) := \arg\min_{\mathcal{T}} g(\mathcal{R}, \mathcal{T}), \tag{5.6}$$

satisfies

$$\operatorname{vec}(\mathcal{T}^*(\mathcal{R})) = -(C^\dagger \otimes I_3)\operatorname{vec}(\mathcal{R}B) = -\operatorname{vec}(\mathcal{R}BC^\dagger) \;\;\Leftrightarrow\;\; \mathcal{T}^*(\mathcal{R}) = -\mathcal{R}BC^\dagger. \tag{5.7}$$

Note that the data matrix C is singular, since $C1_N = 0$, where 1_N is the $N \times 1$ vector consisting of unity elements. Thus the pseudo inverse $C^\dagger$ is used. Clearly, the absolute transformations can not be recovered. Since rotating each view by R_0 and then translating it by t_0 does not change the value of the cost function, as now indicated,

$$\|\,\big(R_0(R_i v_{ij}^k + t_i) + t_0\big) - \big(R_0(R_j v_{ji}^k + t_j) + t_0\big)\,\| = \|(R_i v_{ij}^k + t_i) - (R_j v_{ji}^k + t_j)\|,$$

it makes sense to estimate the rigid body transformations relative to (say) the first reference frame, by fixing (R_1, t_1) as $(I_3, 0_3)$ where 0_3 denotes the 3×1 zero vector.

Global Registration Error Independent of Translation

From the previous subsection, we see that optimizing the translation can be decoupled from optimizing the rotation. Substituting $\mathcal{T}^*(\mathcal{R})$ from (5.7) into (5.4) leads to a registration error cost function depending only on rotations,

$$f(\mathcal{R}) := g(\mathcal{R}, \mathcal{T}(\mathcal{R})) = \operatorname{tr}(\mathcal{R}\mathcal{M}\mathcal{R}^\top) = \operatorname{vec}^\top(\mathcal{R}^\top)(I_3 \otimes \mathcal{M})\operatorname{vec}(\mathcal{R}^\top), \tag{5.8}$$

where $\mathcal{M} := A - BC^\dagger B^\top$.

Problem Statement Given data matrix $\mathcal{M}$, recover the rotation matrices $\mathcal{R}$ and subsequently $\mathcal{T}$, that allow one to register all scan data simultaneously.

5.3 Optimization on the N-fold Product of Special Orthogonal Group

5.3.1 Geometry of the N-fold Product of Special Orthogonal Group

Here we review the geometry of the special orthogonal group and its product manifold. Let SO_3 denote the group of 3×3 orthogonal matrices with determinant $+1$. Recall that SO_3 is a Lie group and its associated Lie algebra $\mathfrak{so}_3$ is the set of 3×3 skew symmetric matrices of the form,

$$\Omega = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}. \tag{5.9}$$

There is a well known isomorphism from the Lie algebra $(\mathbb{R}^3, \times)$ to the Lie algebra $(\mathfrak{so}_3, [.,.])$, where × denotes the cross product and $[.,.]$ denotes the matrix commutator. This allows one to identify $\mathfrak{so}_3$ with $\mathbb{R}^3$ using the mapping in (5.9), which maps a vector $\omega = \begin{bmatrix}\omega_x & \omega_y & \omega_z\end{bmatrix} \in \mathbb{R}^3$ to a matrix $\Omega \in \mathfrak{so}_3$. Denoting

$$Q_x := \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}, \quad Q_y := \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}, \quad Q_z := \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{5.10}$$

note that

$$\Omega = Q_x\omega_x + Q_y\omega_y + Q_z\omega_z.$$

Our interest here is a product manifold of SO_3 which is a smooth manifold of dimension $3N$, given by

$$SO_3^N = \overbrace{SO_3 \times \cdots \times SO_3}^{N \text{ times}}.$$

Tangent Space of SO_3^N

Recall that the tangent space of SO_3 at R_i for $i = 1, 2, \cdots N$ is given as $\mathrm{T}_{R_i}SO_3 = \{R_i\Omega_i \mid \Omega_i \in \mathfrak{so}_3\}$ and the corresponding affine tangent space is $\mathrm{T}^{\text{aff}}_{R_i}SO_3 = \{R_i + R_i\Omega_i \mid \Omega_i \in \mathfrak{so}_3\}$. Note that direct sum $\oplus$ of matrices is equal to a block diagonal matrix with the individual matrices as the diagonal blocks and define

$$\widetilde{\Omega} := \Omega_1 \oplus \Omega_2 \oplus \cdots \oplus \Omega_N, \quad \Omega_i \in \mathfrak{so}_3. \tag{5.11}$$

Due to isomorphism, the tangent space of SO_3^N at $\mathcal{R} = [R_1\ R_2\ \cdots\ R_N] \in SO_3^N$ can be identified as,

$$\mathrm{T}_{\mathcal{R}}SO_3^N = \mathcal{R}\widetilde{\Omega}, \tag{5.12}$$

and the affine tangent space is

$$\mathrm{T}^{\text{aff}}_{\mathcal{R}}SO_3^N = \mathcal{R} + \mathcal{R}\widetilde{\Omega}. \tag{5.13}$$

Local Parameterization of SO_3^N

For every point $R_i \in SO_3$, there exists a smooth exponential map

$$\mu_{R_i} : \mathbb{R}^{3N} \to SO_3, \quad \omega_i \mapsto R_i e^{\Omega(\omega_i)},$$

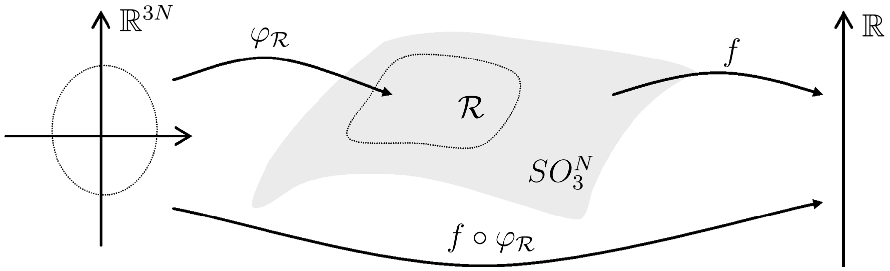


Figure 5.2: The mapping $\varphi_{\mathcal{R}}$ is the local parameterization of SO_3^N around point $\mathcal{R}$ such that $\varphi_{\mathcal{R}}(0) = \mathcal{R}$, f is the smooth function defined on SO_3^N and $f \circ \varphi_{\mathcal{R}}$ is f expressed in local parameter space $\mathbb{R}^{3N}$.

which is a diffeomorphism about the origin in $\mathbb{R}^3$. Due to isomorphism, every point $\mathcal{R} \in SO_3^N$ can be locally parameterized by the smooth map

$$\begin{aligned} \varphi_{\mathcal{R}} : \mathbb{R}^{3N} &\rightarrow SO_3^N, \\ \omega = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_N \end{bmatrix} &\mapsto \mathcal{R}\left(e^{\Omega(\omega_1)} \oplus e^{\Omega(\omega_2)} \oplus \cdots \oplus e^{\Omega(\omega_N)}\right) = \mathcal{R}e^{\widetilde{\Omega}(\omega)}. \end{aligned} \tag{5.14}$$

5.3.2 Cost Function

Cost Function on SO_3^N

Recall $\mathcal{M}$ from (5.8), consider the smooth function,

$$f : SO_3^N \rightarrow \mathbb{R}, \quad f(\mathcal{R}) = \mathrm{tr}(\mathcal{R}\mathcal{M}\mathcal{R}^\top) = \mathrm{vec}^\top(\mathcal{R}^\top)(I_3 \otimes \mathcal{M})\mathrm{vec}(\mathcal{R}^\top). \tag{5.15}$$

Minimization of this function penalizes the alignment error among all range images simultaneously.

Local Cost Function

The cost function f at $\mathcal{R} \in SO_3^N$ expressed in local parameter space using the smooth local parameterization $\varphi_{\mathcal{R}}$ defined in (5.14) is given by,

$$f \circ \varphi_{\mathcal{R}} : \mathbb{R}^{3N} \rightarrow \mathbb{R}, \quad f \circ \varphi_{\mathcal{R}}(\omega) = \mathrm{tr}(\mathcal{R}e^{\widetilde{\Omega}(\omega)}\mathcal{M}e^{\widetilde{\Omega}(\omega)^\top}\mathcal{R}^\top). \tag{5.16}$$

Quadratic Model of Local Cost Function

The second order Taylor approximation of $f \circ \varphi_{\mathcal{R}}$ about $0 \in \mathbb{R}^{3N}$ in direction ω is

$$\begin{gathered} j_0^{(2)}(f \circ \varphi_{\mathcal{R}}) : \mathbb{R}^{3N} \to \mathbb{R}, \\ \omega \mapsto \left((f \circ \varphi_{\mathcal{R}})(t\omega) + \frac{\mathrm{d}}{\mathrm{d}t}(f \circ \varphi_{\mathcal{R}})(t\omega) + \frac{1}{2}\frac{\mathrm{d}^2}{\mathrm{d}t^2}(f \circ \varphi_{\mathcal{R}})(t\omega) \right)\bigg|_{t=0}. \end{gathered}$$

The mapping consists of three terms,

(i) a constant term,

$$(f \circ \varphi_{\mathcal{R}})(t\omega)|_{t=0} = \operatorname{tr}(\mathcal{R}\mathcal{M}\mathcal{R}^\top).$$

(ii) a linear term,

$$\frac{\mathrm{d}}{\mathrm{d}t}(f \circ \varphi_{\mathcal{R}})(t\omega)\bigg|_{t=0} = 2\operatorname{tr}(\mathcal{R}\widetilde{\Omega}\mathcal{M}\mathcal{R}^\top) = 2\omega^\top \nabla_{f\circ\varphi_{\mathcal{R}}}(0),$$

Recall (5.9)–(5.11) and let $\operatorname{vec}(\widetilde{\Omega}^\top) := \widetilde{Q}\omega$, the explicit formula for the gradient of $f \circ \varphi_{\mathcal{R}}$ evaluated at $0 \in \mathbb{R}^{3N}$ is

$$\nabla_{f\circ\varphi_{\mathcal{R}}}(0) = J^\top \operatorname{vec}(\mathcal{M}\mathcal{R}^\top), \tag{5.17}$$

where

$$J := (\mathcal{R} \otimes I_{3N})\widetilde{Q}, \quad \widetilde{Q} := Q_{e_1} \oplus Q_{e_2} \oplus \cdots \oplus Q_{e_N}, \quad Q_{e_i} := \begin{bmatrix} e_i \otimes Q_x \\ e_i \otimes Q_y \\ e_i \otimes Q_z \end{bmatrix}. \tag{5.18}$$

(iii) a quadratic term which consists of a sum of two terms. The first term is given as

$$\operatorname{tr}(\mathcal{R}\widetilde{\Omega}\mathcal{M}\widetilde{\Omega}^\top\mathcal{R}^\top) = \omega^\top \widehat{\mathrm{H}}_{f\circ\varphi_{\mathcal{R}}}(0)\omega,$$

and the second quadratic term is

$$\operatorname{tr}(\mathcal{R}\widetilde{\Omega}^2\mathcal{M}\mathcal{R}^\top) = \operatorname{vec}^\top(\widetilde{\Omega}^\top)\operatorname{vec}(\mathcal{M}\mathcal{R}^\top\mathcal{R}\widetilde{\Omega}) = \omega^\top \widetilde{\mathrm{H}}_{f\circ\varphi_{\mathcal{R}}}(0)\omega$$

Thus the Hessian of $f \circ \varphi_{\mathcal{R}}$ evaluated at zero is

$$\mathrm{H}_{f\circ\varphi_{\mathcal{R}}}(0) = \widehat{\mathrm{H}}_{f\circ\varphi_{\mathcal{R}}}(0) + \widetilde{\mathrm{H}}_{f\circ\varphi_{\mathcal{R}}}(0), \tag{5.19}$$

where

$$\widehat{\mathrm{H}}_{f\circ\varphi_{\mathcal{R}}}(0) = J^\top(I_3 \otimes \mathcal{M})J \ \geq 0, \quad \widetilde{\mathrm{H}}_{f\circ\varphi_{\mathcal{R}}}(0) = -\widetilde{Q}^\top(I_{3N} \otimes \mathcal{M}\mathcal{R}^\top\mathcal{R})\widetilde{Q}. \tag{5.20}$$

5.3.3 Algorithm

The proposed algorithm consists of the iteration,

$$s = \pi_2 \circ \pi_1 : SO_3^N \to SO_3^N, \tag{5.21}$$

where π_1 maps a point $\mathcal{R} \in SO_3^N$ to an element in the affine tangent space $\mathrm{T}_{\mathcal{R}}^{\mathrm{aff}} SO_3^N$ that minimizes $j_0^{(2)}(f \circ \varphi_{\mathcal{R}})(0)$ and π_2 projects that element back to SO_3^N by means of the parametrization $\varphi_{\mathcal{R}}$, as illustrated in Fig. 5.3.

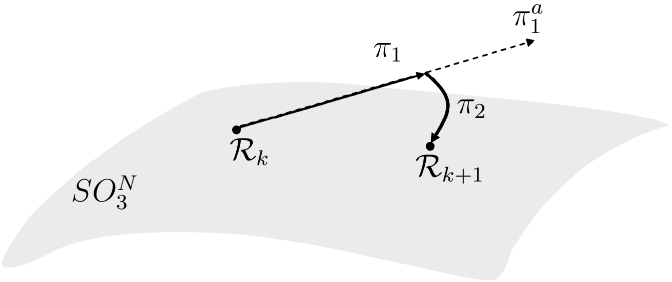


Figure 5.3: The proposed algorithm first maps a point $\mathcal{R}_k \in SO_3^N$ to an element of the affine tangent space $\mathrm{T}_{\mathcal{R}_k}^{\mathrm{aff}} SO_3^N$ via π_1, followed by step π_2 to project that vector back to the manifold.

Optimization in Local Parameter Space, π_1

Optimization in local parameter space consists of two steps, first calculate a suitable descent direction and then search for a step length that ensures reduction in cost function, as described by the mapping

$$\pi_1 = \pi_1^b \circ \pi_1^a : SO_3^N \to \mathrm{T}^{\mathrm{aff}} SO_3^N. \tag{5.22}$$

Here π_1^a is used to obtain a descent direction,

$$\pi_1^a : SO_3^N \to \mathrm{T}^{\mathrm{aff}} SO_3^N, \;\; \mathcal{R} \mapsto \mathcal{R} + \mathcal{R}\widetilde{\Omega}(\omega_{\mathrm{opt}}(\mathcal{R})), \tag{5.23}$$

where $\omega_{\mathrm{opt}} \in \mathbb{R}^{3N}$ as a function of $\mathcal{R} = \varphi_{\mathcal{R}}(0)$ can be given by the Newton direction when $\mathrm{H}_{f\circ\varphi_{\mathcal{R}}}(0) > 0$ as,

$$\omega_{\mathrm{opt}}^{\mathrm{Newton}}(\mathcal{R}) = -[\mathrm{H}_{f\circ\varphi_{\mathcal{R}}}(0)]^{-1}\nabla_{f\circ\varphi_{\mathcal{R}}}(0), \tag{5.24}$$

or by a Gauss direction otherwise, as

$$\omega_{\text{opt}}^{\text{Gauss}}(\mathcal{R}) = -[\widehat{\mathrm{H}}_{f\circ\varphi_{\mathcal{R}}}(0)]^{-1}\nabla_{f\circ\varphi_{\mathcal{R}}}(0). \tag{5.25}$$

Once an optimal direction is computed, an approximate one dimensional line search is carried out in this direction. We proceed with a search on the scalar $\lambda > 0$ which ensures that the cost function $f \circ \varphi_{\mathcal{R}}(\lambda\omega_{\text{opt}})$ is reduced at every step, giving rise to the mappings,

$$\begin{aligned} \pi_1^b : \mathrm{T}^{\text{aff}}SO_3^N &\to \mathrm{T}^{\text{aff}}SO_3^N, \\ \mathcal{R} + \mathcal{R}\widetilde{\Omega}(\omega_{\text{opt}}(\mathcal{R})) &\mapsto \mathcal{R} + \mathcal{R}\widetilde{\Omega}(\lambda_{\text{opt}}\omega_{\text{opt}}(\mathcal{R})), \end{aligned} \tag{5.26}$$

where λ_{opt} is the step length that reduces the cost function in direction ω_{opt}, and is found using the simple backtracking line search. Since we are using a descent direction, for sufficiently small step size, the cost function will go downhill.

Remark 5.3.1 *Given a descent direction ω_{opt} for function $f \circ \varphi_{\mathcal{R}}$ at $0 \in \mathbb{R}^{3N}$,*

- *An exact line search is,*

$$\lambda_{\text{opt}} = \arg\min_{\lambda>0} f \circ \varphi_{\mathcal{R}}(\lambda\omega_{\text{opt}}), \tag{5.27}$$

- *A backtracking line search with parameters $\alpha \in (0, 0.5), \beta \in (0, 1)$, then starting at $\lambda := 1$, we proceed as follows,*

$$\begin{aligned} \text{While} \quad & f \circ \varphi_{\mathcal{R}}(\lambda\omega_{\text{opt}}) > f \circ \varphi_{\mathcal{R}}(0) + \alpha\lambda[\nabla_{f\circ\varphi_{\mathcal{R}}}(0)]^\top\omega_{\text{opt}}, \\ \text{do} \quad & \lambda := \beta\lambda, \end{aligned}$$

and $\lambda_{\text{opt}} := \lambda$.

Projecting Back via Parametrization

Once the descent direction and downhill step size is obtained, we project it back to the manifold via the parametrization,

$$\begin{aligned} \pi_2 : \mathrm{T}^{\text{aff}}SO_3^N &\to SO_3^N, \\ \mathcal{R} + \widetilde{\Omega}(\lambda_{\text{opt}}\omega_{\text{opt}}(\mathcal{R})) &\mapsto \mathcal{R}e^{\widetilde{\Omega}(\lambda_{\text{opt}}\omega_{\text{opt}}(\mathcal{R}))} \\ &= \mathcal{R}\left(e^{\Omega(\lambda_{\text{opt}}\omega_1^{\text{opt}}(R_1))} \oplus \cdots \oplus e^{\Omega(\lambda_{\text{opt}}\omega_N^{\text{opt}}(R_N))}\right), \end{aligned} \tag{5.28}$$

since $\omega_{\text{opt}}(\mathcal{R}) = \begin{bmatrix} \omega_1^{\text{opt}}(R_1)^\top & \cdots & \omega_N^{\text{opt}}(R_N)^\top \end{bmatrix}^\top$.

5.3.4 Convergence Analysis of the Algorithm

Local Quadratic Convergence of the Algorithm

Theorem 5.3.1 *Let $\mathcal{R}_* \in SO_3^N$ be a nondegenerate minimum of the smooth function $f : SO_3^N \to \mathbb{R}$ defined in* (5.8)*. Let $\mathcal{R}_k$ be a point in an open neighbourhood of $\mathcal{R}_*$. Consider the proposed iteration on SO_3^N,*

$$\mathcal{R}_{k+1} = s(\mathcal{R}_k), \quad s = \pi_2 \circ \pi_1, \tag{5.29}$$

where π_1 is given by the Newton direction defined in (5.24)*, π_2 involves projection back to SO_3^N via the smooth exponential map of* (5.28)*. Then the point sequence generated by s converges quadratically to $\mathcal{R}_*$.*

Proof 5.3.1 *See Appendix A.5.*

5.4 Algorithm Initialization

Here we present a new closed form solution based on singular value decomposition that simultaneously registers all range images. This is used as the initial estimate for the proposed iterative algorithm of the previous section. In the noise free case, it gives optimal estimates of the rotation matrices in a single step. Moreover, these are the desired (exact) rotation matrices. In the presence of noise, this step leads to an 'optimal' matrix $\mathcal{R} \in \mathbb{R}^{3\times 3N}$ but such that $R_i \notin SO_3$ for some or all i typically. Thus, an additional projection step to the manifold is required.

Noise Free Solution

In the noise free case, for $\mathcal{R} \in SO_3^N$, the optimal value of the cost function (5.8) is zero, as

$$\text{vec}^\top(\mathcal{R}^\top)\text{vec}(\mathcal{M}\mathcal{R}^\top) = 0 \quad \Rightarrow \text{vec}(\mathcal{M}\mathcal{R}^\top) = 0 \quad \Rightarrow \quad \mathcal{M}\mathcal{R}^\top = 0. \tag{5.30}$$

Since $\mathcal{M}$ is symmetric, a singular value decomposition gives

$$\mathcal{M} = U\Sigma U^\top = \begin{bmatrix} U_a & U_b \end{bmatrix} \begin{bmatrix} \Sigma_a & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_a^\top \\ U_b^\top \end{bmatrix} \quad \Rightarrow \quad \mathcal{M}U_b = 0. \tag{5.31}$$

To obtain $\mathcal{R}$ such that $R_1 = I_3$, let $\widehat{U} := \begin{bmatrix} I_3 & 0 \end{bmatrix} U_b$, then the closed form solution is

$$\mathcal{R} = \widehat{U}^{-\top} U_b^\top. \tag{5.32}$$

Initialization for Noisy Case

In the presence of noise, the optimal cost function is no longer equal to zero. In this case, U_b is chosen to be the set of right singular vectors associated with 3 least singular values of $\mathcal{M}$, which may not be zero. These singular vectors might not be on SO_3^N. Thus, an additional projection step is required. Denoting $G_i := \widehat{U}^{-\top} U_b (e_i \otimes I_3)$, we have

$$R_i^{\text{opt}} = \arg \min_{R_i \in SO_3} \|R_i - G_i\| = \arg \max_{R_i \in SO_3} \operatorname{tr}(R_i^\top G_i). \tag{5.33}$$

By applying a singular value decomposition on G_i, we obtain

$$G_i = W \Lambda Z^\top, \quad R_i^{\text{opt}} = W \begin{bmatrix} I_2 & 0 \\ 0 & \det(WZ^\top) \end{bmatrix} Z^\top, \tag{5.34}$$

where $\det(R_i^{\text{opt}}) = +1$.

5.5 Implementation of Algorithm

Start with an initial estimate of the rotation matrices $\mathcal{R} = [R_1 \; R_2 \; \cdots \; R_N] \in SO_3^N$ obtained from the initialization algorithm of the previous section.

Step 1: Carry out the optimization step,

- Compute the gradient $\nabla_{f \circ \varphi_{\mathcal{R}}}(0)$ and the Hessian $\mathrm{H}_{f \circ \varphi_{\mathcal{R}}}(0)$ via (5.17), (5.19) respectively.
- If $\mathrm{H}_{f \circ \varphi_{\mathcal{R}}}(0) > 0$,
  Compute the Newton step, $\omega_{\text{opt}} = -[\mathrm{H}_{f \circ \varphi_{\mathcal{R}}}(0)]^{-1} \nabla_{f \circ \varphi_{\mathcal{R}}}(0)$,
  otherwise compute the Gauss step $\omega_{\text{opt}} = -[\widehat{\mathrm{H}}_{f \circ \varphi_{\mathcal{R}}}(0)]^{-1} \nabla_{f \circ \varphi_{\mathcal{R}}}(0)$.

- Compute the optimum step size λ_{opt} in direction ω_{opt} using a backtracking line search, as described in *Remark 5.3.1*.

Step 2: Carry out the projection step, $\widehat{\mathcal{R}} = \mathcal{R}\left(e^{\Omega(\lambda_{\text{opt}}\omega_1^{\text{opt}})} \oplus \cdots \oplus e^{\Omega(\lambda_{\text{opt}}\omega_N^{\text{opt}})}\right)$.

Step 3: Set $\mathcal{R} = \widehat{\mathcal{R}}$, go back to Step 1 if $\|\nabla_{f\circ\varphi_{\mathcal{R}}}(0)\| > \epsilon$, a prescribed accuracy.

Remark 5.5.1 *To reduce computational effort per iteration of the algorithm, the sparse matrix J (5.18) for Hessian and gradient computation can be manipulated further as follows. Recalling Ω from (5.9), then*

$$\begin{aligned} J &= \begin{bmatrix}(R_1 \otimes I_{3N})Q_{e_1} & (R_2 \otimes I_{3N})Q_{e_2} & \cdots & (R_N \otimes I_{3N})Q_{e_N}\end{bmatrix} \\ &= \begin{bmatrix} \Omega(\bar{e}_1^\top R_1) \oplus \Omega(\bar{e}_1^\top R_2) \oplus \cdots \oplus \Omega(\bar{e}_1^\top R_N) \\ \Omega(\bar{e}_2^\top R_1) \oplus \Omega(\bar{e}_2^\top R_2) \oplus \cdots \oplus \Omega(\bar{e}_2^\top R_N) \\ \Omega(\bar{e}_3^\top R_1) \oplus \Omega(\bar{e}_3^\top R_2) \oplus \cdots \oplus \Omega(\bar{e}_3^\top R_N) \end{bmatrix}. \end{aligned} \tag{5.35}$$

5.6 Experimental Evaluation

We now present an experimental study of our algorithm, focusing primarily on the quality of the registrations it produces, and the convergence rate of the method.

Methods We will compare our algorithm (which we will refer to as Manifold-based registration, **MBR**) to the schemes proposed by Benjemaa and Schmitt [7] (**QUAT**) and Williams and Bennamoun [84] (**MAT**). **MBR** and **MAT** are matrix based and are written in `MATLAB`. **MAT**, which uses quaternions in its formulation, is written in C. We used a maximum iteration limit of 1000 for all the methods. Our method of comparison between various algorithms will be based on both visual quality as well as iteration counts and error convergence rates (we will not use clock time).

Data Our first data set consists of actual 3D models from the Stanford 3D Scanning Repository. For each of three models, we generated a collection of views as follows: we first generate a unit vector (representing a view) and extracted the points on all front-facing triangles with respect to this view. Next, each view is randomly rotated and translated into a local coordinate system. Finally, each point in each view is randomly perturbed using a Gaussian noise model. This yields a collection of views that possess a global noisy registration. With this data, we have ground truth (exact correspondences) since we have the original model. Table 5.1 summarizes the statistics of this data.

Our second data set consists of 3D range scan data from the Digital Michelangelo Project [50]. The individual scans come with an original alignment (stored in `.xf` files). We perform ICP on pairs of scans, using the routines built into `scanalyze`, and retain all pairs of scans that have at least three points in common as determined by ICP. In each instance, we run ICP five times and take the best alignment thus generated (each instance of ICP runs for ten iterations). The model of correspondence used is point-point.

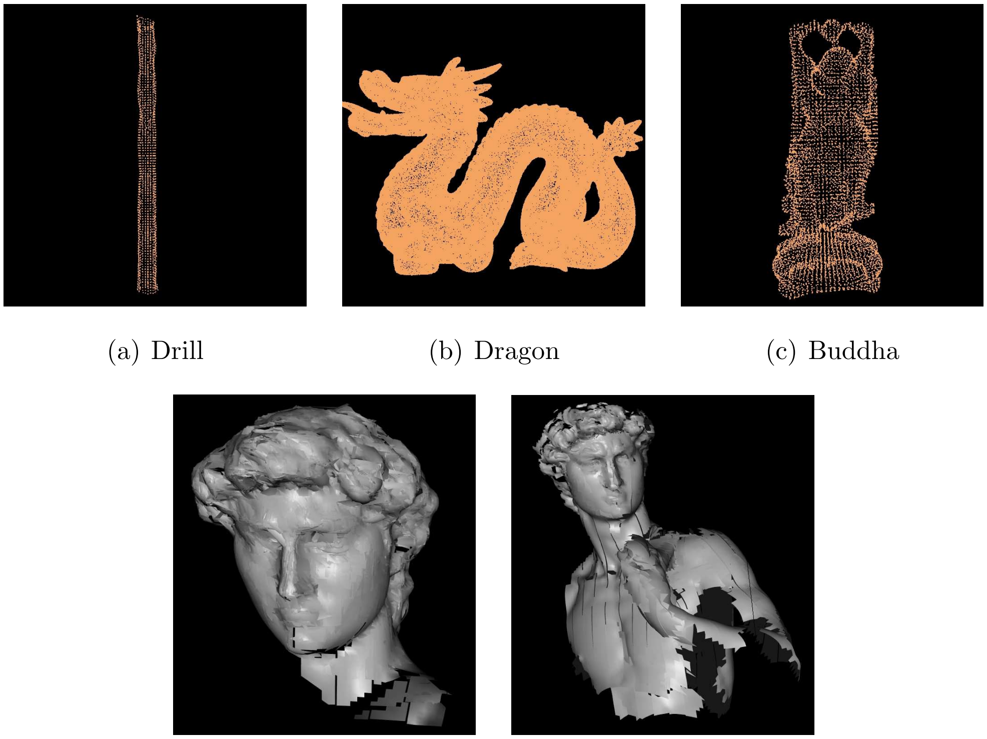
Model	Number of vertices	Number of scans	Total size of all scans	Number of view pairs generated	Time (in secs.) per iteration (**MBR**)
DRILL	1961	20	23298	77	0.015
DRAGON	100250	20	1142487	98	0.016
BUDDHA	32328	50	252580	526	0.093

Table 5.1: Statistics for the synthetic 3D models used for global registration

5.6.1 3D Models

We have run the three algorithms on the view pairs obtained from the three 3D models. In Figure 5.4 (a)–(c), we show the output registrations obtained by **MBR**. For these examples, the other two schemes produced similar registrations, although with higher error. In Table 5.2, we compare the performance of the three schemes on the models, in terms of both the number of iterations till convergence, and the final error. The final error is computed by evaluating the function defined in (5.5).

What is striking about the numbers is that although mostly the other approaches yield comparable error (except for DRILL), their iteration counts are orders of magnitude higher than that of our scheme. This is a clear demonstration of locally quadratic



(a) Drill

(b) Dragon

(c) Buddha

(d) David Head

(e) David Head and Bust

Figure 5.4: Registrations produced by our Optimization-on-a-Manifold algorithm, **MBR**, on synthetic and real data sets. The real data set was obtained from the Digital Michelangelo project at Stanford University.

convergence properties of our scheme.

Factors that influence iteration counts We investigated other factors that might affect algorithm performance. Some of the parameters that influence iteration counts are the density of the correspondence graph *i.e.* how many view pairs are provided, and the strength of match for each pair (average number of points in each view pair).

In all cases, the number of iterations required by our method to converge was unaffected by these parameters. However, for the other methods, we noticed a fairly weak correlation between the density of the correspondence graph and the number of iterations needed; as the graph got denser, implying a more constrained system, the number of iterations needed to converge reduced. For example, the iteration counts

	MBR		**MAT**		**QUAT**	
	Iter.	Error	Iter.	Error	Iter.	Error
Drill	**2**	3.5e-7	47	3.5e-7	48	7e-7
Dragon	**4**	5e-3	933	1e-2	1000	1e-2
Buddha	**2**	2e-4	534	2e-3	718	3e-3

Table 5.2: Performance of the three registration methods - our Optimization-on-a-Manifold method **MBR**, Williams and Bennamoun's SVD-based method **MAT** and Benjemaa and Schmitt's Quaternion-based method **QUAT** on the synthetic data sets

for **MAT** and **QUAT** went from close to 1000 (for a sparse graph in the Dragon) to 47 (for a dense graph in the Drill).

Cost per iteration We do not provide a comparison of actual time per iteration for the three methods because they have been implemented on different platforms. However, **MBR** and **MAT** exhibit cubic dependence on the number of scans (for N scans, each iteration takes $O(N^3)$ time), while **QUAT** take quadratic time per iteration at the expense of many more iterations. There is no running time dependence on the actual size of the model or size of each scan; there is however a preprocessing cost dependent on the total size of the corresponding points. Using our Matlab code, we measured the time per iteration only for our algorithm, **MBR**, and is shown in the last column of Table 5.1. All timing measurements were performed on a PC running Windows XP with a 2.8 GHz Pentium IV processor and 512 MBytes of RAM. For the models we tried in this paper, we roughly had anywhere from 8 to 80 corresponding points between pairs of scans.

5.6.2 Range Scan Data

Having evaluated the performance of our scheme in relation to prior art in a controlled setting where ground truth (exact correspondences) are known, we now present the results of running the schemes on range scan data. We focus on the model of David, specifically the views corresponding to the head and bust region. After implementing the view generation procedure described earlier, we obtain a 10-scan instance of the bust and a 38-scan instance of the head. We also use a 21-

scan instance that has bad starting alignment. The registration produced by our algorithm for 10-scan instance of the bust and a 38-scan instance of the head are shown in Figure 5.4 (d) and (e) respectively.

Figure 5.5 shows the registrations obtained by **MBR**, **MAT**, and **QUAT**. In all cases, the registration produced by our algorithm is quite plausible. The other methods do not fare so well; a typical problem is that the two halves of David's face do not register properly, creating the false effect of two heads. Table 5.3 summarizes the performance of the three algorithms in terms of iteration counts. For absolute times per iteration, our algorithm, **MBR**, took 9 milliseconds for the 10-scan instance of the bust, 47 milliseconds for the 38-scan instance of the head and 20 milliseconds for the 21-scan instance of the bust with bad initial alignment.

	MBR	**MAT**	**QUAT**
	Iter.	Iter.	Iter.
Head	48	247	1000
Bust	12	1000	1000
Bust - Bad Alignment	81	1000	1000

Table 5.3: Performance of the three registration methods - our Optimization-on-a-Manifold method **MBR**, Williams and Bennamoun's SVD-based method **MAT** and Benjemaa and Schmitt's Quaternion-based method **QUAT** on the David model - courtesy of the Digital Michelangelo project

5.7 Summary

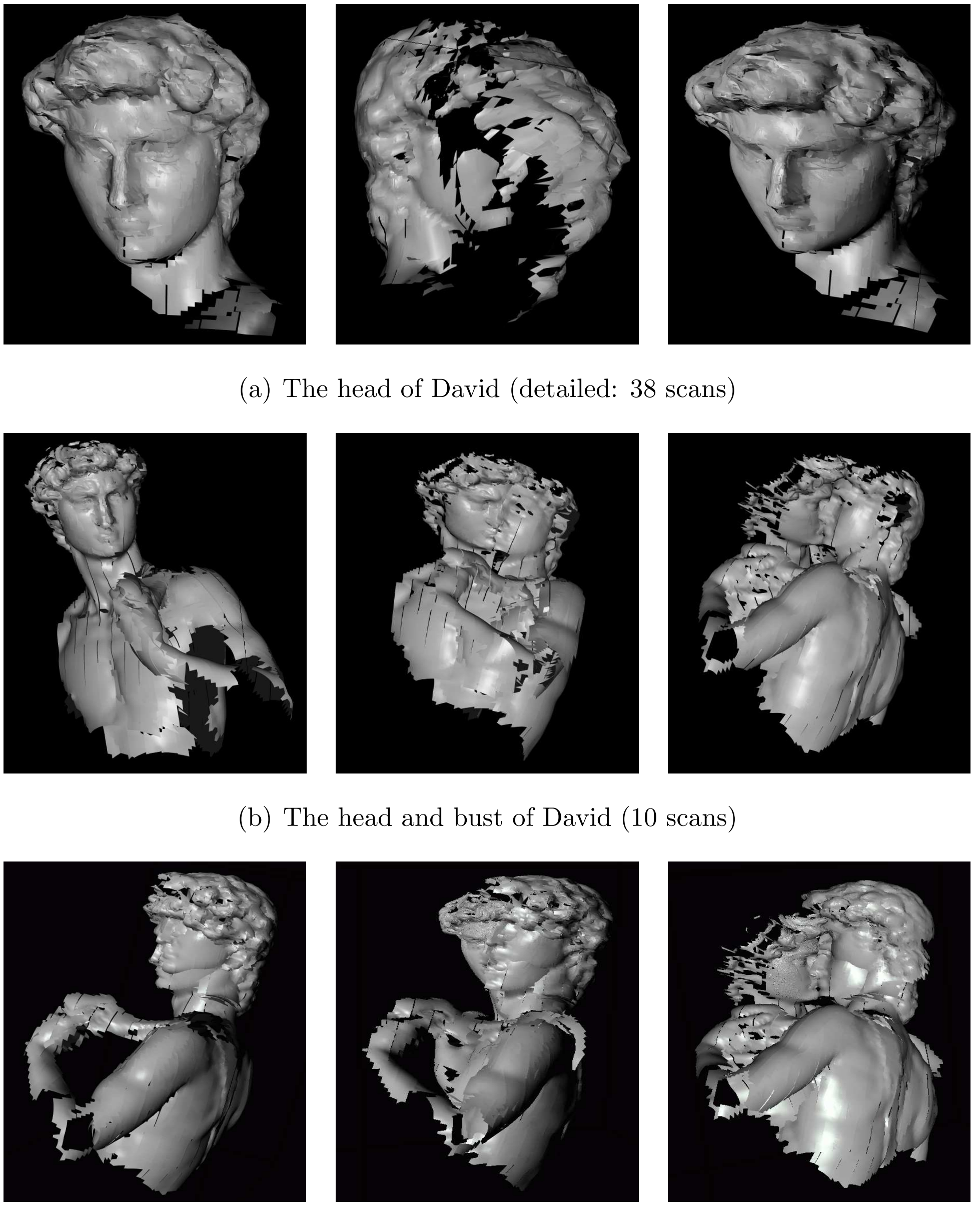
This chapter presented two novel algorithms for simultaneous registration of multiple 3D point sets. The first algorithm is iterative in nature, based on an optimization on manifolds approach while the second is a new analytic method based on singular value decomposition. Main research contributions can be summarized as follows,

- We propose new geometric optimization framework for minimizing smooth function over N-product manifold of special orthogonal group,

- We derived Newton-like algorithms based on the suggested framework. It is is locally quadratically convergent as demonstrated by rigorous mathematical proof and simulation results.
- The algorithm gives better estimates than a recent iterative method proposed by Williams and Bennamoun [84] in the case when correspondence pairs are in chain-like structure and only limited number of match points are present for each pair.
- We put forward a new analytic method that gives exact solutions in the noise free case in a single step and can be used as a good initial estimate for any iterative algorithm.

In this chapter, we have presented a novel algorithm for simultaneous registration of multiple 3D point sets. The algorithm is iterative in nature, based on an optimization-on-a-manifold approach. The algorithm is locally quadratically convergent as demonstrated by rigorous mathematical proof and simulation results. It also converges much faster than prior methods for simultaneous registration. We also propose a new analytic method that provides a closed form solution based on singular value decomposition. It gives exact solutions in the noise free case and can be used as a good initial estimate for any iterative algorithm.

Acknowledgements We acknowledge Prof. Marc Levoy and the Digital Michelangelo Project at Stanford University for providing access to the raw scan data used in this chapter.



(a) The head of David (detailed: 38 scans)

(b) The head and bust of David (10 scans)

(c) Head and bust: Bad initial alignment (21 scans)

Figure 5.5: This figure shows the results of three algorithms for simultaneous registration of multiple 3D point sets - our Optimization-on-a-Manifold method **MBR**, Williams and Bennamoun's SVD-based method **MAT**, and Benjemaa and Schmitt's Quaternion-based method **QUAT** (from left to right) on different instances of the David model. In all cases, the registration produced by our algorithm is quite plausible. The other methods do not fare so well; a typical problem is that the two halves of David's face do not register properly, creating the false effect of two heads.

Chapter 6

Conclusions

This research has been motivated by the fundamental role of pose estimation in many computer vision tasks and the benefit of exploiting the geometric structure of optimization problems. At the beginning of this report, we introduced the subject of computer vision and geometric optimization, reviewed its development, highlighted the research issues and presented research goals together with a strategy as to how the problems can be tackled. This was followed by four technical chapters, mainly focused on the development of proposed parameterization-based framework and its implementation in recovering pose information from images. In this final chapter, we revisit the parameterization-based framework proposed by others in the literature as to put our approach in perspective. We then wrap up the thesis with a summary of research contributions and some remarks for future research.

6.1 Parameterization-based Framework

To the best of our knowledge, there are only two pieces of research based on the parameterization approach we adopt prior to ours, as discussed below.

Taylor and Kriegman [82]: We came across this unpublished technical report during the writing of the thesis. Taylor and Kriegman proposed minimization techniques based on successive parameterization of Lie groups such as the rotation group, the group of rigid body transformation, the sphere and set of infinite straight lines to avoid the singularities problem arising in global parameterization. Unlike our work which present alternative projections such as exponential map, Cayley and orthogonal projection, the report adopted only the exponential map. The report also claimed local quadratic convergence of the proposed algorithms but provide neither mathematical proof nor simulation results. Unlike our algorithms which incorporate one dimensional search strategy to globalize Newton's method, the report focused only on pure Newton method.

Manton [62]: This piece of work was brought to our attention by the author during a workshop presentation. The parameterization-based framework proposed in this paper focused on addressing problems with unitary constraints. While we propose same mapping for both pull back and push forward, [62] introduced the concept of varying parameterization, i.e, the pulled back and push forward can be done by various types of parameterization. The varying parameterization might be attractive from computational viewpoint. It would be interesting to see if the varying parameterization framework does improve computation efficiency of the algorithms proposed in this thesis.

Since our work, Hüper and Trumpf [41] provided a proof of local quadratic convergence of Newton-like method for optimizing a smooth function over a smooth manifold based on global analysis and exemplified their approach to the Rayleigh quotient on the Grassmann manifold.

6.2 Conclusions

Bridging the gap between the abstract mathematical concept of geometric optimization and its practical applications, as well as emphasizing the importance of iterative computer vision algorithms which exploit the geometry of the underlying parameter space have been the objectives of this thesis. This section summarizes the research contribution by re-evaluating their relevance with respect to the research goals.

New Theory for Geometric Optimization

At first glance our technical approach appears little different from what can be achieved using earlier theory, particularly the Riemannian approach. However, it is not at all clear that there is local quadratic convergence of the proposed algorithm unless one sets up the appropriate mathematical objects and mappings. This thesis, along with other independent work, opens up the possibility to develop even more general classes of algorithms evolving on manifolds with different mathematical objects and mappings which achieve the same guaranteed convergence results. This works illustrate convincingly that there is a simpler way of generalizing Euclidean techniques to manifolds, which is more accessible to readers with background in engineering and computer science. Furthermore, it is simpler to implement, computationally more efficient and thus make its suitable for real time applications.

New Computer Vision Algorithms

Again at first glance numerical algorithms proposed for the four pose estimation tasks investigated in the report, appear very little different from what has been around a long time or in recent refinements. However, the differences lead to improved local convergence properties. By developing numerical algorithms directly on the constraint set, we end up with algorithms that explicitly preserve the constraint set at very iteration. Also, by exploiting the geometry of the underlying parameter space such as the curvature of the manifold, faster convergence rate can be achieved. The proposed geometric algorithms are conceptually simpler and intuitive, allow the designer to keep track of what is happening in the algorithm rather than simply using standard optimization packages and solve the problem in the dark ('blackbox' concept). Moreover, the door is opened to develop related advanced algorithms in other contexts. Although it is unreasonable to claim that our algorithms will be the definitive ones in the field, which is what we have sought, it is reasonable to challenge the next generation of researchers to come up with algorithms that take less computational effort to achieve the same accuracy and preserve the constraints at each iteration.

Our current results suggest that there is a fruitful interplay between addressing practical challenges in the area of computer vision and advancing mathematical results in more abstract settings. On the theoretical side, this thesis has shown how to fully exploit the concept of local coordinate charts in differential geometry for solving nonlinear constrained optimization problems in the area of computer vision. As it happens other researchers have independently shown similar results in other application domains. On the practical side, we have introduced the geometric optimization as new tool to design new computer vision algorithms that requires both accuracy and real time computation.

6.3 Future Work

We are convinced that there will be active research in this areas in the next 10 years. This is based on the observation of increasing number of workshops and special sessions in major conferences in diverse areas such as computer vision, pattern recognition, signal processing, machine learning, neural network and so on. We hope this thesis have or will motivate further research in this area, not just from mathematics community, but also from engineering and computer science.

It will be interesting to investigate the effectiveness of geometric optimization framework versus its traditional counterpart, transform method in solving equality constrained optimization problems. We hope this framework can be further extended to inequality constraint sets, instead of the surface of a sphere, we have a ball. There are a few open problems that could be pursued after this thesis, in areas related and unrelated to computer vision. We now discussed a further computer vision task as an example.

Estimation of Fundamental Matrix

This problem is parallel to the estimation of the essential matrix discussed in Chapter 3. Here, we will focus on estimating the fundamental matrix from a set of point correspondences between a pair of uncalibrated images. Both the essential matrix and fundamental matrix are 3 × 3 singular matrices. The essential matrix encapsulates the epipolar geometry between two calibrated images whereas the fundamental matrix algebraically represents the epipolar geometry between two uncalibrated images. Unlike the essential matrix which is characterized by an algebraic constraint, the only constraint on fundamental matrix is that it is of rank 2.

One can apply similar techniques as in this thesis to estimate fundamental matrix although it is not a straightforward generalization from essential matrix case. Algebraically, epipolar geometry relates two uncalibrated images is represented by a fundamental matrix F, as

$$m_2^\top F m_1 = 0,$$

Here, image points m_1 and m_2 are described in pixel image coordinates. The *fundamental manifold* is defined as

$$\mathcal{F} := \{F \in \mathbb{R}^{3\times 3} |\, \mathrm{rank}(F) = 2\} \tag{6.1}$$

A key issue here will be the particular factorization of the fundamental matrix which will be the one that leads to the most useful and general results. There are several equivalent product representations of rank two matrices known. Both of them are related to the fact that $\mathcal{F}$ is a homogeneous space. The first standard result characterizes fundamental matrices in terms of the well known singular value decomposition, as

$$\mathcal{F} = \left\{ U \begin{bmatrix} 1 & 0 & 0 \\ 0 & \sigma & 0 \\ 0 & 0 & 0 \end{bmatrix} V^\top \,\middle|\, U, V \in \mathcal{O}_3, 0 \le \sigma \le 1 \right\}. \tag{6.2}$$

Thus $\mathcal{F}$ is seen as the smooth manifold of (3×3)-matrices with singular values $\{1, \sigma, 0\}$, see [32] for details on the geometry of such manifolds.

A second description of fundamental matrices is as follows, any fundamental matrix F has a factorization

$$F = XY,$$

where X, Y are full rank (3×2) and (2×3) matrices, respectively.

Finally, we mention a third factorization, that is derived from a generalization of the classical Bruhat factorization from Lie groups to singular matrices. For details see the preprint [Manthey and Helmke]. A generic fundamental matrix F has a unique factorization as

$$F = XY,$$

where X, Y are full rank (3×2) and (2×3) matrices, respectively, of the forms

$$X = \begin{bmatrix} x_1 & 0 \\ x_2 & x_3 \\ x_4 & x_5 \end{bmatrix}, \quad Y = \begin{bmatrix} 1 & y_1 & y_2 \\ 0 & 1 & y_3 \end{bmatrix}, \tag{6.3}$$

where x_1, x_5 are nonzero entry and the rest of the element of the X and Y matrix can be arbitrary entry.

Consider now the tangent spaces to $\mathcal{F}$ using the fundamental matrix characterization in (6.2).

Proposition 6.3.1 *The tangent space $T_F\mathcal{F}$ at the fundamental matrix*

$$F = U \begin{bmatrix} 1 & 0 & 0 \\ 0 & \sigma & 0 \\ 0 & 0 & 0 \end{bmatrix} V^\top$$

provided $\sigma \neq 1$ is

$$\begin{aligned} T_F\mathcal{F} &= \left\{ \Omega F - F\Psi + U \begin{bmatrix} 0 & 0 & 0 \\ 0 & \delta & 0 \\ 0 & 0 & 0 \end{bmatrix} V^\top \;\middle|\; \Omega, \Psi \in \mathfrak{so}_3, \delta \in \mathbb{R} \right\} \\ &= U \begin{bmatrix} 0 & -\omega_3\sigma + \psi_3 & -\psi_2 \\ \omega_3 - \psi_3\sigma & \delta & \psi_1\sigma \\ -\omega_2 & \omega_1\sigma & 0 \end{bmatrix} V^\top, \\ &= U \begin{bmatrix} 0 & a & b \\ c & 0 & d \\ e & f & 0 \end{bmatrix} V^\top. \end{aligned} \tag{6.4}$$

If $\sigma = 1$, then $c = -a$ in the above description, and equality with the tangent space is no longer true.

Proof 6.3.1 *For the first equality see [32]. The second is a straightforward computation.*

The above result shows that by using the SVD description we run into a problem. In fact, the SVD parametrization is not submersive at equal singular values and therefore does not yield the full tangent space. Thus this would force us to distinguish in each step of the subsequent analysis between the cases of distinct and equal singular values. It makes sense to work with the third parametrization. We could approach this problem using the technique we propose in this report.

Appendix A

Appendix

A.1 Proof of Theorem 2.2.1

Fixed Point of the Algorithm Let $R_* \in SO_3$ be a fixed point of $s = \pi_3 \circ \pi_2 \circ \pi_1$, we have

$$s(R_*) = R_* \quad \Leftrightarrow \quad \pi_3 \circ \pi_2 \circ \pi_1(R_*) = R_* \quad \Leftrightarrow \quad R_* e^{\Omega(\theta_{\text{opt}} \omega_{\text{opt}})} = R_* \quad \Leftrightarrow \quad e^{\Omega(\theta_{\text{opt}} \omega_{\text{opt}})} = I.$$

If the exponential map in the projection step π_2 is within its injectivity radius, then only $\omega_{\text{opt}} = 0$ satisfies the above equation (since θ_{opt} is a positive scalar). Thus, the only fixed points of the algorithm are critical points. However, as noted in the text only local minima are stable fixed points. Indeed with our geodesic search feature, only global minimum is a stable fixed point. Notice that $\omega_{\text{opt}} = 0$ is the (unique) minimum of $j_0^{(2)}(f \circ \mu_{R_*})(\omega)$ if and only if $\mu_{R_*}(0) = R_*$ is a non-degenerate minimum of $f : SO_3 \to \mathbb{R}$.

Smoothness Properties of the Algorithm Under the assumption that the cost function f is smooth and Hessian of $f \circ \mu_R$ is invertible everywhere, the optimization step π_1 is smooth. The projection step π_2 which involves only the exponential mappings is also smooth. Although the operation π_3 is designed to be discontinuous to escape local minima not a global minimum, yet in a sufficiently small neighbourhood of R_*, the operation π_3 is continuous since all critical points other than R_* have a higher cost than the current cost. It is then more than twice differentiable since it concerns the isolated minimal cost solution of a polynomial equation.

Local Quadratic Convergence of the Algorithm Let R_* denote a stable fixed point of $s = \pi_3 \circ \pi_2 \circ \pi_1$, being also the unique and non-degenerate global minimum of the function f, as already established under our assumptions. We will compute the first derivative of s at this fixed point. Applying the chain rule to the algorithmic mapping $s = \pi_3 \circ \pi_2 \circ \pi_1$, and using the fact that $\pi_1(R_*) = \pi_2(R_*) = \pi_3(R_*) = R_*$, for all elements $\xi \in \mathrm{T}_{R_*}SO_3$, the first derivative of s at fixed point R_* is,

$$\mathrm{D}s(R_*) \cdot \xi = \mathrm{D}\pi_3(R_*) \cdot \mathrm{D}\pi_2(R_*) \cdot \mathrm{D}\pi_1(R_*) \cdot \xi. \tag{A.1}$$

Considering s in the local parameter space, we have the self map

$$\mu_{R_*}^{-1} \circ s \circ \mu_{R_*} : \mathbb{R}^3 \to \mathbb{R}^3. \tag{A.2}$$

Thus, rewriting (A.1) in terms of local parameterization defined by

$$\mu_{R_*} : \mathbb{R}^3 \to SO_3, \quad \omega \mapsto R_* e^{\Omega(\omega)}, \tag{A.3}$$

with Ω as in (2.13), we have

$$\begin{aligned} & \mathrm{D}\mu_{R_*}^{-1} \circ s \circ \mu_{R_*}(0) \cdot h \\ = \;& \mathrm{D}\mu_{R_*}^{-1}(R_*) \cdot \mathrm{D}s(R_*) \cdot \mathrm{D}\mu_{R_*}(0) \cdot h, \\ = \;& \mathrm{D}\mu_{R_*}^{-1}(R_*) \cdot \mathrm{D}\pi_3(R_*) \cdot \mathrm{D}\pi_2 \circ \pi_1 \circ \mu_{R_*}(0) \cdot h. \end{aligned} \tag{A.4}$$

Consider the composite function

$$\pi_2 \circ \pi_1 \circ \mu_{R_*} : \mathbb{R}^3 \to SO_3, \quad \omega \mapsto \mu_{R_*}(\omega) e^{\Omega\left(\omega_{\mathrm{opt}}^{\mathrm{Newton}}(\mu_{R_*}(\omega))\right)}. \tag{A.5}$$

where

$$\omega_{\mathrm{opt}}^{\mathrm{Newton}} \circ \mu_{R_*}(\omega) = \omega_{\mathrm{opt}}^{\mathrm{Newton}}(\mu_{R_*}(\omega)) = -[\mathrm{H}_{f\circ\mu_{R_*}}(\omega)]^{-1} \nabla_{f\circ\mu_{R_*}}(\omega). \tag{A.6}$$

Exploiting linearity of the mapping Ω, using the well known formula for differentiating the matrix exponential and the fact that

$$\mu_{R_*}(0) = R_*, \quad \omega_{\mathrm{opt}}^{\mathrm{Newton}}(R_*) = 0,$$

we have

$$\mathrm{D}\mu_{R_*}(0) \cdot h = R_* \Omega(h), \tag{A.7}$$

and

$$\begin{aligned} & \mathrm{D}\omega_{\text{opt}}^{\text{Newton}} \circ \mu_{R_*}(0) \cdot h && \text{(A.8)} \\ = \;& -[\mathrm{H}_{f \circ \mu_{R_*}}(0)]^{-1} \mathrm{D}\nabla_{f \circ \mu_{R_*}}(0) \cdot h - \mathrm{D}[(\mathrm{H}_{f \circ \mu_{R_*}}(0)]^{-1} \cdot h \nabla_{f \circ \mu_{R_*}}(0), \\ = \;& -[\mathrm{H}_{f \circ \mu_{R_*}}(0)]^{-1} \mathrm{H}_{f \circ \mu_{R_*}}(0) h, \quad \text{since } \nabla_{f \circ \mu_{R_*}}(0) = 0 \\ = \;& -h. \end{aligned}$$

Now, we compute the first derivative of the composite function (A.5) in the limit as ω approaches zero as,

$$\begin{aligned} \mathrm{D}\pi_2 \circ \pi_1 \circ \mu_{R_*}(0) \cdot h &= R_* \Omega(h) + R_* \Omega(\mathrm{D}\omega_{\text{Newton}} \circ \mu_{R_*}(0) \cdot h) && \text{(A.9)} \\ &= 0. \end{aligned}$$

Substituting (A.9) into (A.4) shows that for all $h \in \mathbb{R}^3$

$$\mathrm{D}\mu_{R_*}^{-1} \circ s \circ \mu_{R_*}(0) \cdot h = 0. \tag{A.10}$$

Since the iterate R_k is in an open neighbourhood of R_*, then by inverse mapping, $\omega_k = \mu_{R_*}^{-1}(R_k)$ stays in a sufficiently small open neighborhood of the origin in $\mathbb{R}^3$. Vanishing of the first derivative then implies local quadratic convergence by the Taylor-type argument, for some positive κ

$$\left\| \mu_{R_*}^{-1} \circ s \circ \mu_{R_*}(\omega_k) \right\| \leq \sup_{y \in \overline{\mathcal{N}(0)}} \kappa \left\| \mathrm{D}^2 \mu_{R_*}^{-1} \circ s \circ \mu_{R_*}(y) \right\| \cdot \|\omega_k\|^2 \tag{A.11}$$

with $\overline{\mathcal{N}(0)}$ the topological closure of a sufficiently small open neighbourhood of origin in $\mathbb{R}^3$.

Remark A.1.1 *The result holds for π_3 being an identity operation as for a Newton step, or is given by a smooth geodesic search. Notice that if the geodesic search is switched off in the neighbourhood of R_*, so that π_3 is an identity operator, then the assumption of uniqueness of R_* can be relaxed for this proof to still hold.*

A.2 Rank of $\mathcal{D}$.

Consider $\mathcal{D}$ as defined in (2.11). In order to evaluate rank($\mathcal{D}$), observe that $\mathcal{D}$ can be reformulated as

$$\mathcal{D} = \hat{U}(M^\top \otimes I),$$

where $M := \left[\begin{array}{cccc} m_1 & m_2 & \cdots & m_n \end{array} \right]$, and denoting $\mathbf{1}_n$ as $n \times 1$ vector consists of 1,

$$\begin{array}{rcl} \hat{U} & := & \mathrm{diag}(\tilde{U}_1, \cdots, \tilde{U}_n)(I - (\mathbf{1}_n \otimes \bar{A})), \;\; \tilde{U}_i := I - U_i, \\ \bar{A} & := & (\mathcal{A}^\top \mathcal{A})^{-1} \mathcal{A}^\top \left[\begin{array}{ccc} \tilde{U}_1 & \cdots & \tilde{U}_n \end{array} \right]. \end{array}$$

Then,

$$\mathrm{rank}(\mathcal{D}) \leq \min\{\mathrm{rank}(\hat{U}), \mathrm{rank}(M^\top \otimes I)\}$$

To analyze the minimum rank of $\hat{U}$, observe that

$$\hat{U}X = 0, \tag{A.12}$$

where

$$X := \left[\begin{array}{ccccc} I & \tilde{U}_1 & I & \cdots & I \\ \vdots & I & \tilde{U}_2 & & \vdots \\ & & & \ddots & \\ I & \cdots & & & \tilde{U}_n \end{array} \right].$$

Elementary row and column operations on X gives that rank(X) $= (n + 3)$ in the generic case when $\{\mathbf{u}_i\}_{\{i=1,\ldots,n\}}$ are linearly independent. Since $\hat{U}$ has full rank of $3n$ and it can loose at most $(n+3)$ rank, then rank($\hat{U}$) $= 3n - (n+3)$. The necessary conditions for such existence is that $3n - (n+3) \geq 0$ or $n \geq 2$. For generic models when rank(M) = 3, then

$$\begin{array}{rcl} \mathrm{rank}(\mathcal{D}) & \leq & \min\{(3n - (n+3)), (3 \times \mathrm{rank}(M))\} \\ & \leq & \min\{(2n-3), 9\}. \end{array}$$

A.3 Proof of Theorem 3.2.2

Fixed Points of the Algorithm

If the projection step is either π_2^{cay} or π_2^{svd}, but not $\pi_2^{\mu_E}$, then the only fixed points of the corresponding algorithm $s = \pi_2 \circ \pi_1$ are those elements of $\mathcal{E}$ which are the critical points of the objective function $f : \mathcal{E} \to \mathbb{R}$. Indeed, the only stable fixed points are the local minima of f.

To justify this, notice that $E_* = U_* E_0 V_*^\top \in \mathcal{E}$ is a fixed point of s if it is already a fixed point of π_1, because π_2 following π_1 is a projection. Let $x_{\text{opt}}(E_*) := \begin{bmatrix} x_1 & \cdots & x_5 \end{bmatrix}^\top$, a fixed point of π_1 in turn is characterised by

$$\begin{aligned}
&\pi_1(U_* E_0 V_*^\top) = U_* E_0 V_*^\top, \\
\iff\ & U_*\big(E_0 + \Omega_1(x_{\text{opt}}(E_*))E_0 - E_0 \Omega_2(x_{\text{opt}}(E_*))\big)V_*^\top = U_* E_0 V_*^\top, \\
\iff\ & \Omega_1(x_{\text{opt}}(E_*))E_0 - E_0\Omega_2(x_{\text{opt}}(E_*)) = 0, \\
\iff\ & \begin{bmatrix} 0 & -\sqrt{2}x_3 & -x_5 \\ \sqrt{2}x_3 & 0 & x_4 \\ -x_2 & x_1 & 0 \end{bmatrix} = 0, \\
\iff\ & x_{\text{opt}}(E_*) = 0.
\end{aligned}$$

Notice that $x_{\text{opt}}(E_*) = 0$ happens to be the (unique) minimum of $j_0^{(2)}(f \circ \mu_{E_*})(x)$ if and only if $\mu_{E_*}(0) = E_*$ is a non-degenerate local minimum of $f : \mathcal{E} \to \mathbb{R}$.

It remains to show that $s(E) = E$ together with $\pi_1(E) \neq E$ is not possible irrespective of the type of projection π_2 we use in our algorithm, i.e., π_2^{svd} or π_2^{cay}. Let $E = U E_0 V^\top$. Assume $x \in \mathbb{R}^5 \setminus \{0\}$, i.e., we assume

$$\pi_1(E) = U(E_0 + \Omega_1(x)E_0 - E_0\Omega_2(x))V^\top \neq E.$$

(i) π_2^{svd}**:** There exist $H, K \in \mathcal{SO}_3$ such that

$$\pi_1(E) = H \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} K^\top, \quad \sigma_1 \geq \sigma_2 \geq \sigma_3. \tag{A.13}$$

The projection $\pi_2 = \pi_2^{\mathrm{svd}}$ maps then $\pi_1(E)$ to $HE_0K^\top$, and is well defined in the generic case $\sigma_2 > \sigma_3$, as

$$\pi_2(\pi_1(E)) = HE_0K^\top. \tag{A.14}$$

Now suppose that $\pi_2 = \pi_2^{\mathrm{svd}}$ maps $\pi_1(E)$ back to E, i.e.,

$$HE_0K^\top = UE_0V^\top \tag{A.15}$$

holds. It is easily seen that Eq. (A.15) forces the special orthogonal matrices $H^\top U$ and $K^\top V$ to be of the following form

$$H^\top U = K^\top V = \begin{bmatrix} \varepsilon \cos t & \varepsilon \sin t & 0 \\ -\sin t & \cos t & 0 \\ 0 & 0 & \varepsilon \end{bmatrix}, \ \varepsilon \in \{\pm 1\}, t \in \mathbb{R}.$$

Using (A.13), $H^\top U$ and $K^\top V$ of this form already diagonalise $U^\top \cdot \pi_1(E) \cdot V$, i.e.,

$$\begin{aligned} &H^\top U\big(E_0 + \Omega_1(x)E_0 - E_0\Omega_2(x)\big)V^\top K \\ = \ &\frac{1}{\sqrt{2}} H^\top U \begin{bmatrix} 1 & -\sqrt{2}x_3 & -x_5 \\ \sqrt{2}x_3 & 1 & x_4 \\ -x_2 & x_1 & 0 \end{bmatrix} V^\top K = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix}. \end{aligned}$$

Multiplying out and equating terms implies $x = 0$, a contradiction.

(ii) π_2^{cay}: We will show that

$$E_0 = \mathrm{cay}(\Omega_1(x))E_0 \,\mathrm{cay}(-\Omega_2(x)) \tag{A.16}$$

implies $x = 0$, a contradiction. Rewriting (A.16) gives

$$\left(I - \frac{1}{2}\Omega_1(x)\right) E_0 \left(I + \frac{1}{2}\Omega_2(x)\right) = \left(I + \frac{1}{2}\Omega_1(x)\right) E_0 \left(I - \frac{1}{2}\Omega_2(x)\right)$$

$$\Longleftrightarrow$$

$$\Omega_1(x)E_0 = E_0\Omega_2(x)$$

$$\Longleftrightarrow$$

$$x = 0.$$

Unfortunately, the situation is more involved if the projection we use is $\pi_2^{\mu_E}$. Consider the following example. For arbitrary $U, V \in SO_3$ let $E = UE_0V^\top$ and suppose

$$x_{\text{opt}} = \sqrt{2}\begin{bmatrix}0 & \pi & 0 & 0 & \pi\end{bmatrix}^\top, \tag{A.17}$$

therefore

$$\Omega_1(x_{\text{opt}}) = \Omega_2(x_{\text{opt}}) = \begin{bmatrix}0 & 0 & \pi\\ 0 & 0 & 0\\ -\pi & 0 & 0\end{bmatrix} \quad \text{and} \quad \mathrm{e}^{\Omega_1(x_{\text{opt}})} = \mathrm{e}^{\Omega_2(x_{\text{opt}})} = \begin{bmatrix}-1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & -1\end{bmatrix}. \tag{A.18}$$

Then

$$\pi_1(E) = U\begin{bmatrix}1 & 0 & -\pi\\ 0 & 1 & 0\\ -\pi & 0 & 0\end{bmatrix}V^\top \neq E, \tag{A.19}$$

but

$$\pi_2^{\mu_E}(\pi_1(E)) = UE_0V^\top = E. \tag{A.20}$$

One can easily find other examples, but the reason that one can construct such examples is that the injectivity radius of the exponential map $\exp : \mathfrak{so}_3 \to SO_3$ is finite, namely equal to π.

Smoothness Properties of the Algorithm

The optimization step π_1 is smooth under the assumption of the Hessian of $f \circ \mu_E$ being everywhere invertible. Indeed, under this assumption the linear system to be solved in each optimization step has a unique solution. For the projection step π_2, both $\pi_2^{\mu_E}$ and π_2^{cay} are smooth mappings. For detailed proofs on the smoothness of the mapping π_2^{svd}, see [34].

Local Quadratic Convergence of the Algorithm

Let E_* denote a stable fixed point of $s = \pi_2 \circ \pi_1$, being also a non-degenerate minimum of the function f. Also, $\pi_1(E_*) = E_*$, $\pi_2(E_*) = E_*$ and let $\xi \in T_{E_*}\mathcal{E}$. By chain rule, we have

$$\mathrm{D}s(E_*)\cdot\xi = \mathrm{D}\pi_2(E_*)\cdot\mathrm{D}\pi_1(E_*)\cdot\xi. \tag{A.21}$$

Now, re-express s in local parameter space give

$$\mu_{E_*}^{-1}\circ s\circ\mu_{E_*} : \mathbb{R}^5 \to \mathbb{R}^5.$$

Since $E_* = \mu_{E_*}(0)$, rewriting (A.21) in terms of local parameterization gives

$$\begin{aligned}\mathrm{D}\mu_{E_*}^{-1}\circ s\circ\mu_{E_*}(0)\cdot h &= \mathrm{D}\mu_{E_*}^{-1}(E_*)\cdot\mathrm{D}s(E_*)\cdot\mathrm{D}\mu_{E_*}(0)\cdot h,\\ &= \mathrm{D}\mu_{E_*}^{-1}(E_*)\cdot\mathrm{D}\pi_2(E_*)\cdot\mathrm{D}\pi_1\circ\mu_{E_*}(0)\cdot h.\end{aligned}$$

where

$$x_{\text{opt}}^{\text{Newton}}\circ\mu_{E_*}(x) = x_{\text{opt}}^{\text{Newton}}(\mu_{E_*}(x)) = -[\mathrm{H}_{f\circ\mu_{E_*}}(x)]^{-1}\nabla_{f\circ\mu_{E_*}}(x) \tag{A.22}$$

Exploiting linearity of the mapping $\Omega_1,\ \Omega_2$, using the well known formula for differentiating the matrix exponential and the fact that

$$\mathrm{D}\mu_{E_*}(0)\cdot h = U_*[\Omega_1(h)E_0 - E_0\Omega_2(h)]V_*^\top,\quad x_{\text{opt}}^{\text{Newton}}(\mu_{E_*}(0)) = 0,$$

and consider the derivative of $x_{\text{opt}}^{\text{Newton}}\circ\mu_{E_*}$ at $0\in\mathbb{R}^5$,

$$\begin{aligned}&\mathrm{D}x_{\text{opt}}^{\text{Newton}}\circ\mu_{E_*}(0)\cdot h\\ =\ & -[\mathrm{H}_{f\circ\mu_{E_*}}(0)]^{-1}\mathrm{D}\nabla_{f\circ\mu_{E_*}}(0)\cdot h - \mathrm{D}[(\mathrm{H}_{f\circ\mu_{E_*}}(0)]^{-1}\cdot h\nabla_{f\circ\mu_{E_*}}(0),\\ =\ & -[\mathrm{H}_{f\circ\mu_{E_*}}(0)]^{-1}\mathrm{H}_{f\circ\mu_{E_*}}(0)h,\quad \text{since } \nabla_{f\circ\mu_{E_*}}(0) = 0\\ =\ & -h. \end{aligned}\tag{A.23}$$

Consider three composite functions and its first derivative at the fixed point $E_* = U_*E_0V_*^\top$. Each function is relevant when different projection step is used,

- Orthogonal projection, π_{svd}

$$\begin{aligned}\pi_1\circ\mu_{E_*} : \mathbb{R}^5 &\to T^{\text{aff}}\mathcal{E},\\ x &\mapsto \mu_{E_*}(x) + U_*e^{\Omega_1(x)}\Omega_1\left(x_{\text{opt}}^{\text{Newton}}(\mu_{E_*}(x))\right)E_0e^{\Omega_2(x)^\top}V_*^\top\\ &\quad -U_*e^{\Omega_1(x)}E_0\Omega_2\left(x_{\text{opt}}^{\text{Newton}}(\mu_{E_*}(x))\right)e^{\Omega_2(x)^\top}V_*^\top,\end{aligned}$$

and derivative of this composite function at fixed point E_* is

$$\begin{aligned}\mathrm{D}\pi_1 \circ \mu_{E_*}(0) \cdot h &= \mathrm{D}\mu_{E_*}(0) \cdot h + U_* \Omega_1 \left(\mathrm{D}x_{\text{opt}}^{\text{Newton}} \circ \mu_{E_*}(0) \cdot h\right) E_0 V_*^\top \\ &\quad - U_* E_0 \Omega_2 \left(\mathrm{D}x_{\text{opt}}^{\text{Newton}} \circ \mu_{E_*}(0) \cdot h\right)^\top V_*^\top, \\ &= 0.\end{aligned}$$

- Exponential map, π_{μ_E}

$$\begin{aligned}&\pi_{\mu_{E_*}} \circ \pi_1 \circ \mu_{E_*} : \mathbb{R}^5 \to \mathcal{E}, \\ &x \mapsto U_* e^{\Omega_1(x)} e^{\Omega_1\left(x_{\text{opt}}^{\text{Newton}}(\mu_{E_*}(x))\right)} E_0 e^{\Omega_2\left(x_{\text{opt}}^{\text{Newton}}(\mu_{E_*}(x))\right)^\top} e^{\Omega_2(x)^\top} V_*^\top,\end{aligned}$$

and first derivative of this composite function at E_* is

$$\begin{aligned}&\mathrm{D}\pi_{\mu_{E_*}} \circ \pi_1 \circ \mu_{E_*}(0) \cdot h \qquad \text{(A.24)} \\ = \; & U_* \Omega_1(h) E_0 V_*^\top + U_* \Omega_1 \left(\mathrm{D}x_{\text{opt}}^{\text{Newton}} \circ \mu_{E_*}(0) \cdot h\right) E_0 V_*^\top + \\ & U_* E_0 \Omega_2 \left(\mathrm{D}x_{\text{opt}}^{\text{Newton}} \circ \mu_{E_*}(0) \cdot h\right)^\top V_*^\top + U_* E_0 \Omega_2(h)^\top V_*^\top, \\ = \; & 0.\end{aligned}$$

- Cayley-like projection, π_{cay}

$$\pi_{\text{cay}} \circ \pi_1 \circ \mu_{E_*} : \mathbb{R}^5 \to \mathcal{E},$$

$$x \mapsto U_* e^{\Omega_1(x)} \text{cay}\left(\Omega_1(x_{\text{opt}}^{\text{Newton}}(\mu_{E_*}(x)))\right) E_0 \text{cay}\left(-\Omega_2(x_{\text{opt}}^{\text{Newton}}(\mu_{E_*}(x)))\right) e^{\Omega_2(x)^\top} V_*^\top$$

and its derivative at E_* is

$$\mathrm{D}\pi_{\text{cay}} \circ \pi_1 \circ \mu_{E_*}(0) \cdot h = \mathrm{D}\pi_{\mu_{E_*}} \circ \pi_1 \circ \mu_{E_*}(0) \cdot h = 0. \qquad \text{(A.25)}$$

Substituting (A.24), (A.24) or (A.25) into (A.22) shows that for all $h \in \mathbb{R}^5$

$$\mathrm{D}\mu_{E_*}^{-1} \circ s \circ \mu_{E_*}(0) \cdot h = 0 \qquad \text{(A.26)}$$

irrespective of whether π_2 is an orthogonal projection, a Cayley-like projection or an exponential map on $\mathcal{E}$. Let E_k denote a point in an open neighbourhood of E_*, generated by s. By inverse mapping, $x_k = \mu_{E_*}^{-1}(E_k)$ stays in a sufficiently small open neighbourhood of the origin in $\mathbb{R}^5$. Vanishing of the first derivative then implies local quadratic convergence by the Taylor-type argument, for some positive κ,

$$\left\|\mu_{E_*}^{-1} \circ s \circ \mu_{E_*}(x_k)\right\| \leq \sup_{y \in \overline{\mathcal{N}(0)}} \kappa \left\|\mathrm{D}^2 \mu_{E_*}^{-1} \circ s \circ \mu_{E_*}(y)\right\| \cdot \|x_k\|^2, \tag{A.27}$$

with $\overline{\mathcal{N}(0)}$ the topological closure of a sufficiently small open neighborhood of 0 in $\mathbb{R}^5$.

A.4 Proof of Theorem 4.2.1

Fixed Point of the Algorithm Let $T_* \in SE_3$ be a fixed point of $s = \pi_2 \circ \pi_1$, we have

$$s(T_*) = T_* \quad \Leftrightarrow \quad \pi_2 \circ \pi_1(T_*) = T_* \quad \Leftrightarrow \quad T_* e^{\zeta(\lambda_{\text{opt}} x_{\text{opt}})} = T_* \quad \Leftrightarrow \quad e^{\zeta(\lambda_{\text{opt}} x_{\text{opt}})} = I.$$

If the exponential map in the projection step π_2 is within its injectivity radius, then only $x_{\text{opt}} = 0$ satisfies the above equation (since λ_{opt} is a positive scalar). Thus all critical points are fixed points, although only local minima are stable fixed points. Notice that $x_{\text{opt}} = 0$ is the (unique) minimum of $j_0^{(2)}(f \circ \mu_{T_*})(x)$ if and only if $\mu_{T_*}(0) = T_*$ is a non-degenerate local minimum of $f : SE_3 \to \mathbb{R}$.

Smoothness Properties of the Algorithm Under the assumption that the cost function f is smooth and Hessian of $f \circ \mu_T$ is invertible everywhere, the optimization step π_1 is smooth. The projection step π_2 which involves only the exponential mappings is also smooth.

Local Quadratic Convergence of the Algorithm Let T_* denote a stable fixed point of $s = \pi_2 \circ \pi_1$, being also a non-degenerate minimum of the function f. We will compute the first derivative of s at this fixed point. Applying chain rule and using the fact that $\pi_1(T_*) = \pi_2(T_*) = T_*$, we have for all tangent elements $\xi \in \mathrm{T}_{T_*} SE_3$,

$$\mathrm{D}s(T_*) \cdot \xi = \mathrm{D}\pi_2(T_*) \cdot \mathrm{D}\pi_1(T_*) \cdot \xi. \tag{A.28}$$

Consider s in the local parameter space, we have the self map

$$\mu_{T_*}^{-1} \circ s \circ \mu_{T_*} : \mathbb{R}^6 \to \mathbb{R}^6. \tag{A.29}$$

Thus, rewriting (A.29) in terms of local parameterization defined by

$$\mu_{T_*} : \mathbb{R}^6 \to SE_3, \quad x \mapsto T_* e^{\zeta(x)} \tag{A.30}$$

with $\mu_{T_*}(0) = T_*$ and ζ defined in (4.4), we have for all $h \in \mathbb{R}^6$,

$$\mathrm{D}\mu_{T_*}^{-1} \circ s \circ \mu_{T_*}(0) \cdot h = \mathrm{D}\mu_{T_*}^{-1}(T_*) \cdot \mathrm{D}s \circ \mu_{T_*}(0) \cdot h. \tag{A.31}$$

Next, consider the composite function,

$$s \circ \mu_{T_*} : \mathbb{R}^6 \to SE_3, \quad x \mapsto \mu_{T_*}(x) e^{\zeta(x_{\text{opt}}^{\text{Newton}}(\mu_{T_*}(x)))}, \tag{A.32}$$

since in the neighbourhood of T_*, the algorithm uses the Newton step with $\lambda_{\text{opt}} = 1$ and

$$x_{\text{opt}}^{\text{Newton}} \circ \mu_{T_*}(x) = x_{\text{opt}}^{\text{Newton}}(\mu_{T_*}(x)) = -[\mathrm{H}_{f\circ\mu_{T_*}}(x)]^{-1} \nabla_{f\circ\mu_{T_*}}(x). \tag{A.33}$$

Exploiting linearity of the mapping ζ, using the well known formula for differentiating the matrix exponential and the fact that,

$$\mu_{T_*}(0) = T_*, \quad x_{\text{opt}}^{\text{Newton}} \circ \mu_{T_*}(0) = 0,$$

we have

$$\mathrm{D}\mu_{T_*}(0) \cdot h = T_* \zeta(h), \tag{A.34}$$

and

$$\begin{aligned} & \mathrm{D}x_{\text{opt}}^{\text{Newton}} \circ \mu_{T_*}(0) \cdot h \\ = \quad & -[\mathrm{H}_{f\circ\mu_{T_*}}(0)]^{-1} \mathrm{D}\nabla_{f\circ\mu_{T_*}}(0) \cdot h - \mathrm{D}[\mathrm{H}_{f\circ\mu_{T_*}}]^{-1}(0) \cdot h \nabla_{f\circ\mu_{T_*}}(0), \\ = \quad & -[\mathrm{H}_{f\circ\mu_{T_*}}(0)]^{-1} \mathrm{H}_{f\circ\mu_{T_*}}(0) h, \quad \text{since } \nabla_{f\circ\mu_{T_*}}(0) = 0 \\ = \quad & -h. \end{aligned} \tag{A.35}$$

Now, using (A.34) and (A.35), we compute the first derivative of the composite function in (A.32),

$$\mathrm{D}s \circ \mu_{T_*}(0) \cdot h = T_* \zeta(h) + T_* \zeta\left(\mathrm{D}x_{\text{opt}}^{\text{Newton}} \circ \mu_{T_*}(0) \cdot h\right) = 0. \tag{A.36}$$

Substituting (A.36) into (A.31) shows that for all $h \in \mathbb{R}^6$

$$\mathrm{D}\mu_{T_*}^{-1} \circ s \circ \mu_{T_*}(0) \cdot h = 0 \tag{A.37}$$

Let T_k denote a point in an open neighbourhood of T_*, generated by s. By inverse mapping, $x_k = \mu_{T_*}^{-1}(T_k)$ stays in a sufficiently small open neighbourhood of the origin in $\mathbb{R}^6$. Vanishing of the first derivative then implies local quadratic convergence by the Taylor-type argument, for some positive κ,

$$\left\| \mu_{T_*}^{-1} \circ s \circ \mu_{T_*}(x_k) \right\| \leq \sup_{y \in \overline{\mathcal{N}(0)}} \kappa \left\| \mathrm{D}^2 \mu_{T_*}^{-1} \circ s \circ \mu_{T_*}(y) \right\| \cdot \|x_k\|^2 , \tag{A.38}$$

with $\overline{\mathcal{N}(0)}$ the topological closure of a sufficiently small open neighbourhood of origin in $\mathbb{R}^6$.

A.5 Proof of Theorem 5.3.1

The proof of this theorem follows exactly that of Theorem 4.2.1, except several replacements: constraint manifold $SE_3 \to SO_3^N$, local parameter space $\mathbb{R}^6 \to \mathbb{R}^{3N}$, fixed point $T_* \to \mathcal{R}_*$, mapping $\mu_{T_*} \to \varphi_{\mathcal{R}_*}$, element $\zeta \to \widetilde{\Omega}$, and parameters $x \to \omega$.

Bibliography

[1] P. A. Absil, “Trust Region Methods on Riemannian Manifolds with Applications in Numerical Linear Algebra”, *Proceedings of the 16th International Symposium on Mathematical Theory of Networks and Systems*, Leuven, Belgium, 2004.

[2] P. A. Absil, R. Mahony, R. Sepulchre, “Riemannian Geometry of Grassmann Manifolds with a View on Algorithmic Computation”, *Acta Applicandae Mathematicae*, vol. 80, no. 2, pp. 199–220, 2004.

[3] R. L. Adler, J. -P. Dedieu, “Newton’s Method on Riemannian Manifolds and a Geometric Model for the Human Spine”, *IMA Journal of Numerical Analysis*, vol. 22, pp. 359–390, 2002.

[4] A. Ansar, and K. Daniilidis, “Linear Pose Estimation from Points or Lines”, *Proceedings of the International Conference on Computer Animation*, Switzerland, June, 2002.

[5] K. Arun, T. S. Hwang, and S. Bolstein, “Least Squares Fitting of Two 3D Point Sets”, *IEEE Transactions on Pattern Analysis and Machine Intelligent*, vol. 9, no. 5, pp. 698–700, 1987.

[6] M. Baeg, H. Hashimoto, F. Harashima, and J. B. Moore, “Pose Estimation of Quadratic Surface of Quadratic Surface Using Surface Fitting Technique”, *Proceedings of the International Conference on Intelligent Robots and Systems*, pp. 204–209, 1995.

[7] R. Benjemaa, and F. Schmitt, “A Solution for the Registration of Multiple 3D

Point Sets Using Unit Quaternion", *Proceedings of the European Conference on Computer Vision*, Freiburg, Germany, pp. 34–50, 1998.

[8] R. Bergevin, M. Soucy, H. Gagnon, and D. Laurendeau, "Towards a General Multiview Registration Technique", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 5, pp. 540–547, 1996.

[9] P. Besl, and N. D. McKay, "A Method for Registration of 3D Shapes", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

[10] G. L. Bilbro, and W. E. Synder, "Linear Estimation of Object Pose from Local Fits to Segments", *Proceedings of the IEEE International Conference of Robotics and Automation*, pp. 1747–1752, 1987.

[11] R. W. Brockett, "Differential Geometry and the Design of Gradient Algorithms", *Proceedings of the Symposium on Pure Mathematics*, Ed. R. Green and S. T. Yau, pp. 69–92, Providence, RI, 1993.

[12] M. Buss, L. Faybusovich, and J. B. Moore, "Dikin-Type algorithms for Dextrous Grasping Force Optimization", *International Journal on Robotics Research*, vol. 17, no. 8, pp. 831-839, 1998.

[13] Y. Chen, and G. Medioni, "Object Modelling by Registration of Multiple Range Images", *Image Vision Computing*, vol. 10, no. 3, pp. 145–155, 1992.

[14] C. Chen, Y. Hung, and J. Cheng, "A Fast Automatic Method for Registration of Partially Overlapping Range Images", *Proceedings of International Conference on Computer Vision*, 1998.

[15] C. Chen, Y. Hung, and J. Cheng, "RANSAC-based Darces: A New Approach to Fast Automatic Registration of Partially Overlapping Range Images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 11, pp. 1229–1234, 1999.

[16] Y. Choquet-Bruhat, C. DeWitt-Morette, M. Dillard-Bleick, *Analysis, Manifolds and Physics, Part I*, Elsevier Science Publishers B. V. (North Holland), Amsterdam, 1991.

[17] S. Cunnington, and A. J. Stoddart, "N-View Point Set Registration: A Comparison", *Proceedings of the British Machine Vision Conference*, Nottingham, U. K., 1999.

[18] W. Davidon, "Variable Metric Method for Minimization", *Technical Report ANL–5990*, Argonne National Laboratory, 1959. See also *SIAM Journal of Optimization*, vol. 1, pp. 1–17, 1991.

[19] J.-P. Dedieu, and P. Priouret, "Newton's Method on Riemannian Manifolds: Covariant Alpha Theory", *International Journal of Computer Vision*, vol. 15, no.1–2, pp. 124–141, 1995.

[20] A. Edelman, T. A. Arias, and S. T. Smith, "The Geometry of Algorithms with Orthogonality Constraints", *SIAM Journal on Matrix Analysis and Applications*, vol. 20, no. 2, pp. 303–353, 1998.

[21] D. Eggert, A. W. Fitzgibbon, and R. Fisher, "Simultaneous Registration of Multiple Range Views for Use in Reverse Engineering of CAD Models", *Computer Vision and Image Understanding*, vol. 69, no. 3, pp. 253–272, 1998.

[22] O. D. Faugeras, and M. Herbert, "The Representation, Recognition, and Location of 3D objects", *International Journal of Robotics Research*, vol. 5, no. 3, pp. 27–52, 1986.

[23] M. A. Fischler, and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography", *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[24] R. Fletcher, and C. M. Reeves, "Functions Minimization by Conjugate Gradients", *Computer Journal*, vol. 7, pp. 149–154, 1964.

[25] C. Dorai, J. Weng, and A. K. Jain, "Optimal Registration of Object Views Using Range Data", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 19, no. 10, pp. 1131–1138, 1997.

[26] D. Gabay, "Minimizing a Differentiable Function over a Differentiable Manifold", *Journal of Optimization Theory and Applications*, vol. 37, no. 2, pp. 177–219, 1982.

[27] V. M. Govindu, "Lie-Algebraic Averaging for Globally Consistent Motion Estimation", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[28] K. Gunnarsson, and F. Prinz, "CAD Model Based Recognition and Localization of Parts in Manufacturing", *Computer*, vol. 20, no. 8, pp. 67–74, 1987.

[29] Y. Han, W. E. Synder, and G. L. Bilbro, "Pose Determination using Tree Annealing", *Proceedings of the IEEE International Conference of Robotics and Automation*, pp. 427–432, 1990.

[30] R. M. Haralick, H. Joo, C. Lee, X. Zhuang, V. G. Vaidya, and M. B. Kim, "Pose Estimation from Corresponding Point Data", *IEEE Transactions on Systems, Man and Cybernetics*, vol. 19, no. 6, 1426–1446, 1989.

[31] R. Hartley, and A. Zisserman, *Multiple View Geometry*, Cambridge University Press, 2000.

[32] U. Helmke, and J. B. Moore, *Optimization and Dynamical Systems*, CCES, Springer, London, 1994.

[33] U. Helmke, K. Hüper, and J. B. Moore, "Quadratically Convergent Algorithms for Optimal Dextrous Hand Grasping", *IEEE Transactions on Robotics and Automation*, vol. 18, no. 2, 2002.

[34] U. Helmke, K. Hüper, P. Y. Lee, and J. B. Moore, "Essential Matrix Estimation via Newton-type Methods", *Proceedings of the 16th International Symposium on Mathematical Theory of Networks and Systems*, Belgium, Leuven, 2004.

[35] M. R. Hestenes, and E. Stiefel, "Methods of Conjugate Gradients for Solving Linear Systems", *Journal of Research of the National Bureau of Standards*, vol. 49, pp. 409–436, 1952.

[36] B. K. P. Horn, "Closed Form Solution of Absolute Orientation using Unit Quaternions", *Journal of the Optical Society of America A*, vol. 4, pp. 629–642, 1987.

[37] B. K. P. Horn, "Relative orientation", *International Journal of Computer Vision*, vol. 4, no. 1, pp. 59–78, 1990.

[38] T. S. Huang, and O. D. Faugeras, "Some Properties of the E matrix in Two-View Motion Estimation", *IEEE Transactions on Pattern Analysis and Machine Inteligence*, vol. 11, no. 12, pp. 1310–1312, 1989.

[39] T. S. Huang, and A. N. Netravali, "Motion and Structure from Feature Correspondence: A Review", *Proceedings of the IEEE*, vol. 82, no. 2, pp. 252–268, 1994.

[40] P. J. Huber, *Robust Statistics*, John Wiley & Sons, Inc., 1981.

[41] K. Hüper, and J. Trumpf, "Newton-like Methods for Numerical Optimization on Manifolds", *Proceedings of the 38th Asilomar Conference on Signals, Systems and Computers*, California, United States, 2004.

[42] D. C. Jiang, J. B. Moore, and H. B. Ji, "Self-Concordant Functions for Optimization on Smooth Manifolds", *Proceedings of the 43rd IEEE Conference on Decision and Control*, Bahamas, pp. 3631–3636, 2004.

[43] A. Johnson, and M. Hebert, "Surface Registration by Matching Oriented Points", *Proceedings of the International Conference on 3D Digital Imaging and Modeling*, 1997.

[44] K. Kanatani, "Analysis of 3D Rotation Fitting", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 5, pp. 543–549, 1994.

[45] S. Krishnan, P. Y. Lee, J. B. Moore, and S. Venkatasubramania, “Global Registration of Multiple 3D Point Sets via Optimization-on-a-Manifold”, *Proceedings of Eurographics Symposium on Geometry Processing*, pp. 187–196, 2005.

[46] S. Lang, *Fundamentals of Differential Geometry*, Springer, New York, 1999.

[47] J. M. Lee, *Introduction to Smooth Manifolds*, Graduate Texts in Mathematics, Springer, New York, 2003.

[48] P. Y. Lee, and J. B. Moore, “Pose Estimation via Gauss-Newton-on-manifold Approach”, *Proceedings of the 16th International Symposium on Mathematical Theory of Networks and Systems*, Leuven, Belgium, 2004.

[49] P. Y. Lee, and J. B. Moore, “3D Localization of Quadratic Surfaces”, *Proceedings of the 38th Asilomar Conference on Signals, Systems and Computers*, California, United States, 2004.

[50] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade and D. Fulk, “The Digital Michelangelo Project: 3D Scanning of Large Statues”, *Proceedings of SIGGRAPH*, pp. 131–144, 2000.

[51] H. C. Longuet-Higgins, “A Computer Algorithm for Reconstructing a Scene from Two Projections”, *Nature*, vol. 293, pp. 133135, September 1981.

[52] A. Lorusso, D. W. Eggert, and R. B. Fisher, “A Comparison of Four Algorithms for Estimating 3D Rigid Transformations”, *Proceedings of the British Machine Vision Conference*, Birmingham, England, pp. 237–246, 1995.

[53] D. G. Lowe, “Three-Dimensional Object Recognition from Single Two-Dimensional Images”, *Artificial Intelligence*, vol. 31, no. 3, pp. 355–395, 1987.

[54] D. G. Lowe, “Fitting Parametrized Three-Dimensional Models to Images”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 5, pp. 441–450, 1991.

[55] C. -P. Lu, G. D. Hager, and E. Mjolsness, “Fast and Globally Convergent Pose Estimation from Video Images”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 6, pp. 610–622, 2000.

[56] D. G. Luenberger, “The Gradient Projection Method along Geodesics”, *Management Science*, vol. 18, pp. 620–631, 1972.

[57] D. G. Luenberger, *Introduction to Linear and Nonlinear Programming*, Addison-Wesley Publishing Company, Reading, 1973.

[58] Y. Ma, J. Košecká, and S. Sastry, “Optimization Criteria and Geometric Algorithms for Motion and Structure Estimation”, *International Journal of Computer Vision*, vol. 44, no. 3, pp. 219–249, 2001.

[59] R. Mahony, *Optimization Algorithms on Homogeneous Spaces*, PhD thesis, Australian National University, Canberra, March 1994.

[60] R. Mahony, “The Constrained Newton Method on a Lie Group and the Symmetric Eigenvalue Problem”, *Linear Algebra and Its Applications*, pp. 248–267, 1996.

[61] R. Mahony, and J. H. Manton, “The Geometry of the Newton Method on Non-Compact Lie Groups”, *Journal of Global Optimization*, 309–327, vol. 23, 2002.

[62] J. H. Manton, “Optimization Algorithms Exploiting Unitary Constraints”, *IEEE Transactions on Signal Processing*, vol. 50, no. 3, pp. 635–650, March 2002.

[63] J. H. Manton, “On the Various Generalisations of Optimisation Algorithms to Manifolds”, *Proceedings of the 16th International Symposium on Mathematical Theory of Networks and Systems*, Leuven, Belgium, 2004.

[64] T. Masuda, and N. Yokoya, “A Robust Method for Registration and Segmentation of Multiple Range Images”, *Computer Vision and Image Understanding*, vol. 61, no. 3, pp. 295–307, 1995.

[65] J. B. Moore, M. Baeg, and U. Helmke, "Gradient Flow Techniques for Pose Estimation of Quadratic Surfaces", *Proceedings of the World Congress in Computational Methods and Applied Mathematics*, Atlanta, Georgia, pp. 360–363, 1994.

[66] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, CRC Press, 1994.

[67] Y. Nesterov, *Introductory Lectures on Convex Optimization - A Basic Course*, Kluwer Academic Publishers, 2004.

[68] J. Nocedal, and S. J. Wright, *Numerical Optimization*, Springer Series in Operation Research, Springer Verlag, New York, 1999.

[69] B. Owren, and B. Welfert, "The Newton Iterations on Lie Groups", *BIT*, vol. 40, no. 1, pp. 121–145, 2000.

[70] X. Pennec, "Multiple Registration and Mean Rigid Shapes: Application to the 3D Case", *Proceedings of 16th Leeds Annual Statistical Workshop*, Leeds, U. K., pp 178–185, 1996.

[71] H. Pottmann, Q. X. Huang, Y. L. Yang, and S. M. Hu, "Geometry and Convergence Analysis of Algorithms for Registration of 3D Shapes", *Technical Report*, no. 117, Technical University Wien, 2004.

[72] K. Pulli, "Multiview Registration for Large Data Sets", *Proceedings of the International Conference on 3D Digital Imaging and Modeling*, Ottawa, Canada, pp. 160–168, 1999.

[73] L. Quan, and Z. Lan, "Linear N-Point Camera Pose Determination", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 8, pp. 774–780, 1999.

[74] S. Rusinkiewicz, and M. Levoy, "Efficient Variant of the ICP Algorithm", *Proceedings of the International Conference on 3D Digital Imaging and Modeling*, 2001.

[75] G. C. Sharp, S. W. Lee, and D. K. Wehe, "Multiview Registration of 3D Scenes by Minimizing Error between Coordinate Frames", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 1037–1050, 2004.

[76] M. Shub, "Some Remarks on Dynamical Systems and Numerical Analysis", *Dynamical Systems and Partial Differential Equations, Proceedings of VII ELAM*, Caracas: Equinoccio, Universidad Simon Bolivar, pp. 69–92, 1986.

[77] L. Silva, O. R. P. Bellon, and K. L. Boyer, "Enhanced, Robust Genetic Algorithms for Multiview Range Image Registration", *Proceedings of the International Conference on 3D Digital Imaging and Modeling*, Banff, Canada, pp. 268–275, 2003.

[78] S. T. Smith, *Geometric Optimization Methods for Adaptive Filtering*, PhD Thesis, Harvard University, Cambridge Massachusetts, 1993.

[79] Special Session on Applications of Differential Geometry to Signal Processing, *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Philadelphia, United States, 2005

[80] F. Stein, and G. Medioni, "Structural Indexing: Efficient 3D Object Recognition", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 125–145, 1992.

[81] A. Stoddart, and A. Hilton, "Registration of Multiple Point Sets", *Proceedings of the International Conference on Pattern Recognition*, Vienna, pp. 40–44, 1996.

[82] C. J. Taylor, and D. J. Kriegman, "Minimization on the Lie Group $SO(3)$ and Related Manifolds", *Yale University Technical Report no. 9405*, April 1994.

[83] C. Udriste, *Convex Functions and Optimization Methods on Riemannian Manifolds*, Kluwer Academic Publishers, 1994.

[84] J. Williams, and M. Bennamoun, "Simultaneous Registration of Multiple Corresponding Point Sets", *Computer Vision and Image Understanding*, vol. 81, no. 1, pp. 117–142, 2001.

[85] Y. Yang, "Optimization on Riemannian Manifold", *Proceedings of the 38th Conference on Decision and Control*, pp. 888–893, 1999.

[86] N. Yokoya, and M. D. Levine, "Range Image Segmentation Based on Differential Geometry: A Hybrid Approach", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 6, pp. 643–649, 1989.