OPTIMAL MOTION FROM IMAGE SEQUENCES: A RIEMANNIAN VIEWPOINT

by

Yi Ma, Jana Košecká, and Shankar Sastry

Memorandum No. UCB/ERL M98/37

22 June 1998

OPTIMAL MOTION FROM IMAGE SEQUENCES: A RIEMANNIAN VIEWPOINT

by

Yi Ma, Jana Košecká, and Shankar Sastry

Memorandum No. UCB/ERL M98/37

22 June 1998

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Optimal Motion From Image Sequences: A Riemannian Viewpoint *

Yi Ma   Jana Košecká   Shankar Sastry

Electronics Research Laboratory
University of California at Berkeley
Berkeley, CA 94720-1774
{mayi, janka, sastry}@robotics.eecs.berkeley.edu

June 22, 1998

Abstract

Motion recovery from image correspondences is typically a problem of optimizing an objective function associated with the epipolar (or Longuet-Higgins) constraint. This objective function is defined on the so called essential manifold, which has a nice intrinsic Riemannian structure. Based on existing optimization techniques on Riemannian manifolds, in particular on Stiefel or Grassmann manifolds, we propose a Riemannian Newton algorithm to solve the motion recovery problem, making use of the natural geometric structure of the essential manifold. The same ideas also apply to conjugate gradient algorithms. The proposed geometric algorithms have quadratic rates of convergence.

Key words: essential manifold, motion recovery, Newton's method, conjugate gradient method, Grassmann manifold, Stiefel manifold.

1 Introduction

The problem of recovering structure and motion from a sequence of images has been one of the central problems in computer vision over the past decade and has been studied extensively from various perspectives. The proposed techniques varied in the type of features they used, types of assumptions they made about the environment, projection models and the type of algorithms. Based on measurements the techniques can be viewed either as discrete: using point, line features, or differential: using measurements of optical flow.

The seminal work of Longuett-Higgins [9] on characterization of so called *epipolar constraint*, enabled decoupling of the structure and motion problems and led to the development of numerous linear and nonlinear algorithms for motion estimation (see [12, 4, 7, 23] for overviews). The epipolar constraint has been formulated both in discrete and differential setting and the recent work of the

*Computer vision research report. This work is supported by ARO under the MURI grant DAAH04-96-1-0341. The authors also would like to thank Dr. Steven T. Smith (MIT Lincoln Laboratory) for invaluable discussion on this subject and thank professor Alan Weinstein (Berkeley Mathematics Department) for proofreading the manuscript.

authors [10, 11] demonstrated the possibility of the parallel development of linear algorithms for both cases: that of point feature measurements and optical flow. The original 8-point algorithm proposed by Longuet-Higgins is easily generalizable for uncalibrated camera case, where the epipolar constraint is captured by fundamental matrix. Detailed analysis of linear and nonlinear techniques for estimation of fundamental matrix, exploring the use of different objective functions can be found in [13].

While the (analytic) geometrical aspects of the problem have been understood the proposed solutions to the problem have been shown very sensitive to noise and often failed in practical applications. These experiences motivated further studies which focused on the use of statistical analysis of existing techniques and understanding of various assumptions which affected the performance of existing algorithms. These studies had been done both in analytical [2, 16] and experimental setting [20].

The appeal of linear algorithms which use the epipolar constraint, (in discrete case [23, 7, 9, 12] and in differential case [6, 10, 19] is the closed form solution to the problem which in the absence of noise provides true estimate of the motion. Further analysis of linear techniques reveals inherent bias in the translation estimate [6]. Attempts were made to compensate for the bias improving slightly the performance of the linear techniques [7]. By eliminating the nonlinear effects caused by perspective projection [21] proposed a linear algorithm for both the shape and motion recovery assuming orthographic projection.

The performance of the numerical optimization techniques which use nonlinear objective function has been shown superior to the linear ones. The objective functions used are either a version of essential constraint or a 2-norm of image measurements. These techniques either require iterative numerical optimization [23, 15] or use Monte-Carlo simulations [6] to sample the space of one of the unknown parameters. Extensive experiments revealed problems with convergence when initialized far away from the true solution [21]. They nonlinear algorithms are often initialized using the linear ones and merely refine the initial estimates [22].

Different objective function has been proposed by Horn [5], where instead of minimizing directly the essential (coplanar) constraint the objective function expresses the true errors in relative orientation (i.e. translation and rotation). Horn proposed an iterative procedure where the update of the estimate takes into account the orthonormal constraint of the unknown rotation. Horn's algorithm and the algorithm proposed by [18] are some of the few which consider explicitly the differential geometric properties of $SO(3)$. However they do not make a connection and full exploitation of the differential geometric properties of the entire space of essential matrices as characterized in our recent papers [10, 11].

More recent studies [15] clarified the source of some of the difficulties (e.g. rotation and translation ambiguity) from the point of view of noise and explored the source and presence of local minima of the objective function. Soatto's chooses to minimize an objective function which is equivalent to essential constraint and proposed an iterative Bilinear Projection Algorithm (BPA) for obtaining the optimal solution. The algorithm is in an essence Newton-Gauss iterative scheme for updating in each step one of the unknowns.

Even from the efforts which rightly concentrated on the noise related aspects of the problem (i.e. trying different nonlinear objective functions etc.) none of the existing algorithms truly took advantage of the underlying geometrical structure of the problem. The underlying search space was usually parameterized for computational convenience [5, 18, 12] instead of being consistent to its intrinsic geometric structure. Consequently, in these algorithms, solving for optimal updating direction typically involved using Lagrangian multipliers to deal with the constraints on the search space; and "walking" on such space was done approximately by the *update-then-project* procedure.

Due to recent developments in optimization techniques on Riemannian manifolds [14, 3], in this paper we will give a top level mathematical view for the nonlinear optimization problem associated with the motion recovery and, using Newton's method as an example, show how to apply the optimization theory on Riemannian manifolds to solve this problem by using the intrinsic Riemannian structure of the underlying search space. In this paper, only the discrete case will be worked out in detail while the generalization to the differential case is also discussed.

This paper relies on familiarity with Riemannian geometry. Section 2 shows how to generalize optimization schemes on single Riemannian manifold to their product space. Section 3 then studies the intrinsic Riemannian structure of the essential manifold (the space of all essential matrices). Section 4 discusses how to optimize a general objective function on the essential manifold using the (Riemannian) Newton's method. Section 5 works out in details the (Riemannian) Newton algorithm for optimizing the least square objective function associated with the epipolar constraint. Some simulation results are presented in Section 6 and Section 7 discusses how to generalize to the differential case.

2 Optimization on Riemannian Manifold Preparation

Newton and conjugate gradient methods are classical nonlinear optimization techniques to minimize a function $f(x)$, where x belongs to an open subset of Euclidean space $\mathbb{R}^n$. Recent developments in optimization algorithms on Riemannian manifolds have provided geometric insights for generalization of Newton and conjugate gradient methods to certain classes of Riemannian manifolds. Smith [14] gave a detailed treatment of the general theory of optimization on Riemannian manifolds; Edelman, Arias and Smith [3] then studied the case where the Riemannian manifolds are Stiefel and Grassmann manifolds, and presented a uniform geometric framework for Newton and conjugate gradient algorithms on Stiefel and Grassmann manifolds.

These new mathematical results solve the more general optimization problem of minimizing the function $f(x)$, where x is constrained to a Riemannian manifold (M, g) which is, in turn, usually given as a submanifold in a Euclidean Space $\mathbb{R}^n$. Previous algorithms for solving such problems were application dependent: the performance of proposed algorithms relied on particular parameterization chosen for the submanifold M and also depended on certain approximation schemes applied to update the states on the search space. However, the new results of [3] show that, on Stiefel and Grassmann manifolds, one may make use of the canonical Riemannian structure of these manifolds and systematically develop Newton and conjugate gradient methods on them. Since the parameterization and metrics are canonical and the state is updated using geodesics, the performance of so obtained algorithms is guaranteed: they typically have polynomial complexity and super-linear (quadratic) rate of convergence [14].

The purpose of this paper is to apply these new Riemannian optimization schemes to solve the long-existing problem in computer vision: recovering 3D motion from image correspondences. As we will soon see the underlying Riemannian manifold for this vision problem (the so called essential manifold) is a product of Stiefel manifolds instead of a single one. We first need to generalize Edelman *et al*'s methods to the product of Stiefel (or Grassmann) manifolds. This paper relies on familiarity with Edelman *et al*'s work [3] and some background of modern Riemannian geometry (a good reference for Riemannian geometry is Spivak [17] or Kobayashi [8]).

Suppose (M_1, g_1) and (M_2, g_2) are two Riemannian manifolds with Riemannian metrics:

$$g_1(\cdot,\cdot) : TM_1 \times TM_1 \to C^\infty(M_1),$$
$$g_2(\cdot,\cdot) : TM_2 \times TM_2 \to C^\infty(M_2)$$

where TM_1 is the tangent bundle of M_1, similarly for TM_2. The corresponding Levi-Civita connections of these manifolds are denoted as:

$$\nabla_1 : \mathcal{X}(M_1) \times \mathcal{X}(M_1) \to \mathcal{X}(M_1),$$
$$\nabla_2 : \mathcal{X}(M_2) \times \mathcal{X}(M_2) \to \mathcal{X}(M_2)$$

where $\mathcal{X}(M_1)$ stands for the space of smooth vector fields on M_1, similarly for $\mathcal{X}(M_2)$.

Now let M be the product space of M_1 and M_2, *i.e.* $M = M_1 \times M_2$. Let $i_1 : M_1 \to M$ and $i_2 : M_2 \to M$ be the natural inclusions and $\pi_1 : M \to M_1$ and $\pi_2 : M \to M_2$ be the projections. To simplify the notation, we identify TM_1 and TM_2 with $i_{1*}(TM_1)$ and $i_{2*}(TM_2)$ respectively. Then $TM = TM_1 \times TM_2$ and $\mathcal{X}(M) = \mathcal{X}(M_1) \times \mathcal{X}(M_2)$. For any vector field $X \in \mathcal{X}(M)$ we can write X as the composition of its components in the two subspaces TM_1 and TM_2: $X = (X_1, X_2) \in TM_1 \times TM_2$. The canonical Riemannian metric $g(\cdot,\cdot)$ on M is determined as:

$$g(X,Y) = g_1(X_1, Y_1) + g_2(X_2, Y_2), \quad X, Y \in \mathcal{X}(M).$$

Define a ∇ connection on M as:

$$\nabla_X Y = (\nabla_{1X_1} Y_1, \nabla_{2X_2} Y_2) \in \mathcal{X}(M_1) \times \mathcal{X} M_2, \quad X, Y \in \mathcal{X}(M).$$

One can directly check that this connection is torsion free and compatible with the canonical Riemannian metric g on M (*i.e.* preserving the metric) hence it is the Levi-Civita connection for the Riemannian manifold (M, g). From the construction of ∇, it is also canonical.

According to Edelman et al's paper [3], in order to apply Newton and conjugate gradient methods on a Riemannian manifold, one needs to know how to explicitly calculate parallel transport of vectors on the manifolds and also needs the explicit expression of geodesics. The reason that Edelman et al's methods can be easily generalized to any product of Stiefel (or Grassmann) manifolds is because there are simple relations between the parallel transports on the product manifold and the factor manifolds. The following theorem follows directly from the definition of the Levi-Civita connection on the product manifold.

Theorem 1 *Consider $M = M_1 \times M_2$ the product Riemannian manifold of M_1 and M_2. Then for two vector fields $X, Y \in \mathcal{X}(M)$, Y is parallel along X if and only if Y_1 is parallel along X_1 and Y_2 is parallel along X_2.*

As a corollary to this theorem, the geodesics in the product manifold are just the products of geodesics in the two factor manifolds. Consequently, the calculation of parallel transport and geodesics in the product space can be reduced to those for each factor manifold.

3 Riemannian Structure of the Essential Manifold

In this section we study the Riemannian structure of the essential manifold, which plays an important role in the motion recovery from image correspondences (for details see [11]). For a vector

$S = (s_1, s_2, s_3)^T \in \mathbb{R}^3$, the notation $\hat{S}$ means the associated skew-symmetric matrix:

$$\begin{pmatrix} 0 & -s_3 & s_2 \\ s_3 & 0 & -s_1 \\ -s_2 & s_1 & 0 \end{pmatrix}.$$

The *essential manifold* is defined to be:

$$\mathcal{E} = \{R\hat{S} \mid R \in SO(3), \hat{S} \in so(3)\}.$$

$SO(3)$ is the Lie group of 3×3 rotation matrices (special orthogonal matrices with determinant $+1$), and $so(3)$ is the Lie algebra of $SO(3)$, *i.e.* the tangent plane of $SO(3)$ at the identity. $so(3)$ then consists of all 3×3 skew-symmetric matrices. A matrix E with the form $E = R\hat{S}, R \in SO(3), \hat{S} \in so(3)$ is then a point in this manifold. Such a matrix E is called an *essential matrix*. In the motion recovery problem, the objective function is a function on the essential manifold $F(E)$ which is linear with respect to E. Because of this linearity, the problem may be reduced to optimize the function on the *normalized essential manifold*:

$$\mathcal{E}_1 = \{R\hat{S} \mid R \in SO(3), \hat{S} \in so(3), \frac{1}{2}tr(\hat{S}^T\hat{S}) = 1\}.$$

Notice $\frac{1}{2}tr(\hat{S}^T\hat{S}) = S^TS$. In order to understand the Riemannian structure of the normalized essential manifold, we start with the Riemannian structure on the tangent bundle of the Lie group $SO(3)$, *i.e.* $T(SO(3))$.

The tangent space of $SO(3)$ at the identity e is simply its Lie algebra $so(3)$:

$$T_e(SO(3)) = so(3).$$

Since $SO(3)$ is a compact Lie group, it has an intrinsic bi-invariant metric [1] (such metric is unique up to a constant scale). In the matrix form, this metric is given explicitly as:

$$g_0(\hat{S}_1, \hat{S}_2) = \frac{1}{2}tr(\hat{S}_1^T\hat{S}_2), \quad \hat{S}_1, \hat{S}_2 \in so(3).$$

Notice that this metric is induced from the Euclidean metric on $SO(3)$ as a Stiefel submanifold embedded in $\mathbb{R}^{3\times3}$. The tangent space at any other point $R \in SO(3)$ is given by the push-forward map R_*:

$$T_R(SO(3)) = R_*(so(3)) = \{R\hat{S} | \hat{S} \in so(3)\}.$$

Thus the tangent bundle of $SO(3)$ is:

$$T(SO(3)) = \bigcup_{R\in SO(3)} T_R(SO(3))$$

Since the tangent bundle of a Lie group is trivial [17], $T(SO(3))$ is then equivalent to the product $SO(3) \times so(3)$. $T(SO(3))$ can then be expressed as:

$$T(SO(3)) = \{(R, R\hat{S}) \mid R \in SO(3), \hat{S} \in so(3)\} \cong SO(3) \times so(3).$$

If identify the tangent space of $so(3)$ with itself, then the left-invariant metric g_0 of $SO(3)$ induces a canonical metric on the tangent bundle $T(SO(3))$:

$$\tilde{g}(X, Y) = g_0(X_1, X_2) + g_0(Y_1, Y_2), \quad X, Y \in so(3) \times so(3).$$

Note that the metric defined on the fiber $so(3)$ of $T(SO(3))$ is the same as the Euclidean metric if we identify $so(3)$ with $\mathbb{R}^3$. Such induced metric on $T(SO(3))$ is left-invariant under the action of $SO(3)$.

Comments 1 *Averaging the above left-invariant metric on $T(SO(3))$ with respect to all the right action of $SO(3)$, one obtains a natural bi-invariant metric on $T(SO(3))$. However, such bi-invariant metric does not allow a product structure. We therefore avoid using such bi-invariant structure and use the left-invariant one instead, since the product structure has much more computational advantage, as we will soon see.*

Then the metric $\tilde{g}$ on the whole tangent bundle $T(SO(3))$ induces a canonical metric g on the unit tangent bundle of $T(SO(3))$,

$$T_1(SO(3)) \cong \{(R, R\hat{S}) \mid R \in SO(3), \hat{S} \in so(3), \frac{1}{2}tr(\hat{S}^T\hat{S}) = 1\}.$$

It is direct to check that with the identification of $so(3)$ with $\mathbb{R}^3$, the unit tangent bundle is simply the product $SO(3) \times S^2$ where S^2 is the 2-sphere embedded in $\mathbb{R}^3$. According to Edelman et al [3], $SO(3)$ and S^2 both are Stiefel manifolds $V_{n,p}$ with the type $n = p = 3$ and $n = 3, p = 1$, respectively. As Stiefel manifolds, they both possess canonical metrics by regarding them as quotients between orthogonal groups. Here $S^2 = SO(3)/SO(2)$. Fortunately, for Stiefel manifolds with the special type $p = n$ or $p = 1$, the canonical metrics are the same as the Euclidean metrics induced as submanifolds embedded in $\mathbb{R}^{n\times p}$. From the above discussion, we have

Theorem 2 *The unit tangent bundle $T_1(SO(3))$ is equivalent to $SO(3) \times S^2$. Its Riemannian metric g induced from the left-invariant metric on $SO(3)$ is the same as that induced from the Euclidean metric with $T_1(SO(3))$ naturally embedded in $\mathbb{R}^{3\times 4}$. Further, ($T_1(SO(3))$,g) is the product Riemannian manifold of $(SO(3), g_1)$ and (S^2, g_2) with g_1 and g_2 canonical quotient metrics for $SO(3)$ and S^2 as Stiefel manifolds.*

However, the unit tangent bundle $T_1(SO(3))$ is not exactly the normalized essential manifold $\mathcal{E}_1$. Indeed, $T_1(SO(3))$ is a double covering of the normalized essential space $\mathcal{E}_1$, *i.e.* $\mathcal{E}_1 = T_1(SO(3))/\mathbb{Z}^2$ (for details see [11]). The natural covering map from $T_1(SO(3))$ to $\mathcal{E}_1$ is:

$$\begin{aligned} h : T_1(SO(3)) &\rightarrow \mathcal{E}_1 \\ (R, R\hat{S}) \in T_1(SO(3)) &\mapsto R\hat{S} \in \mathcal{E}_1. \end{aligned}$$

The inverse of this map is given by:

$$h^{-1}(R\hat{S}) = \left\{(R, R\hat{S}),\ (R\exp(-\hat{S}\pi), R\hat{S})\right\}.$$

Comments 2 *As we know, the two pairs of rotation and translation corresponding to the same normalized essential matrix $R\hat{S}$ are $(R, \hat{S})$ and $(R\exp(-\hat{S}\pi), \exp(\hat{S}\pi)\hat{S})$. As pointed out by professor Weinstein, this double covering h is equivalent to identify a left-invariant vector field on $SO(3)$ with the one obtained by flowing it along the corresponding geodesic by distance π, the so-called time-π map of the geodesic flow on $SO(3)$.*

If we take for $\mathcal{E}_1$ the Riemannian structure induced from the covering map h, the original optimization problem of optimizing $F(E)$ on $\mathcal{E}_1$ is equivalent to optimizing $F(R, S)$ on $T_1(SO(3))$.

Generalizing Edelman et al's methods to the product Riemannian manifolds, we obtain elegant geometric Newton or conjugate gradient algorithms for solving the later problem. Due to Theorem 2, we can simply choose the induced Euclidean metric on $T_1(SO(3))$ and explicitly give out these intrinsic algorithms in terms of the matrix representation of $T_1(SO(3))$. Since this Euclidean metric is the same as the intrinsic metrics, the apparently extrinsic representation preserves all the intrinsic geometric properties of the given problems. In this sense, the algorithms we are about to develop for the motion recovery are different from other existing algorithms which make use of particular parameterizations of the underlying search manifold $T_1(SO(3))$, such as quaternions [5, 18, 7, 12].

4 Optimization on the Essential Manifold

Let $F(R,S)$ be a function on $T_1(SO(3)) \cong SO(3) \times S^2$ with $R \in SO(3)$ represented by a 3×3 rotation matrix and $S \in S^2$ a unit vector in $\mathbb{R}^3$. This section derives the Newton algorithm for optimizing a function on this manifold (please refer to [3] for the Newton algorithm on general Stiefel or Grassmann manifolds).

Let g_1 and g_2 be the canonical metrics for $SO(3)$ and S^2 respectively and ∇_1 and ∇_2 be the corresponding Levi-Civita connections. Let g and ∇ be the induced Riemannian metric and connection on the product manifold $T_1(SO(3))$. The gradient of a $F(R,S)$ on $T_1(SO(3))$ is a vector field $G = \text{grad } F$ on $T_1(SO(3))$ such that:

$$dF(Y) = g(G,Y), \quad \text{for all vector fields } Y \text{ on } T_1(SO(3)).$$

This gradient is explicitly given as:

$$G = (F_R - RF_R^T R,\ F_S - SF_S^T S) \in T_R(SO(3)) \times T_S(S^2)$$

where $F_R \in \mathbb{R}^{3\times 3}$ is the matrix of partial derivatives of F with respect to the elements of R and $F_S \in \mathbb{R}^3$ is the vector of partial derivatives of F with respect to the elements of S, *i.e.* ,

$$(F_R)_{ij} = \frac{\partial F}{\partial R_{ij}}, \quad (F_S)_k = \frac{\partial F}{\partial S_k}, \quad 1 \le i,j,k \le 3.$$

For any $X = (X_1, X_2), Y = (Y_1, Y_2) \in T(SO(3)) \times T(S^2)$, the Hessian of $F(R,S)$ is given by:

$$\begin{aligned}\text{Hess } F(X,Y) &= F_{RR}(X_1,Y_1) - trF_R^T\Gamma_R(X_1,Y_1)\\ &+ F_{SS}(X_2,Y_2) - trF_S^T\Gamma_S(X_2,Y_2)\\ &+ F_{RS}(X_1,Y_2) + F_{SR}(Y_1,X_2).\end{aligned}$$

where the Christoffel functions Γ_R for $SO(3)$ and Γ_S for S^2 are:

$$\begin{aligned}\Gamma_R(X_1,Y_1) &= \frac{1}{2}R(X_1^TY_1 + Y_1^TX_1),\\ \Gamma_S(X_2,Y_2) &= \frac{1}{2}S(X_2^TY_2 + Y_2^TX_2)\end{aligned}$$

and the other terms are:

$$\begin{aligned}F_{RR}(X_1,Y_1) &= \sum_{ij,kl} \frac{\partial^2 F}{\partial R_{ij}\partial R_{kl}}(X_1)_{ij}(Y_1)_{kl}, \quad F_{SS}(X_2,Y_2) = \sum_{i,j} \frac{\partial^2 F}{\partial S_i \partial S_j}(X_2)_i(Y_2)_j,\\ F_{RS}(X_1,Y_2) &= \sum_{ij,k} \frac{\partial^2 F}{\partial R_{ij}\partial S_k}(X_1)_{ij}(Y_2)_k, \quad F_{SR}(Y_1,X_2) = \sum_{i,jk} \frac{\partial^2 F}{\partial S_i \partial R_{jk}}(Y_1)_i(X_2)_{jk}\end{aligned}$$

For Newton's method, we need to find the *optimal updating* tangent vector Δ such that:

$$\text{Hess } F(\Delta, Y) = g(-G, Y) \quad \text{for all tangent vectors } Y.$$

In order to solve for Δ, first find the tangent vector $Z(\Delta) = (Z_1, Z_2) \in T_R(SO(3)) \times T_S(S^2)$ (in terms of Δ) satisfying the equations:

$$\begin{aligned} F_{RR}(\Delta_1, Y_1) + F_{SR}(Y_1, \Delta_2) &= g_1(Z_1, Y_1) \quad \text{for all tangent vectors } Y_1 \in T(SO(3)) \\ F_{SS}(\Delta_2, Y_2) + F_{RS}(\Delta_1, Y_2) &= g_2(Z_2, Y_2) \quad \text{for all tangent vectors } Y_2 \in T(S^2) \end{aligned}$$

From the expression of the gradient G, the vector $\Delta = (\Delta_1, \Delta_2)$ then satisfies the linear equations:

$$\begin{aligned} &Z_1 - R\,\text{skew}(F_R^T \Delta_1) - \text{skew}(\Delta_1 F_R^T)R = -(F_R - RF_R^T R) \\ &Z_2 - F_S^T S \Delta_2 = -(F_S - SF_S^T S) \end{aligned}$$

with $R^T\Delta_1$ skew-symmetric and $S^T\Delta_2 = 0$. In the above expression, the notation skew(A) means the skew-symmetric part of the matrix A: skew$(A) = (A - A^T)/2$. For this system of linear equations to be solvable, the Hessian has to be non-degenerate, in other words the corresponding Hessian matrix in local coordinates is invertible. The non-degeneracy depends on the chosen objective function.

According to Newton's algorithm, knowing Δ, the search state is then updated from (R, S) in direction Δ along geodesics to $(\exp(R, \Delta_1), \exp(S, \Delta_2))$, where $\exp(R, \cdot)$ stands for the exponential map from $T_R(SO(3))$ to $SO(3)$ at point R, similarly for $\exp(S, \cdot)$. For explicit expressions for the geodesics $\exp(R, \Delta_1 t)$ on $SO(3)$ and $\exp(S, \Delta_2 t)$ on S^2 see [3]. The overall algorithm can be summarized in the following:

The Newton Algorithm for Minimizing $F(R, S)$ on the Essential Manifold:

- At the point (R, S),
  - Compute the gradient $G = (F_R - RF_R^T R,\ F_S - SF_S^T S)$,
  - Compute $\Delta = -\ \text{Hess}^{-1} G$.
- Move (R, S) in the direction Δ along the geodesic to $(\exp(R, \Delta_1), \exp(S, \Delta_2))$.
- Repeat if $\|G\| \geq \epsilon$ for pre-determined $\epsilon > 0$.

Since the manifold $SO(3) \times S^2$ is compact, this algorithm is guaranteed to converge to a (local) extremum of the objective function $F(R, S)$. Note that this algorithm works for any objective function on $SO(3) \times S^2$. For an objective function with non-degenerate Hessian, the Newton algorithm has quadratic (super-linear) rate of convergence [14]. In the next section, we will apply this algorithm to a particular objective function which is widely used in the computer vision literature to recover relative motion from image correspondences.

5 Optimal Motion Recovery

From computer vision, we know that two corresponding image points $p, q \in \mathbb{R}^3$ satisfy the so called epipolar (or Longuet-Higgins) constraint:

$$p^T R \hat{S} q = 0$$

where $R \in SO(3)$ and $S \in S^2$ are relative rotation and translation between the two image frames.[1] Thus to recover the motion R, S from a given set of image correspondences $p_i, q_i \in \mathbb{R}^3, i = 1, \ldots, N$, it is standard to optimize the following objective function:

$$F(R,S) = \sum_{i=1}^{N} (p_i^T R \hat{S} q_i)^2, \quad p_i, q_i \in \mathbb{R}^3, (R,S) \in T_1(SO(3)).$$

In this section, we apply the Newton algorithm introduced in the previous section to solve this problem, and give explicit expressions for calculating all the quantities needed: geodesics, gradient G, Hessian Hess F and the optimal updating vector $\Delta = -\text{Hess}^{-1} G$. Further, we will show, under certain conditions, the Hessian of this function is non-degenerate, whence the Newton algorithm has quadratic convergence in search for the optimal solution.

Notice that for this objective function we have $F(R,S) = F(R,-S)$. It then suffices to optimize $F(R,S)$ as a function on $SO(3) \times \mathbb{P}^2$ instead of $SO(3) \times S^2$, where $\mathbb{P}^2$ is the two dimensional real projective plane (also referred to as $\mathbb{RP}^2$). The calculation of the gradient and Hessian is done by using the explicit formula of geodesics on these manifolds.

On $SO(3)$, the formula for the geodesic at R in the direction $\Delta_1 \in T_R(SO(3)) = R_*(so(3))$ is:

$$R(t) = \exp(R, \Delta_1 t) = R \exp \hat{\omega} t = R(I + \hat{\omega} \sin t + \hat{\omega}^2 (1 - \cos t))$$

where $\hat{\omega} = R^T \Delta_1 \in so(3)$. The last equation is called the *Rodrigues' formula*. $\mathbb{P}^2$ is a Grassmann manifold and it has very simple expression for geodesics too. At the point S along the direction $\Delta_2 \in T_S(S^2)$ the geodesic is given by:

$$S(t) = \exp(S, \Delta_2 t) = S \cos \sigma t + U \sin \sigma t$$

where $\sigma = \|\Delta_2\|$ and $U = \Delta_2 / \sigma$, then $S^T U = 0$ since $S^T \Delta_2 = 0$.

Using the formula for geodesics, we can calculate the first and second derivatives of $F(R,S)$ in the direction $\Delta = (\Delta_1, \Delta_2) \in T_R(SO(3)) \times T_S(S^2)$:

$$\begin{aligned} dF(\Delta) &= \sum_{i=1}^{N} p_i^T R \hat{S} q_i (p_i^T \Delta_1 \hat{S} q_i + p_i^T R \hat{\Delta}_2 q_i), \\ \text{Hess } F(\Delta, \Delta) &= \sum_{i=1}^{N} \left[p_i^T (\Delta_1 \hat{S} + R \hat{\Delta}_2) q_i \right]^2 + p_i^T R \hat{S} q_i \left[p_i^T (-\Delta_1 \Delta_1^T R \hat{S} - \Delta_2^T \Delta_2 R \hat{S} + 2 \Delta_1 \hat{\Delta}_2) q_i \right] \end{aligned}$$

From the first order derivative, the gradient $G = (G_1, G_2) \in T_R(SO(3)) \times T_S(S^2)$ of $F(S,R)$ is:

$$G = \sum_{i=1}^{N} p_i^T R \hat{S} q_i \left(p_i q_i^T \hat{S}^T - R \hat{S} q_i p_i^T R,\ \hat{q}_i R^T p_i - S p_i^T R \hat{q}_i^T S \right)$$

It is direct to check that $R^T G_1 \in so(3)$ and $S^T G_2 = 0$, so the G given by the above expression is a vector in $T_R(SO(3)) \times T_S(S^2)$.

[1] In the literature, for different definitions of the rotation R, the matrix R in the above expression might differ by a transpose.

For any pair of vectors $X, Y \in T_R(SO(3)) \times T_S(S^2)$, polarize Hess $F(\Delta, \Delta)$ we get the expression for Hess $F(X, Y)$:

$$\begin{aligned}
\text{Hess } F(X,Y) &= \frac{1}{4}\left[\text{Hess } F(X+Y, X+Y) - \text{Hess } F(X-Y, X-Y)\right] \\
&= \sum_{i=1}^{N} p_i^T (X_1 \hat{S} + R\hat{X}_2) q_i p_i^T (Y_1 \hat{S} + R\hat{Y}_2) q_i \\
&+ p_i^T R\hat{S} q_i \left[p_i^T \left(-\frac{1}{2}(X_1 Y_1^T + Y_1 X_1^T) R\hat{S} - X_2^T Y_2 R\hat{S} + (X_1 \hat{Y}_2 + Y_1 \hat{X}_2) \right) q_i \right].
\end{aligned}$$

To make sure this expression is correct, let $X = Y = \Delta$. One obtains the same expression for Hess $F(\Delta, \Delta)$ as that obtained directly from the second order derivative.

The following theorem shows that this Hessian is non-degenerate in a neighborhood of the optimal solution, therefore the Newton algorithm will have quadratic rate of convergence by Theorem 3.4 of Smith [14].

Theorem 3 *Consider the objective function $F(R, S)$ as above. Its Hessian is non-degenerate in a neighborhood of the optimal solution if there is a unique (up to a scale) solution to the system of linear equations:*

$$p_i^T E q_i = 0, \quad E \in \mathbb{R}^{3\times 3}, \quad i = 1, \ldots, N.$$

If so, the Riemannian Newton algorithm has quadratic rate of convergence.

Proof: It suffices to prove for any $\Delta \neq 0$, Hess $F(\Delta, \Delta) > 0$. According to the epipolar constraint, at the optimal solution, we have $p_i^T R\hat{S} q_i \equiv 0$. The Hessian is then simplified as:

$$\text{Hess } F(\Delta, \Delta) = \sum_{i=1}^{N} \left[p_i^T (\Delta_1 \hat{S} + R\hat{\Delta}_2) q_i \right]^2.$$

Thus Hess $F(\Delta, \Delta) = 0$ if and only if

$$p_i^T (\Delta_1 \hat{S} + R\hat{\Delta}_2) q_i = 0, \quad i = 1, \ldots, N.$$

Since we also have

$$p_i^T R\hat{S} q_i = 0, \quad i = 1, \ldots, N.$$

Then both $\Delta_1 \hat{S} + R\hat{\Delta}_2$ and $R\hat{S}$ are solutions for the same system of linear equations, hence Hess $F(\Delta, \Delta) = 0$ if and only if

$$\begin{aligned}
& \Delta_1 \hat{S} + R\hat{\Delta}_2 = \lambda R\hat{S}, \quad \text{for some } \lambda \in \mathbb{R} \\
\Leftrightarrow\ & R^T(\Delta_1 \hat{S} + R\hat{\Delta}_2) = \lambda \hat{S} \quad \Leftrightarrow \quad \hat{\omega}\hat{S} + \hat{\Delta}_2 = \lambda \hat{S} \\
\Leftrightarrow\ & \hat{\omega}\hat{S} = \lambda \hat{S}, \text{ and } \Delta_2 = 0, \quad \text{since } S^T \Delta_2 = 0 \\
\Leftrightarrow\ & \omega = 0, \text{ and } \Delta_2 = 0, \quad \text{since } S \neq 0 \\
\Leftrightarrow\ & \Delta = 0.
\end{aligned}$$

■

In the previous theorem, regarding the 3×3 matrix E in the equations $p_i^T E q_i = 0$ as a vector in $\mathbb{R}^9$, one needs at least eight equations to uniquely solve E up to a scale. This implies we need at least eight image correspondences (p_i, q_i) to guarantee the Hessian non-degenerate whence the iterative search algorithm converge in quadratic rate. If we study this problem more carefully, using transversality theory, one may show that five image correspondences in general position is the minimal data to guarantee the Hessian non-degenerate [12]. However, the five point paradigm usually leads to many (up to twenty) ambiguous solutions, as pointed out by Horn [5]. Moreover, numerical errors usually make the algorithm not work exactly on the essential manifold and the extra solutions for the equations $p_i^T E q_i = 0$ may cause the algorithm converge very slowly in these directions. It is not just a coincidence that the conditions for the Hessian to be non-degenerate are exactly the same as that for the eight-point linear algorithm (see [12, 11]) to have a unique solution. A heuristic explanation is that the objective function here is a quadratic form of the epipolar constraint which the linear algorithm is directly based on.

Returning to the Newton algorithm, assume that the Hessian is non-degenerate, *i.e.* invertible. Then, we need to solve for the optimal updating vector Δ such that $\Delta = \text{Hess}^{-1} G$, or equivalently:

$$\text{Hess } F(Y, \Delta) = g(-G, Y) = -dF(Y), \quad \text{for all vector fields } Y.$$

Pick five linearly independent vectors, *i.e.* a basis of $T_R(SO(3)) \times T_S(S^2)$: $E^j, j = 1, \ldots, 5$. One obtains five linear equations:

$$\text{Hess } F(E^j, \Delta) = -dF(E^j), \quad j = 1, \ldots, 5.$$

In general, these five linear equations uniquely determine Δ. In particular, one can choose the simplest basis such that for $j = 1, 2, 3$: $E^j = (R\hat{e}_j, 0)$ with e_j, $j = 1, 2, 3$ the standard basis for $\mathbb{R}^3$, and for $j = 4, 5$: $E^j = (0, e_j)$ such that $\{S, e_4, e_5\}$ form an orthonormal basis for $\mathbb{R}^3$. The vectors e_4, e_5 can be obtained using Gram-Schmidt process.

Define the 5×5 matrix $A \in \mathbb{R}^{5 \times 5}$ and the 5 dimensional vector $\mathbf{b} \in \mathbb{R}^5$ to be:

$$(A)_{jk} = \text{Hess } F(E^j, E^k), \quad (\mathbf{b})_j = -dF(E^j), \quad j, k = 1, \ldots, 5.$$

Then solve for the vector $\mathbf{a} = (a_1, a_2, a_3, a_4, a_5)^T \in \mathbb{R}^5$:

$$\mathbf{a} = A^{-1}\mathbf{b}.$$

Let $\omega = (a_1, a_2, a_3)^T \in \mathbb{R}^3$ and $v = a_4 e_4 + a_5 e_5 \in \mathbb{R}^3$. Then for the optimal updating vector $\Delta = (\Delta_1, \Delta_2)$, we have $\Delta_1 = R\hat{\omega}$ and $\Delta_2 = v$. We now summarize the Riemannian Newton algorithm for the optimal motion recovery, which can be directly implemented.

The Newton Algorithm for Optimal Motion Recovery from the Objective Function:

$$F(R, S) = \sum_{i=1}^{N} (p_i^T R \hat{S} q_i)^2, \quad p_i, q_i \in \mathbb{R}^3, (R, S) \in SO(3) \times \mathbb{P}^2.$$

- At the point $(R, S) \in SO(3) \times \mathbb{P}^2$,
  - Compute the gradient G:

$$G = \sum_{i=1}^{N} p_i^T R \hat{S} q_i \left(p_i q_i^T \hat{S}^T - R\hat{S} q_i p_i^T R,\ \hat{q}_i R^T p_i - S p_i^T R \hat{q}_i^T S \right).$$

- Compute the optimal updating vector $\Delta = -\text{Hess}^{-1}G$,
  * Compute the vectors e_4, e_5 from S using Gram-Schmidt process and obtain the five basis tangent vectors $E^j \in T_R(SO(3)) \times T_S(S^2), 1 \le j \le 5$ as defined in the above,
  * Compute the 5×5 matrix $(A)_{jk} = \text{Hess } F(E^j, E^k), 1 \le j, k \le 5$ using:

$$\begin{aligned}\text{Hess } F(X,Y) = \sum_{i=1}^{N} & p_i^T(X_1\hat{S} + R\hat{X}_2)q_i p_i^T(Y_1\hat{S} + R\hat{Y}_2)q_i \\ & + p_i^T R\hat{S}q_i \left[p_i^T \left(-\frac{1}{2}(X_1Y_1^T + Y_1X_1^T)R\hat{S} - X_2^T Y_2 R\hat{S} + (X_1\hat{Y}_2 + Y_1\hat{X}_2)\right) q_i\right],\end{aligned}$$

  * Compute the 5 dimensional vector $(\mathbf{b})_j = -dF(E^j), 1 \le j \le 5$ using:

$$dF(X) = \sum_{i=1}^{N} p_i^T R\hat{S}q_i(p_i^T X_1\hat{S}q_i + p_i^T R\hat{X}_2 q_i),$$

  * Compute the vector $\mathbf{a} = (a_1, a_2, a_3, a_4, a_5)^T \in \mathbb{R}^5$ such that $\mathbf{a} = A^{-1}\mathbf{b}$,
  * Define $\omega = (a_1, a_2, a_3)^T \in \mathbb{R}^3$ and $v = a_4e_4 + a_5e_5 \in \mathbb{R}^3$. Then the optimal updating vector

$$\Delta = -\text{Hess}^{-1}G = (R\hat{\omega}, v).$$

- Move (R, S) in the direction Δ along the geodesic to $(\exp(R, \Delta_1), \exp(S, \Delta_2))$, using the formula for geodesics on $SO(3)$ and $\mathbb{P}^2$ respectively:

$$\begin{aligned}\exp(R, \Delta_1) &= R(I + \hat{\omega}\sin t + \hat{\omega}^2(1 - \cos t)), \\ \exp(S, \Delta_2) &= S\cos\sigma + U\sin\sigma,\end{aligned}$$

  where $t = \frac{1}{2}\text{tr}(\Delta_1^T\Delta_1), \omega = R^T\Delta_1/t$ and $\sigma = \|\Delta_2\|$, $U = \Delta_2/\sigma$.
- Repeat if $\|G\| \ge \epsilon$ for some pre-determined $\epsilon > 0$.

Note that the Hessian matrix A is symmetric. This reduces almost half of the computation cost.

From the above calculations, note that one can consider the more general objective function with a (positive) weight $w_i \in \mathbb{R}^+$ associated with each image correspondence (p_i, q_i):

$$F(R,S) = \sum_{i=1}^{N} w_i(p_i^T R\hat{S}q_i)^2, \quad p_i, q_i \in \mathbb{R}^3, (R,S) \in T_1(SO(3)).$$

For example, one may choose $w_i^{-1} = \|p_i\|^2\|q_i\|^2$ to convert the image points from perspective projection to spherical projection. Then, in the above algorithm, the expressions of the gradient, dF and the Hessian only need to be slightly modified.

6 Experimental Results

We have implemented the above Riemannian Newton algorithm for optimal motion recovery in Matlab.[2] The Matlab program is used to demostrate the asympotic convergence properties of the algorithm.

[2]The source codes are available at the authors.

Twenty five pairs of image correspondences are randomly generated. The algorithm starts from a random (R_0, S_0) which is generated by adding up to 10 percent normal Gaussian noise on the true motion (R, S) (which, in our setup, gives roughly a corresponding error 0.1 on the objective function). The searching process stops when the error of the objective function $F(R, S)$ reaches the numerical limit of Matlab (about 10^{-31}).

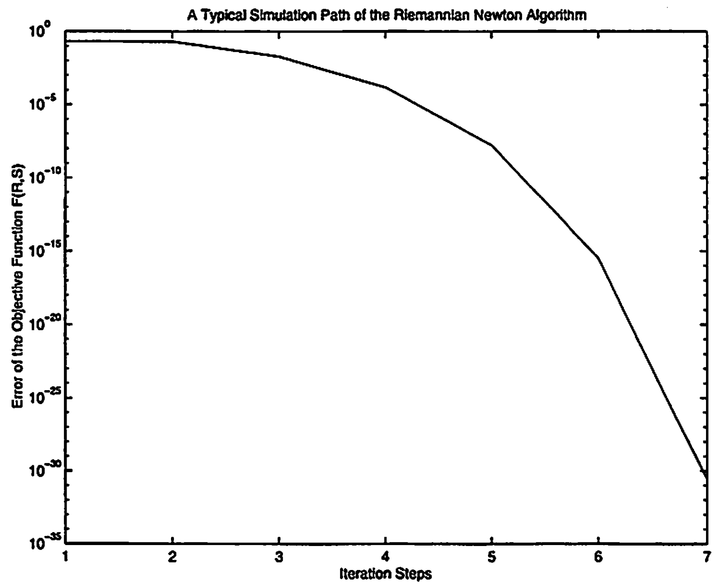

Figure 1: A typical simulation sample path of the Riemannian Newton algorithm searching for the optimal motion. The error indicates the value of the objective function $F(R, S)$.

Figure 1 presents a typical simulation sample path of the searching process. The quadratic rate of convergence is evident, as a consequence of Theorem 3.

7 Optimal Motion Recovery: the Differential Case

The generic similarity between the linear algorithms of the discrete case and the differential case has been revealed in [10, 11]. Their nonlinear algorithms should also be consistent with each other. In the differential case, the epipolar (or Longuet-Higgins) constraint is replaced by its differential version:

$$u^T \hat{v} q + q^T \hat{\omega} \hat{v} q = 0,$$

where $u \in \mathbb{R}^3$ is the optical flow at point $q \in \mathbb{R}^3$ in the image plane, and ω, v are, respectively, the angular velocity and linear velocity of the moving camera frame.[3] Given N optical flow measurements (u_i, q_i), one may consider to recover the motion (ω, v) from optimizing the objective function:

$$f(\omega, v) = \sum_{i=1}^{N} w_i (u_i^T \hat{v} q_i + q_i^T \hat{\omega} \hat{v} q_i)^2, \quad u_i \in \mathbb{R}^3, q_i \in \mathbb{R}^3, (\omega, v) \in \mathbb{R}^3 \times S^2,$$

[3] In the literature, depending on the choice of reference frames, ω or v might be different from here by a sign.

where $w_i \in \mathbb{R}^+$ is a weight associated with each measurement (u_i, q_i). Note when $w_i = 1$, $f(\omega, v)$ is the same objective function used by Soatto in [15], where a Bilinear Projection Algorithm (BPA) is proposed to optimize it.

This is indeed an optimization problem on the space $\mathbb{R}^3 \times S^2$, which has a much simpler Riemannian structure than the essential manifold for the discrete case. Following the same steps we did for the discrete case in the previous sections, one will obtain a Newton algorithm to optimize this objective function $f(\omega, v)$. Since the underlying manifold is nicer now, the expressions for the gradient and the Hessian will be much simpler. Similarly, the Hessian of the function $f(\omega, v)$ is non-degenerate as long as the (differential version) eight-point linear algorithm has a unique solution [11].

8 Discussions and Conclusions

In this paper, we have studied in detail the problem of recovering a discrete motion (displacement) from image correspondences. Similar ideas certainly apply to the differential case we briefly discussed in Section 6, where the rotation and translation are replaced by angular and linear velocities respectively [11]. The optimization schemes for the differential case have also been studied by many researchers, including the most recent Bilinear Projection Algorithm (BPA) proposed by Soatto [15]. However, we hope the Riemannian viewpoint will provide people a different point of view to revisit these schemes.

We only applied Newton's method to the motion recovery problem since Newton's method has the fastest convergence rate (among algorithms using second order information, see [3] for a comparison). In fact, the application of other conjugate gradient algorithms would be easier since they usually only involve the calculation of the first order information (the gradient, not Hessian), at the cost of slower convergence rate.

The motion recovery problem has been studied extensively and many researchers have proposed efficient nonlinear search algorithms. One may find historical reviews of these algorithms in Maybank [12] or Kanatani [7]. Although these algorithms already have good performance in practice, the geometric ideas behind them are not very clear. The non-degeneracy conditions and convergence speed of those algorithms are usually not explicitly addressed. Due to the recent development of optimization methods on Riemannian manifold, we now can have a better mathematical understanding of these algorithms, and propose new geometric algorithms, which exploit the intrinsic geometric structure in the motion recovery problem.

Like most iterative search algorithm, Newton and conjugate gradient algorithms are local methods, *i.e.* they do not guarantee convergence to the global minimum. For the motion recovery problem, one can use linear algorithms to get a closed guess for the optimal motion and initialize the nonlinear search algorithm with it.

As we pointed out in this paper, these Riemannian algorithms can be easily generalized to products of manifolds. Thus, although the proposed Newton algorithm is for single rigid body motion recovery, it can be generalized to multi-body case. Comparing to other existing algorithms and conjugate gradient algorithms, the Newton algorithm involves more computation cost in each iteration step. However, it has the fastest rate of convergence. This is very important when the dimension of the space is high (for instance, multi-body motion recovery problem). This is because the number of search steps usually increases with the dimension, and each step becomes more costly.

However, we will study this in future work. Detailed study of the performance of the proposed algorithm and its sensitivity to noise on synthetic and real images is currently in progress.

9 Acknowledgment

We would like to thank Dr. Steven T. Smith (MIT Lincoln Laboratory) for his pioneering work in the optimization techniques on Riemannian manifolds and his valuable suggestions during the preparation of this manuscript. We also would like to thank professor Alan Weinstein (Berkeley Mathematics Department) for proofreading the manuscript and his insightful comments on some of the differential geometry results associated to this problem.

References

[1] William M. Boothby. *An Introduction to Differential Manifolds and Riemannian Geometry.* Academic Press, second edition, 1986.

[2] K. Danilidis. *Visual Navigation*, chapter Understanding Noise Sensitivity in Structure from Motion. Lawrence Erlbaum Associates, 1997.

[3] Alan Edelman, Tomas Arias, and Steven T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Analysis Applications*, to appear.

[4] Olivier Faugeras. *There-Dimensional Computer Vision.* The MIT Press, 1993.

[5] B. Horn. Relative orientation. *International Journal of Computer Vision*, 4:59–78, 1990.

[6] A. D. Jepson and D. J. Heeger. Linear subspace methods for recovering translation direction. *Spatial Vision in Humans and Robots, Cambridge Univ. Press*, pages 39–62, 1993.

[7] Kenichi Kanatani. *Geometric Computation for Machine Vision.* Oxford Science Publications, 1993.

[8] Shoshichi Kobayashi and Tatsumi Nomizu. *Foundations of Differential Geometry: Volume I.* John Wiley & Sons, Inc., 1996.

[9] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.

[10] Yi Ma, Jana Košecká, and Shankar Sastry. Motion recovery from image sequences: Discrete viewpoint vs. differential viewpoint. In *Proceeding of European Conference on Computer Vision, Volume II*, pages 337–53, 1998.

[11] Yi Ma, Jana Košecká, and Shankar Sastry. Motion recovery from image sequences: Discrete viewpoint vs. differential viewpoint. *Electronic Research Laboratory Memorandum, UC Berkeley*, UCB/ERL M98/11, June 1998.

[12] Stephen Maybank. *Theory of Reconstruction from Image Motion.* Springer-Verlag, 1993.

[13] Luong Q.-T. and O. Faugeras. The fundamental matrix: theory, algorithms, and stability analysis. *International Journal of Computer Vision*, 17(1):43–75, 1996.

[14] Steven T. Smith. *Geometric Optimization Methods for Adaptive Filtering.* PhD thesis. Harvard University, Cambridge, Massachusetts, 1993.

[15] Stefano Soatto and Roger Brockett. Optimal and suboptimal structure from motion. *Proceedings of International Conference on Computer Vision*, to appear.

[16] M. Spetsakis. Models of statistical visual motion estimation. *CVIPG: Image Understanding*, 60(3):300–312, November 1994.

[17] Michael Spivak. *A Comprehensive Introduction to Differential Geometry: Volume II.* Publish or Perish, Inc., second edition, 1979.

[18] C. J. Taylor and D. J. Kriegman. Structure and motion from line segments in multiple images. *IEEE Transactions on PAMI*, 17(11):1021–32, 1995.

[19] I. Thomas and E. Simoncelli. Linear structure from motion. Ms-cis-94-61, Grasp Laboratory, University of Pennsylvania, 1995.

[20] T. Y. Tian, C. Tomasi, and D. Heeger. Comparison of approaches to egomotion computation. In *CVPR*, 1996.

[21] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under ortography. *Intl. Journal of Computer Vision*, 9(2):137–154, 1992.

[22] J. Weng, N. Ahuja, and T. Huang. Optimal motion and structure estimation. *IEEE Transactions PAMI*, 9(2):137–154, 1993.

[23] J. Weng, T.S. Huang, and N. Ahuja. *Motion and Structure from Image Sequences.* Springer Verlag, 1993.