OPTIMAL MOTION ESTIMATION FROM MULTIPLE IMAGES BY NORMALIZED EPIPOLAR CONSTRAINT

by

Yi Ma, Rene Vidal, Shawn Hsu and Shankar Sastry

Memorandum No. UCB/ERL M99/59

1 December 1999

OPTIMAL MOTION ESTIMATION FROM MULTIPLE IMAGES BY NORMALIZED EPIPOLAR CONSTRAINT

by

Yi Ma, Rene Vidal, Shawn Hsu and Shankar Sastry

Memorandum No. UCB/ERL M99/59

1 December 1999

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Optimal Motion Estimation from Multiple Images by Normalized Epipolar Constraint*

Yi Ma René Vidal Shawn Hsu Shankar Sastry

Department of Electrical Engineering and Computer Sciences
University of California at Berkeley, Berkeley, CA 94720-1774
{mayi, rvidal, shawn, sastry}@eecs.berkeley.edu

December 1, 1999

Abstract

In this paper, we study the structure from motion problem as a constrained nonlinear least squares problem which minimizes the so called reprojection error subject to all constraints among multiple images. By converting this constrained optimization problem to an unconstrained one, we contend that multilinear constraints, when used for motion and structure estimation, need to be properly normalized, which makes them no longer tensors. We demonstrate this by using the bilinear epipolar constraints and show how they give rise to a multiview version of the (crossed) normalized epipolar constraint of two views [5]. Such a (crossed) normalized epipolar constraint serves as an optimal objective function for motion (and structure) estimation. This objective function further reveals certain statistic relationship between bilinear and trilinear constraints: Even the rectilinear motion can be correctly estimated by the normalized epipolar constraint as a limit of generic cases, hence trilinear constraints are not really necessary. Since the so obtained objective function is defined naturally on a product of Stiefel manifolds, we show how to use geometric optimization techniques [2] to minimize such a function. Simulation and experimental results are presented to evaluate the proposed algorithm and verify our claims.

1 Introduction

In this paper, we revisit a classic problem in structure from motion: *How to recover camera motion and (Euclidean) scene structure from correspondences of a cloud of points seen in multiple (perspective) images?* With such a vast body of literature studying almost every aspect of this problem (see, for example, reviews of batch methods [13], recursive methods [7, 12], orthographic case [14] and projective reconstruction [16]), it is quite reasonable to ask what, if anything, can still be new in this topic.

First of all, we do not yet have a clear picture about the relationship between multilinear constraints and the (statistic) optimality of motion and structure estimates. Although we have understood very well the geometric (or algebraic) relationship among multilinear constraints [4, 6, 10, 15] (which will be briefly reviewed in Section 3), when it comes to using them for designing

*This work is supported by ARO under the MURI grant DAAH04-96-1-0341 and by AASERT under the grant DAAG55-97-1-0216.

motion or structure recovery algorithms, they are usually used as *objectives* rather than, *constraints*. Many researchers believe that multilinear tensors should be recovered first and, from them, motion and structure could be further retrieved [3]. Algebraically, this is true. Nevertheless, when a noise model is considered and the direct objective is to minimize certain statistics, such as the *reprojection error* (also called *nonlinear least squares error* as in [13]), it becomes quite unclear how to incorporate these multilinear constraints into the objective. More specifically, we want to answer the questions:

> (i) *Can we convert such a constrained estimation (or optimization) problem to an unconstrained one? If so, what weight should be assigned to each constraint?*

Secondly, we have every reason to believe that, for such a constrained estimation problem, its *a posteriori* likelihood function (or some variation of it) still needs to be found. From an estimation theoretic viewpoint, such a function should indeed capture some peculiar statistic nature of the multiview structure from motion problem. Other than the well known algebraic and geometric relationship between bilinear and trilinear constraints, we may ask:

> (ii) *What is the* **statistic relationship** *between bilinear and trilinear constraints? Is trilinear constraint really needed for motion (or structure) estimation in the degenerate rectilinear motion case?*

On the other hand, from an optimization theoretic viewpoint, with such a function we can further understand:

> (iii) *What is the exact nature of the* **optimization** *associated to the original problem? What geometric space does the optimization take place on? Is there any generic optimization technique available for minimizing such a function?*

Finally, in applications which require high accuracy, noise sensitivity becomes the primary issue [1, 5, 17]. Although a specific sensitivity study is needed for every algorithm, it is still possible to study the *intrinsic sensitivity* inherent in the initial problem. From statistics, we know that the Hessian of the *a posteriori* likelihood function at the maximum closely approximates the covariance matrix of the estimates. Hence an *explicit* expression of the likelihood function is absolutely necessary for a systematic study of the intrinsic sensitivity issue. As we will soon see, the normalized epipolar constraint to be derived is such a function and we will show how to compute its Hessian, even though the sensitivity issue is not a main subject of this paper (see Section 5).

In this paper, we will give clear answers to the above questions through the development of a solution to the constrained nonlinear least squares optimization problem which minimizes the reprojection error subject to constraints among multiple images. Question set (i) will be answered in Section 4. The answers will become evident from the derivation and the form of the normalized epipolar constraint. For Question set (ii), the statistic relationship between bilinear and trilinear constraints will be revealed by Simulation 3 in Section 6 and some further explanation will be given in Comment 5. Question set (iii) are to be answered in Section 5 where a generic optimization algorithm is explicitly laid out for minimizing the normalized epipolar constraint. Although our results, including the algorithm, can be easily generalized to trilinear constraints or even to an uncalibrated framework, we choose to present the calibrated case using bilinear (epipolar) constraints so as to clearly convey the main ideas. Nevertheless, we will comment on the trilinear case and uncalibrated case in due time.

Relations to Previous Work: Our algorithm belongs to the so called *batch methods* for motion and structure recovery from multiple views, like that of [13, 14, 16], and is a necessary extension to the unconstrained nonlinear least squares method [13]. We here emphasize again that, our focus is *not* on an algorithm for computing motion or structure faster than the ones in [9, 17], although we will mention briefly how to speed up our algorithm. Instead, we are using our algorithm as a means of revealing the interesting *geometry* in multiview structure from motion, by way of identifying it with the *optimality* of each step of the algorithm. In doing so, one will be able to see what roles multilinear constraints essentially play in the design of optimal algorithms. Especially, the revelation of the statistic relationship between bilinear and trilinear constraints is an important complement to the well known algebraic or geometric results [4, 6, 10, 15]. Our results, especially the normalized epipolar constraint, may also help improve existing *recursive methods* such as in [7, 12] if the filter objective function is modified to the one given by us. Moreover, studying the Hessian of such an objective will allow an extension of existing sensitivity study [1, 5] to the multiview case.

2 Camera Model

We first introduce some notation which will be frequently used in this paper (the notation is consistent with that in [8]). Given a vector $p = [p_1, p_2, p_3]^T \in \mathbb{R}^3$, we define $\widehat{p} \in so(3)$ (the space of skew symmetric matrices in $\mathbb{R}^{3\times 3}$) by:

$$\widehat{p} = \begin{bmatrix} 0 & -p_3 & p_2 \\ p_3 & 0 & -p_1 \\ -p_2 & p_1 & 0 \end{bmatrix}. \tag{1}$$

It then follows from the definition of cross-product of vectors that, for any two vectors $p, q \in \mathbb{R}^3$ we have $p \times q = \widehat{p}q$.

The camera motion is modeled as a rigid body motion in $\mathbb{R}^3$. The displacement of the camera belongs to the special Euclidean group $SE(3)$, represented in **homogeneous coordinates** as:

$$SE(3) = \left\{ g = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} \,\middle|\, p \in \mathbb{R}^3, R \in SO(3) \right\} \tag{2}$$

where $SO(3)$ is the space of 3×3 rotation matrices (orthogonal matrices with determinant $+1$). Clearly, a transformation g is uniquely determined by its rotational part $R \in SO(3)$ and translational part $p \in \mathbb{R}^3$. So sometimes we also express $g \in SE(3)$ by (R, p) as a shorthand. It is also convenient to represent a point $q = [q_1, q_2, q_3]^T \in \mathbb{R}^3$ in homogeneous coordinates as $\underline{q} = [q_1, q_2, q_3, 1]^T \in \mathbb{R}^4$. The set of all such points can also be identified as the subset of $\mathbb{RP}^3$ excluding the plane at infinity, *i.e.*, the plane consisting of all points with coordinates $[q_1, q_2, q_3, 0]^T$. Let $q(t), t \in \mathbb{R}$ be the coordinates of q with respect to the camera coordinate frame at time t. Then the coordinate transformation between $q(t)$ and $q(t_0)$ is given by:

$$\underline{q}(t) = g(t)\underline{q}(t_0). \tag{3}$$

Without loss of generality, we may assume $q(t_0)$ is the coordinates with respect to a pre-fixed inertial frame. In $\mathbb{R}^3$, the above coordinate transformation is equivalent to:

$$q(t) = R(t)q(t_0) + p(t). \tag{4}$$

Define the **projection matrix** $P \in \mathbb{R}^{3\times 4}$ to be $P = [I_{3\times 3}, 0]$. In this paper we always use bold letters to denote image points. Then, in homogeneous coordinates, the (calibrated) image $\mathbf{x} = (x, y, z)^T \in \mathbb{R}^3$ of a point $q \in \mathbb{R}^3$ satisfies:

$$\lambda \mathbf{x} = P\underline{q}. \tag{5}$$

where $\lambda > 0$ encodes the (positive) depth information, defined to be the **scale** of the point q with respect to its image $\mathbf{x}$. For instances, $\lambda = q_3$ for perspective projection and $\lambda = \|q\|$ for spherical projection. If the imaging surface has variable curvature, λ can be more involved.

3 Geometric Interpretation of Multilinear Constraints

Consider n points with coordinates $q^1, q^2, \ldots, q^n \in \mathbb{R}^3$ relative to some inertial coordinate frame. To be consistent in notation, we always use the superscript $i \in \mathbf{N}$ of q^i to enumerate different points. Each of the n points $\{q^i\}_{i=1}^n$ has its corresponding images $\mathbf{x}_1^i, \mathbf{x}_2^i, \ldots, \mathbf{x}_m^i \in \mathbb{R}^3, 1 \le i \le n$, with respect to the m camera frames at m different locations. The subscripts j or k are always used to enumerate the m camera frames. Denote the relative motion (transformation) between the k^{th} and j^{th} frames as $g_{kj} \sim (R_{kj}, p_{kj}) \in SE(3), 1 \le j, k \le m$. For $j = 1, \ldots, m$, let λ_j^i be the scale of the point q^i with respect to its j^{th} image $\mathbf{x}_j^i$. Then from (3) and (5) we have:

$$\begin{bmatrix} \mathbf{x}_1^i & 0 & \cdots & 0 \\ 0 & \mathbf{x}_2^i & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{x}_m^i \end{bmatrix} \begin{bmatrix} \lambda_1^i \\ \lambda_2^i \\ \vdots \\ \lambda_m^i \end{bmatrix} = \begin{bmatrix} Pg_{11} \\ Pg_{21} \\ \vdots \\ Pg_{m1} \end{bmatrix} \underline{q}^i \tag{6}$$

which we rewrite in a more compact notation as:

$$\mathbf{X}^i \vec{\lambda}^i = A\underline{q}^i. \tag{7}$$

We call $A \in \mathbb{R}^{3m\times 4}$ the **motion matrix**. Notice that the motion matrix $A = [a_1, a_2, a_3, a_4]$ has four column vectors $a_l \in \mathbb{R}^{3m}$, $1 \le l \le 4$. A only depends on the relative motions between camera frames and can be viewed as a natural generalization of the **essential matrix** in the two view case.

Now for the j^{th} image $\mathbf{x}_j \in \mathbb{R}^3$ of a point q, we define the vector $\vec{\mathbf{x}}_j \in \mathbb{R}^{3m}$ associated to $\mathbf{x}_j$ to be the j^{th} column of the matrix $\mathbf{X}$:

$$\vec{\mathbf{x}}_j = [0, \ldots, 0, \mathbf{x}_j^T, 0, \ldots, 0]^T \in \mathbb{R}^{3m}, \quad 1 \le j \le m.$$

We then have the well-known results:

Proposition 1 (Multilinear Constraint) *Consider m images $\{\mathbf{x}_j \in \mathbb{R}^3\}_{j=1}^m$ of a point q, and the motion matrix $A = [a_1, a_2, a_3, a_4] \in \mathbb{R}^{3m\times 4}$ of relative motions between camera frames. Then the associated vectors $\{\vec{\mathbf{x}}_j \in \mathbb{R}^{3m}\}_{j=1}^m$ satisfy the following wedge product equation:*

$$a_1 \wedge a_2 \wedge a_3 \wedge a_4 \wedge \vec{\mathbf{x}}_1 \wedge \ldots \wedge \vec{\mathbf{x}}_m = 0. \tag{8}$$

For given camera motions, this equation gives multilinear constraints in the m images $\mathbf{x}_j$ of a single 3D point. Among all the constraints given by this wedge product equation, those involving only four images are called **quadrilinear**, those involving only three images are called **trilinear**, and

those involving only two images are called either **bilinear**, **fundamental** or **epipolar**. It has been shown that constraints involving more than four images are (algebraically) dependent on the trilinear and bilinear ones [4].

For the problem of motion and structure reconstruction, we are more interested in recovering the motion matrix A from measured images $\mathbf{x}_j$'s. In general, coefficients of all the multilinear constraints are minors of the motion matrix A. As for relationships among these coefficients, it is also known that the following statement is true [6]:

Proposition 2 (Multilinear Constraint Dependency) *Coefficients of trilinear or quadrilinear constraints are functions of those of all bilinear (epipolar) constraints (or equivalently the corresponding fundamental matrices) given that the locations of the camera center do not lie on a straight line.*

This proposition states a very important fact: information about the camera motion is already fully contained in the bilinear constraints unless the camera center moves in a straight line – such a motion is also called **rectilinear motion**. Geometrically, this degenerate case is illustrated in Figures 1. In fact, a set of points $\{\mathbf{x}_j\}_{j=1}^m$ on m image planes satisfy all multilinear constraints **if and only if** "rays" extending from camera centers along these image points intersect at a *unique* point in 3D. As a consequence of this geometric interpretation of multilinear constraints, in order for

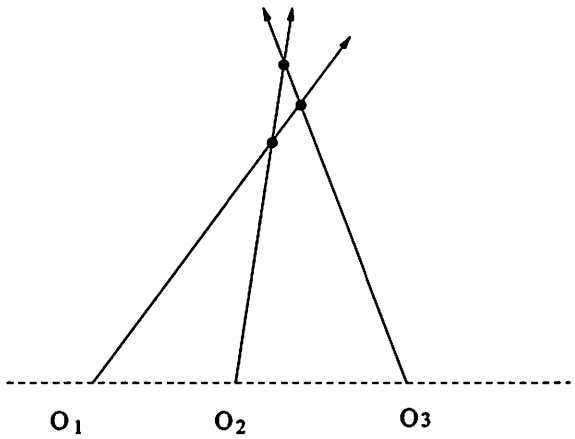

Figure 1: Degeneracy: Centers of camera lie on a straight line. Coplanar constraints are not sufficient to uniquely determine the intersection hence trilinear constraints are needed.

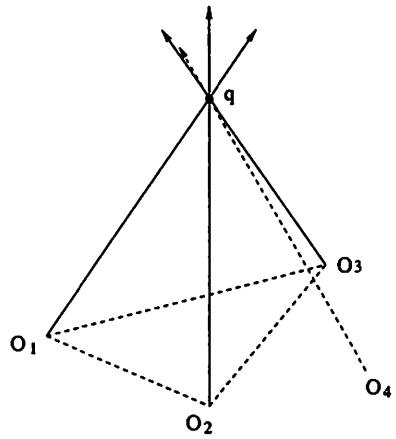

Figure 2: Sufficiency: Centers of camera and the point are not coplanar. Three (bilinear) coplanar constraints are sufficient to uniquely determine the intersection.

an extra image to satisfy all multilinear constraints, it only needs to satisfy two (bilinear) coplanar constraints given that the new camera center is not collinear with the previous ones. For example, in Figure 2, in order for the fourth image to satisfy all multilinear constraints, it is sufficient for the ray (o_4, q) to be coplanar with the ray (o_2, q) and the ray (o_3, q). The coplanar condition between the ray (o_4, q) and the ray (o_1, q) is redundant.

4 Normalized Epipolar Constraint of Multiple Images

Multilinear constraints have conventionally been used to formulate various objective functions for motion recovery. However, if we do use them as constraints, we only need to pick a minimal set

of independent ones. The minimal requirement is needed for Lagrangian multipliers to have a unique solution. The dependency among multilinear constraints suggests that if the centers of the camera do not lie on a straight line, pairwise epipolar constraints already provide a sufficient set of constraints. In this paper we will assume this condition is satisfied unless otherwise stated – Comments 2 and 5 will discuss about the degenerate case. Furthermore, the (pairwise) epipolar constraints among consecutive three images naturally give a minimal set of independent constraints. In this section, we show how to use these constraints to derive a clean form of an optimal objective function for motion (and structure) recovery. In the next section, we will show how to use geometric optimization techniques to find the *optimal* solution which minimizes the objective function derived here.

The rigid body motion between the k^{th} and j^{th} camera frames is $g_{kj} = (R_{kj}, p_{kj}) \in SE(3)$, $1 \leq k, j \leq m$. Thus the coordinates of a 3D point $q \in \mathbb{R}^3$ with respect to frames j and k are related by:

$$q_k = R_{kj} q_j + p_{kj}. \tag{9}$$

Let us denote by $E_{jk} = R_{kj}^T \widehat{p}_{kj} \in \mathbb{R}^{3\times 3}$ the essential matrix associated with the camera motion between the k^{th} and j^{th} frames, then in absence of noise, image points $\mathbf{x}_j^i$ satisfy the epipolar constraints:

$$\mathbf{x}_j^{iT} E_{jk} \mathbf{x}_k^i = 0. \tag{10}$$

In presence of *isotropic* noises, we seek for points $\tilde{\mathbf{x}} = \{\tilde{\mathbf{x}}_j^i\}$ on the image plane and a configuration of m camera frames $\mathcal{G} = \{g_{kj}\}$ such that they minimize the total **reprojection error**. That is, we are to minimize the objective:

$$F(\mathcal{G}, \tilde{\mathbf{x}}) = \sum_{i=1}^{n} \sum_{j=1}^{m} \|\tilde{\mathbf{x}}_j^i - \mathbf{x}_j^i\|^2 \tag{11}$$

subject to the constraints:

$$\tilde{\mathbf{x}}_j^{iT} E_{j,j+1} \tilde{\mathbf{x}}_{j+1}^i = 0, \quad \tilde{\mathbf{x}}_k^{iT} E_{k,k+2} \tilde{\mathbf{x}}_{k+2}^i = 0, \quad \tilde{\mathbf{x}}_l^{iT} e_3 = 1 \tag{12}$$

where $e_3 = (0, 0, 1)^T \in \mathbb{R}^3, 1 \leq j \leq m-1, 1 \leq k \leq m-2, 1 \leq l \leq m$ and $1 \leq i \leq n$. The first two constraints are epipolar constraints among three consecutive images. From the previous section, we know that they form a minimal (but sufficient) set of constraints among multiview images under a generic configuration. We will discuss the degeneracy case in Comments 2 and 5. The last constraint is for the imaging model of perspective projection.[1] Using **Lagrangian multipliers**, the above constrained optimization problem is equivalent to minimizing:

$$\sum_{i=1}^{n} \sum_{j=1}^{m} \left(\|\tilde{\mathbf{x}}_j^i - \mathbf{x}_j^i\|^2 + \sum_{k=j+1}^{j+2} \alpha_{jk}^i \tilde{\mathbf{x}}_j^{iT} E_{jk} \tilde{\mathbf{x}}_k^i 1_{k\leq m} + \beta_j^i (\tilde{\mathbf{x}}_j^{iT} e_3 - 1) \right)$$

for some $\alpha_{jk}^i, \beta_j^i \in \mathbb{R}$. From the necessary condition $\nabla F = 0$ for local minima,

$$2(\tilde{\mathbf{x}}_j^i - \mathbf{x}_j^i) + \sum_{k=j+1}^{j+2} \alpha_{jk}^i E_{jk} \tilde{\mathbf{x}}_k^i 1_{k\leq m} + \sum_{k=j-2}^{j-1} \alpha_{kj}^i E_{kj}^T \tilde{\mathbf{x}}_k^i 1_{k\geq 1} + \beta_j^i e_3 = 0$$

[1] Without loss of generality, we here will only discuss the perspective projection. The spherical projection is similar and hence omitted for simplicity.

for all $i = 1, \ldots, n,\ j = 1, \ldots, m$. Multiplying the above equation by $\hat{e}_3^T \hat{e}_3$ to eliminate β_j^i, we obtain:

$$2(\mathbf{x}_j^i - \tilde{\mathbf{x}}_j^i) = \hat{e}_3^T \hat{e}_3 \Big(\sum_{k=j+1}^{j+2} \alpha_{jk}^i E_{jk} \tilde{\mathbf{x}}_k^i 1_{k \le m} + \sum_{k=j-2}^{j-1} \alpha_{kj}^i E_{kj}^T \tilde{\mathbf{x}}_k^i 1_{k \ge 1} \Big) \tag{13}$$

for all $i = 1, \ldots, n,\ j = 1, \ldots, m$. It is readily seen that, in order to convert the above constrained optimization to an unconstrained one, we need to solve for α_{kj}^i and α_{jk}^i's. For this purpose, we define vectors $\tilde{\mathbf{x}}^i, \mathbf{x}^i, \Delta\mathbf{x}^i \in \mathbb{R}^{3m}$ associated to the i^{th} point: $\tilde{\mathbf{x}}^i = \left[\tilde{\mathbf{x}}_1^{iT}, \ldots, \tilde{\mathbf{x}}_m^{iT}\right]^T, \mathbf{x}^i = \left[\mathbf{x}_1^{iT}, \ldots, \mathbf{x}_m^{iT}\right]^T, \Delta\mathbf{x}^i = \mathbf{x}^i - \tilde{\mathbf{x}}^i$, and the vector of all the Lagrangian multipliers:

$$\alpha^i = [\alpha_{12}^i, \alpha_{13}^i, \alpha_{23}^i, \alpha_{24}^i, \alpha_{34}^i, \ldots, \alpha_{m-2,m}^i, \alpha_{m-1,m}^i]^T \in \mathbb{R}^{2m-3},$$

and matrix $D \in \mathbb{R}^{3m \times 3m}$ with $\hat{e}_3^T \hat{e}_3$ as diagonal blocks:

$$D = \begin{bmatrix} \hat{e}_3^T \hat{e}_3 & \cdots & 0_{3\times3} \\ \vdots & \ddots & \vdots \\ 0_{3\times3} & \cdots & \hat{e}_3^T \hat{e}_3 \end{bmatrix}.$$

We define, for $m \ge 3$, matrices $E = E(m) \in \mathbb{R}^{3m \times 3(2m-3)}$ and $\tilde{X}^i = \tilde{X}^i(m) \in \mathbb{R}^{3m \times (2m-3)}$ recursively as:

$$\begin{aligned} E(m) &= \left[\begin{array}{c|c} E(m-1) & 0_{(3m-9)\times6} \\ 0_{3\times3(2m-5)} & E_m \end{array}\right], \\ \tilde{X}^i(m) &= \left[\begin{array}{c|c} \tilde{X}^i(m-1) & 0_{(3m-9)\times2} \\ 0_{3\times(2m-5)} & \tilde{X}_m^i \end{array}\right] \end{aligned}$$

with

$$E(2) = \begin{bmatrix} E_{12} \\ E_{12}^T \end{bmatrix}, \qquad E_m = \begin{bmatrix} E_{m-2,m} & 0_{3\times3} \\ 0_{3\times3} & E_{m-1,m} \\ E_{m-2,m}^T & E_{m-1,m}^T \end{bmatrix},$$

$$\tilde{X}^i(2) = \begin{bmatrix} \tilde{\mathbf{x}}_2^i \\ \tilde{\mathbf{x}}_1^i \end{bmatrix}, \qquad \tilde{X}_m^i = \begin{bmatrix} \tilde{\mathbf{x}}_m^i & 0_{3\times1} \\ 0_{3\times1} & \tilde{\mathbf{x}}_m^i \\ \tilde{\mathbf{x}}_{m-2}^i & \tilde{\mathbf{x}}_{m-1}^i \end{bmatrix}.$$

We define the **pseudo-array multiplication** $E \cdot \tilde{X}^i$ recursively as:

$$E(m) \cdot \tilde{X}^i(m) = \left[\begin{array}{c|c} E(m-1) \cdot \tilde{X}^i(m-1) & 0_{(3m-9)\times2} \\ 0_{3\times(2m-5)} & E_m \cdot \tilde{X}_m^i \end{array}\right]$$

with

$$E(2) \cdot \tilde{X}^i(2) = \begin{bmatrix} E_{12}\tilde{\mathbf{x}}_2^i \\ E_{12}^T\tilde{\mathbf{x}}_1^i \end{bmatrix},$$

$$E_m \cdot \tilde{X}_m^i = \begin{bmatrix} E_{m-2,m}\tilde{\mathbf{x}}_m^i & 0_{3\times1} \\ 0_{3\times1} & E_{m-1,m}\tilde{\mathbf{x}}_m^i \\ E_{m-2,m}^T\tilde{\mathbf{x}}_{m-2}^i & E_{m-1,m}^T\tilde{\mathbf{x}}_{m-1}^i \end{bmatrix}.$$

Using this notation, the equation (13) can be rewritten as:

$$2\Delta \mathbf{x}^i = DE \cdot \tilde{X}^i \alpha^i. \tag{14}$$

Note that D is a projection matrix, *i.e.*, $D^2 = D$. All the constraints in (12) then can be rewritten compactly as two matrix equations:

$$\tilde{\mathbf{x}}^{iT} E \cdot \tilde{X}^i = 0, \quad D\Delta \mathbf{x}^i = \Delta \mathbf{x}^i. \tag{15}$$

The first equation is simply a matrix expression of all the epipolar constraints. Thus we can solve from equation (14) for α^i:

$$\alpha^i = 2\left(\tilde{X}^{iT} \cdot E^T DE \cdot \tilde{X}^i\right)^{-1} \tilde{X}^{iT} \cdot E^T \mathbf{x}^i \tag{16}$$

given that the matrix $G = \tilde{X}^{iT} \cdot E^T DE \cdot \tilde{X}^i$ is invertible. We call matrix G the **observability Grammian.**

Comment 1 (Observability Grammian) *In general, the observability Grammian is invertible even in cases that the algorithm is not designed for, i.e., the camera motions are such that optical centers lie on a straight line, except for points on the line. In fact, 3D points which make the Grammian degenerate, i.e.,* $\det(G) = 0$ *are very rare. Geometrically, it means that, given a sequence of camera motions, the 3D location of a point whose images make the Grammian degenerate is not* **observable.** *For example, for camera translating in a straight line, points on the line itself then satisfy* $\det(G) = 0$ *hence their images contain no information about neither their 3D location nor the camera motion on the line. In this sense, G can be thought of as the* **observability matrix** *in control theory.*

Substituting the expression of α^i (16) into (14), we then obtain the expression for $\Delta \mathbf{x}^i$ and we have:

$$\|\Delta \mathbf{x}^i\|^2 = \mathbf{x}^{iT} E \cdot \tilde{X}^i \left(\tilde{X}^{iT} \cdot E^T DE \cdot \tilde{X}^i\right)^{-1} \tilde{X}^{iT} \cdot E^T \mathbf{x}^i. \tag{17}$$

Substituting this expression into the objective function $F(\mathcal{G}, \tilde{\mathbf{x}})$ we obtain:

$$F(\mathcal{G}, \tilde{\mathbf{x}}) = \sum_{i=1}^{n} \mathbf{x}^{iT} E \cdot \tilde{X}^i \left(\tilde{X}^{iT} \cdot E^T DE \cdot \tilde{X}^i\right)^{-1} \tilde{X}^{iT} \cdot E^T \mathbf{x}^i. \tag{18}$$

Notice that the terms on the right hand side of the equation are exactly multiview versions of the **crossed normalized epipolar constraints**, but it is *by no means* a trivial sum of the pairwise crossed normalized epipolar constraints [5]. In order to minimize $F(\mathcal{G}, \tilde{\mathbf{x}})$, we need to iterate between the camera motion $\mathcal{G}$ and triangulated structure $\tilde{\mathbf{x}}$, which would be essentially a multiview version of the **optimal triangulation** procedure proposed in [5]. In this paper, however, we will only demonstrate how to obtain optimal motion estimates. Note that, in the expression of $F(\mathcal{G}, \tilde{\mathbf{x}})$, the matrix $\tilde{X}^i$ is a function of $\tilde{\mathbf{x}}^i$ instead of the measured $\mathbf{x}^i$. In general, the difference between $\tilde{\mathbf{x}}^i$ and $\mathbf{x}^i$ is small. Therefore, we may approximate $\tilde{X}^i$ by replacing $\tilde{\mathbf{x}}^i_j$ in $\tilde{X}^i$ by the known $\mathbf{x}^i_j$. We call the resulting matrix as X^i. We then obtain a new function (in camera motion only) $F_n(\mathcal{G}) = F(\mathcal{G}, \mathbf{x})$:

$$F_n(\mathcal{G}) = \sum_{i=1}^{n} \mathbf{x}^{iT} E \cdot X^i \left(X^{iT} \cdot E^T DE \cdot X^i\right)^{-1} X^{iT} \cdot E^T \mathbf{x}^i. \tag{19}$$

In absence of noise, each term of $F_n(\mathcal{G})$ should be:

$$\mathbf{x}^{iT} E \cdot X^i \left(X^{iT} \cdot E^T D E \cdot X^i\right)^{-1} X^{iT} \cdot E^T \mathbf{x}^i = 0. \tag{20}$$

We call this the **normalized epipolar constraint** of multiple images. This is a natural generalization of the normalized epipolar constraint in the two view case [5]. Thus, as in the two view case, $F_n(\mathcal{G})$ can be regarded as a statistically adjusted objective function for directly estimating the camera motions.

Comment 2 (Bilinear vs. Trilinear Constraints) *It is true that one can also use a set of independent trilinear constraints to replace those in (12) and, with a similar exercise, derive its normalized version for motion (and structure) estimation. However, trilinear tensors (as functions of camera motions) do not have as good geometric structure as the bilinear ones. This makes the associated optimization problem harder to describe, even though it is essentially an equivalent optimization problem. One must also be aware that, in the rectilinear motion case, the normalized epipolar constraint objective F_n is not supposed to have a unique minimum (as we will soon see in Simulation 3, in presence of noise, this is not completely true. We will discuss further the new meaning of the minimum in Comment 5) while the corresponding normalized trilinear one always gives a unique solution.*

Comment 3 (Calibrated vs. Uncalibrated Camera) *In the case of an uncalibrated camera, nothing substantial will change in the above derivation except that the essential matrices need to be replaced by fundamental matrices and that the camera intrinsic parameters will introduce 5 new unknowns.*

5 Geometric Optimization Techniques

F_n in the previous section is a function defined on the space of configurations of m camera frames, which is not a regular Euclidean space. Thus conventional optimization techniques cannot be directly applied to minimizing F_n. In this section, we show how to apply newly developed geometric optimization techniques [2, 11] to solve this problem. We here will adopt the Newton's method, although it may not be the fastest, because it allows us to compute the Hessian of the objective function which is potentially useful for sensitivity analysis.

The configuration $\mathcal{G}$ of m camera frames are determined by relative rotations and translations:

$$\begin{aligned} \mathcal{R} &= [R_{21}, R_{32}, \ldots, R_{m,m-1}] \quad \in SO(3)^{m-1}, \\ \mathcal{P} &= [p_{21}^T, p_{32}^T, \ldots, p_{m,m-1}^T]^T \quad \in \mathbb{R}^{3m-3}. \end{aligned}$$

Then $F_n(\mathcal{G})$ can be denoted as $F_n(\mathcal{R}, \mathcal{P})$. It is direct to check that $F_n(\mathcal{R}, \lambda\mathcal{P}) = F_n(\mathcal{R}, \mathcal{P})$ for all $\lambda \neq 0$. Thus $F_n(\mathcal{R}, \mathcal{P})$ is a function defined on the manifold $M = SO(3)^{m-1} \times \mathbb{S}^{3m-4}$ where $\mathbb{S}^{3m-4}$ is a $3m-4$ dimensional spheroid. M is simply a product of Stiefel manifolds and it has total dimension $6m-7$. Furthermore, the (induced) Euclidean metrics on $SO(3)$ and $\mathbb{S}^{3m-4}$ are the same as their canonical metrics as Stiefel manifolds. This gives a natural Riemannian metric $\Phi(\cdot,\cdot)$ on the total manifold M. Note that any tangent vector $\mathcal{X} \in T_{(\mathcal{R},\mathcal{P})}M$ can be represented as $\mathcal{X} = (\mathcal{X}_\mathcal{R}, \mathcal{X}_\mathcal{P})$, with $\mathcal{X}_\mathcal{R} \in T_\mathcal{R}(SO(3)^{m-1})$ and $\mathcal{X}_\mathcal{P} \in T_\mathcal{P}(\mathbb{S}^{3m-4})$ defined by the expressions:

$$\mathcal{X}_\mathcal{R} = [R_{21}\widehat{\omega}_{21}, \ldots, R_{m,m-1}\widehat{\omega}_{m,m-1}], \tag{21}$$

$$\mathcal{X}_\mathcal{P} = [\mathcal{X}_{21}^T, \ldots, \mathcal{X}_{m,m-1}^T]^T \tag{22}$$

where $\omega_{i+1,i} \in \mathbb{R}^3, \mathcal{X}_{i+1,i} \in \mathbb{R}^3, i = 1, \ldots, m-1$ and ${\mathcal{X}_\mathcal{P}}^T \mathcal{P} = 0$. Then the Riemannian metric $\Phi(\cdot, \cdot)$ on the manifold M is explicitly given by:

$$\Phi(\mathcal{X}, \mathcal{X}) = \sum_{i=1}^{m-1} \omega_{i+1,i}^T \omega_{i+1,i} + \mathcal{X}_\mathcal{P}^T \mathcal{X}_\mathcal{P}. \tag{23}$$

Similar to the two view case [5], we can directly apply the Riemannian optimization schemes developed in [2, 11] for minimizing the function $F_n(\mathcal{R}, \mathcal{P})$.

Riemannian Newton's Algorithm Minimizing $F_n(\mathcal{R}, \mathcal{P})$:

1. *Pick an orthonormal basis $\{\mathcal{B}^i\}_{i=1}^{6m-7}$ on $T_{(\mathcal{R},\mathcal{P})}M$. Compute the vector $\mathbf{g} \in \mathbb{R}^{6m-7}$ with its i^{th} entry given by $(\mathbf{g})_i = dF_n(\mathcal{B}^i)$. Compute the matrix $\mathbf{H} \in \mathbb{R}^{(6m-7)\times(6m-7)}$ with its $(i,j)^{th}$ entry given by $(\mathbf{H})_{i,j} = HessF_n(\mathcal{B}^i, \mathcal{B}^j)$. Compute the vector $\delta = -\mathbf{H}^{-1}\mathbf{g} \in \mathbb{R}^{6m-7}$.*
2. *Recover the vector $\Delta \in T_{(\mathcal{R},\mathcal{P})}M$ whose coordinates with respect to the orthonormal basis $\mathcal{B}^i$'s are exactly δ. Update the point $(\mathcal{R}, \mathcal{P})$ along the geodesic to* $\exp(\Delta)$.
3. *Repeat step 1 if $\|\mathbf{g}\| \geq \epsilon$ for some pre-specified tolerance $\epsilon > 0$.*

In the above algorithm, we still need to know: how to pick an orthonormal basis on TM, how to compute geodesics on the manifold M, and how to compute the gradient and Hessian of F_n.

Using the Gram-Schmidt process, we can find vectors $V_\mathcal{P}^1, \ldots, V_\mathcal{P}^{3m-4} \in \mathbb{R}^{3m-3}$ such that, together with $\mathcal{P}$, they form an orthonormal basis of $\mathbb{R}^{3m-3}$. Let $e_1, e_2, e_3 \in \mathbb{R}^3$ be the standard orthonormal basis of $\mathbb{R}^3$. Then a natural orthonormal basis $\{\mathcal{B}^i\}_{i=1}^{6m-7}$ on $T_{(\mathcal{R},\mathcal{P})}M$ is given by:

$$\mathcal{B}^{3i-3+j} = \left([0, \ldots, 0, R_{i+1,i}\widehat{e}_j, 0, \ldots, 0], \mathbf{0}\right)$$

for $1 \leq i \leq m-1$, $1 \leq j \leq 3$ and

$$\mathcal{B}^{3m-3+i} = \left(\mathbf{0}, V_\mathcal{P}^i\right), \quad \text{for } 1 \leq i \leq 3m-4.$$

Given a vector $\mathcal{X} = (\mathcal{X}_\mathcal{R}, \mathcal{X}_\mathcal{P}) \in T_{(\mathcal{R},\mathcal{P})}M$ with $\mathcal{X}_\mathcal{R}$ and $\mathcal{X}_\mathcal{P}$ given by (21) and (22) respectively, the geodesic $(\mathcal{R}(t), \mathcal{P}(t)) = \exp(\mathcal{X}t), t \in \mathbb{R}$ is given by:

$$\mathcal{R}(t) = (R_{21}e^{t\widehat{\omega}_{21}}, R_{32}e^{t\widehat{\omega}_{32}}, \ldots, R_{m,m-1}e^{t\widehat{\omega}_{m,m-1}}), \tag{24}$$

$$\mathcal{P}(t) = \mathcal{P}\cos(\sigma t) + U\sin(\sigma t), \ \sigma = \|\mathcal{X}_\mathcal{P}\|, U = \mathcal{X}_\mathcal{P}/\sigma. \tag{25}$$

The tangent of this geodesic at $t = 0$ is exactly $\mathcal{X}$.

With an orthonormal basis, the computation of gradient and Hessian can be reduced to directional derivatives along geodesics on M. Given a vector $\mathcal{X} \in T_{(\mathcal{R},\mathcal{P})}M$, let $(\mathcal{R}(t), \mathcal{P}(t)) = \exp(\mathcal{X}t)$. Then we have:

$$dF_n(\mathcal{X}) = \frac{dF_n(\mathcal{R}(t), \mathcal{P}(t))}{dt},$$

$$\text{Hess}F_n(\mathcal{X}, \mathcal{X}) = \frac{d^2F_n(\mathcal{R}(t), \mathcal{P}(t))}{dt^2}.$$

Polarizing $\mathrm{Hess}F_n(\mathcal{X},\mathcal{X})$ we can obtain the expression of $\mathrm{Hess}F_n(\mathcal{X},\mathcal{Y})$ for arbitrary $\mathcal{X},\mathcal{Y} \in T_{(\mathcal{R},\mathcal{P})}M$:

$$\begin{aligned} & \mathrm{Hess}F_n(\mathcal{X},\mathcal{Y}) \\ = \; & \frac{1}{4}\big(\mathrm{Hess}F_n(\mathcal{X}+\mathcal{Y},\mathcal{X}+\mathcal{Y}) - \mathrm{Hess}F_n(\mathcal{X}-\mathcal{Y},\mathcal{X}-\mathcal{Y})\big). \end{aligned}$$

According to the definition of gradient, $\mathrm{grad}F_n \in T_{(\mathcal{R},\mathcal{P})}M$, which is given by:

$$dF_n(\mathcal{X}) = \Phi(\mathrm{grad}F_n, \mathcal{X}), \quad \forall \mathcal{X} \in T_{(\mathcal{R},\mathcal{P})}M, \tag{26}$$

is exactly equal to the 1-form dF_n with respect to an orthonormal frame. Therefore, at each point $(\mathcal{R},\mathcal{P})$, we pick the orthonormal basis $\{\mathcal{B}^i\}_{i=1}^{6m-7}$ on $T_{(\mathcal{R},\mathcal{P})}M$ as above and compute the first and second order derivatives of F_n with respect to corresponding geodesics of the base vectors. The gradient and Hessian of F_n are then explicitly expressed by the vector $\mathbf{g}$ and the matrix $\mathbf{H}$ as described in the above algorithm. The updating vector Δ computed in the algorithm is in fact intrinsically defined[2] and satisfies:

$$\mathrm{Hess}F_n(\Delta,\mathcal{X}) = \Phi(-\mathrm{grad}F_n, \mathcal{X}), \quad \forall \mathcal{X} \in T_{(\mathcal{R},\mathcal{P})}M. \tag{27}$$

Note that F_n has a very good structure – only matrix E depends on $(\mathcal{R},\mathcal{P})$ and it consists of blocks of essential matrices $E_{j,j+1}$ and $E_{j,j+2}$. The computation of the Hessian can then be reduced to computing derivatives of these matrices with respect to the chosen base vectors. From the definition of the essential matrix E_{jk}, we have:

$$\begin{aligned} E_{j,j+1} &= R_{j+1,j}^T \widehat{p}_{j+1,j}, \\ E_{j,j+2} &= E_{j,j+1} R_{j+2,j+1}^T + R_{j+1,j}^T E_{j+1,j+2}. \end{aligned}$$

Hence the computation can be further reduced to derivatives of essential matrix $E_{j,j+1}$ only. For a vector $\mathcal{X} \in T_{(\mathcal{R},\mathcal{P})}M$ of the form given by (21) and (22), by direct computation, we have:

$$\begin{aligned} dE_{j,j+1}(\mathcal{X}) &= \widehat{\omega}_{j+1,j}^T R_{j+1,j}^T \widehat{p}_{j+1,j} + R_{j+1,j}^T \widehat{\mathcal{X}}_{j+1,j}, \\ d^2E_{j,j+1}(\mathcal{X},\mathcal{X}) &= \widehat{\omega}_{j+1,j}^2 R_{j+1,j}^T \widehat{p}_{j+1,j} + 2\widehat{\omega}_{j+1,j}^T R_{j+1,j}^T \widehat{\mathcal{X}}_{j+1,j} \\ &\quad - \cdot \mathcal{X}_{\mathcal{P}}{}^T \mathcal{X}_{\mathcal{P}} R_{j+1,j}^T \widehat{p}_{j+1,j} \end{aligned}$$

for $j = 1,\ldots,m-1$. Note that these formulae are consistent to the corresponding ones in the two view case. Thus we now have all the necessary ingredients for implementing the proposed optimization scheme. For any given number of camera frames, we get an optimal estimate of the camera relative configuration by minimizing the normalized epipolar objective F_n.

Comment 4 (Newton vs. Levenberg-Marquardt) *The difference between Newton and Levenberg - Marquardt (LM) methods is that in LM the Hessian is approximated by some form of the objective function's gradient. Since the gradient only involves first order derivatives, LM in general is much less costly in each step. From our implementation of the Newton's algorithm, the Hessian indeed takes more than 95% of the computing time. Nevertheless, we computed the Hessian anyway since the formula would be useful for future sensitivity analysis of motion estimation in the multiview case.*

[2]That is, the definition of Δ is independent of the choice of coordinate frame.

6 Simulations and Experiments

In this section, we show by simulations and experiments the performance of the normalized epipolar constraint. We will apply it to cases *with* or *without* the sufficiency of the epipolar constraint satisfied.

Setup: Table 1 shows the simulation parameters used. In the table, u.f.l. stands for *unit of focal length.* The ratio of the magnitude of translation and rotation, or simply the T/R *ratio*, is

Table 1: Simulation parameters

Parameter	Unit	Value
Number of trials		100 - 500
Number of points		20
Number of frames		3-4
Field of view	degrees	90
Depth variation	u.f.l.	100 - 400
Image size	pixels	500 × 500

compared at the center of the random cloud (scattered in the truncated pyramid specified by the given field of view and depth variation). For all simulations, independent Gaussian noise with std given in unit of pixel is added to each image point. In general, the amount of rotation between consecutive frames is about 20^o and the amount of translation is then automatically given by the T/R ratio. In the following, camera motions will be specified by their translation and rotation axes. For example, between a pair of frames, the symbol XY means that the translation is along the X-axis and rotation is along the Y-axis. If n such symbols are connected by hyphens, it specifies a sequence of consecutive motions. Error measure for rotation is

$$\arccos\left(\frac{tr(R_{ij}\tilde{R}_{ij}^T)-1}{2}\right)$$

in degrees where $\tilde{R}$ is an estimate of the true R. Error measure for translation is the angle between p_{ij} and $\tilde{p}_{ij}$ in degrees where $\tilde{p}$ is an estimate of the true p. All nonlinear (two view or multiview) algorithms are initialized by estimates from the conventional two view linear algorithm.[3]

6.1 Simulation 1: Comparison with Two Frame Bilinear and Normalized Epipolar Constraints

Figure 3 plots the errors of rotation estimates and translation estimates compared with results from the standard 8-point linear algorithm and nonlinear algorithm for pairwise views [5]. As we see, normalization among multiple images indeed performs better than normalization among only pairwise images.

[3] In the multiview case, the relative scales between translations are initialized by triangulation since the directions of translations are known from estimates given by the linear algorithm.

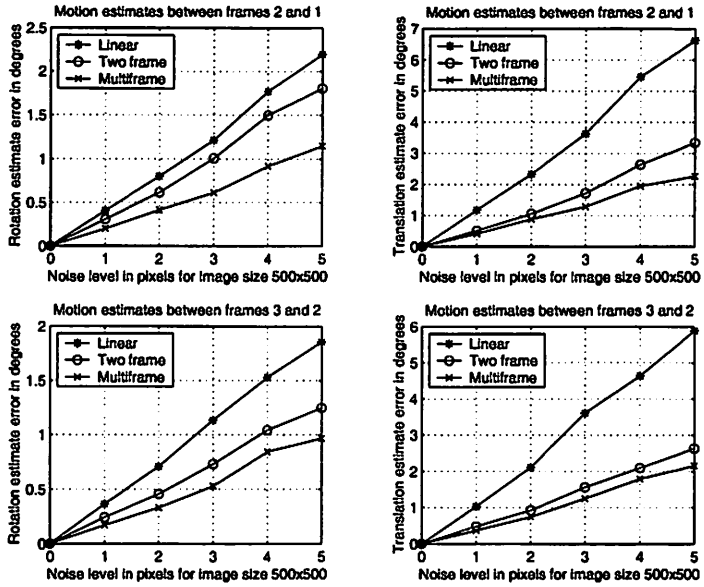

Figure 3: Motion estimate error comparison between normalized epipolar constraint of three frames, normalized epipolar constraint of two frames and (bilinear) epipolar constraint. The number of trials is 500, camera motions are XX-YY and T/R ratio is 1.

6.2 Simulation 2: Axis Dependency Profile

We run the multiview algorithm with consecutive motions along the *same* rotation and translation axes for all nine possible combinations. See Figure 4. Note that our multiview algorithm is not designed to work in rectilinear motion case, such as XX-XX, YY-YY and ZZ-ZZ. Nevertheless, the simulation results in the figure show that the translation estimates still converge to the correct translational direction and the error angles between estimates and the true ones are comparable to other generic cases. As we see, the estimate error is larger when translation along the Z-axis is present. This is because of a smaller signal to noise ratio in this case.

6.3 Simulation 3: A Statistically Stable Solution for Rectilinear Motion from Normalized Epipolar Constraint

From the previous simulation, we notice that the algorithm indeed converges to the correct translational direction in the rectilinear motion case. Then how about the relative scales between consecutive translations? They are usually believed to be captured only by trilinear constraints but not by bilinear ones. This is *not completely true*: The rectilinear motion is indeed a degenerate case for the bilinear constraints, from which there is no unique solution for the relative scales – (for example see Figure 1). However, statistically, the *true* relative scales must be a *stable* solution among all the possible ones. That is, if we properly normalize the epipolar constraint w.r.t. the noise model, the true relative scale should be captured by the epipolar constraints alone as a statistically stable solution. Here, noise essentially plays a positive role of "singling out" the stable solution which otherwise would be lost when degeneracy occurs. Figure 5 plots two histograms of relative scale estimates given by minimizing our normalized epipolar constraint: One is for a rectilinear motion and the other one for a generic motion. Clearly, in both cases, the histogram resembles a Gaussian distribution with the mean centered at the true scale, as a result of the proper normalization. More-

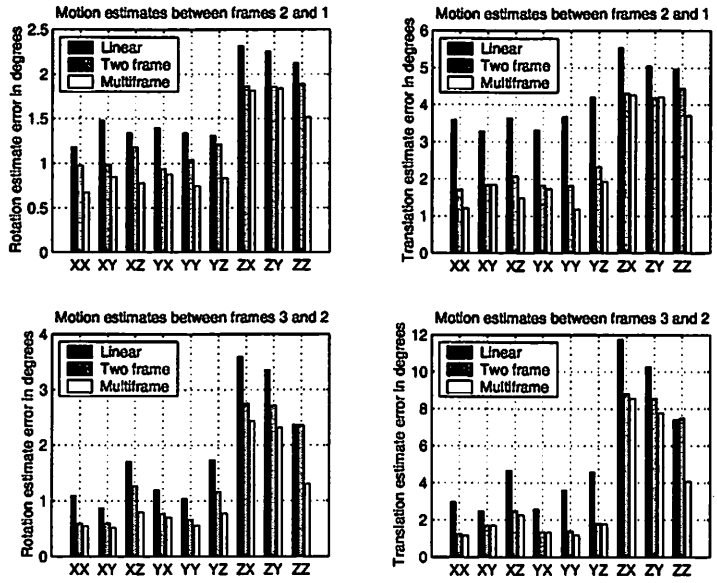

Figure 4: Axis dependency profile: The algorithms are run for all nine combinations of camera rotation and translation w.r.t. the X, Y and Y axes. The number of trials is 100, noise level is 3 pixel std and T/R ratio is 1.

over, the two histograms are comparable to each other, which suggests that, using (normalized) epipolar constraint alone, scale estimates in a degenerate case are not necessarily worse than in a generic case.

Comment 5 (Bilinear vs. Trilinear Constraints Continued) *Simulation 3 reveals a remarkable statistic relationship between bilinear and trilinear constraints: If an optimal estimate is obtained for generic cases, it can still be retrieved as the stable estimate in a degenerate case – the (noise-free) deterministic constraint may be degenerate, but there is no reason for the a posteriori distribution of the estimate to be degenerate as well. Geometrically, the estimate obtained in a degenerate configuration can be interpreted as a "limit" of a sequence of estimates of generic configurations. Such an estimate may also be viewed as the so called "viscous solution" of the*

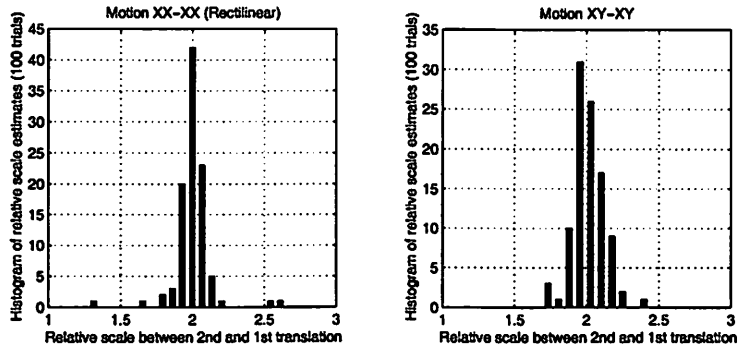

Figure 5: Histogram of relative scale estimates by normalized epipolar constraint in a rectilinear motion case and a generic motion case. The number of trial is 100, noise level is 3 pixel std and the true relative scale between consecutive translation is 2.

normalized epipolar constraint if the Gaussian noise added on images is regarded as some kind of "diffusion". Therefore, in principle, *we do not really need trilinear constraints in order to estimate motion (including relative scales) correctly even in the rectilinear motion case, although such an estimate may be more sensitive or less robust (if the noise model changes).*

6.4 Experiment: Motion Recovery from Real Images

We simply tested our algorithm on a set of real images taken by a commercial pan-tilt camera. Figure 6 shows four images of a cubic corner with feature points, Figure 7 plots the estimated and hand measured actual camera location, and Table 2 gives the errors between the estimated and measured motions. The camera is self-calibrated by Hartley's method for a pure rotating camera.

Since our camera calibration and motion measurements are still crude, errors of this size are

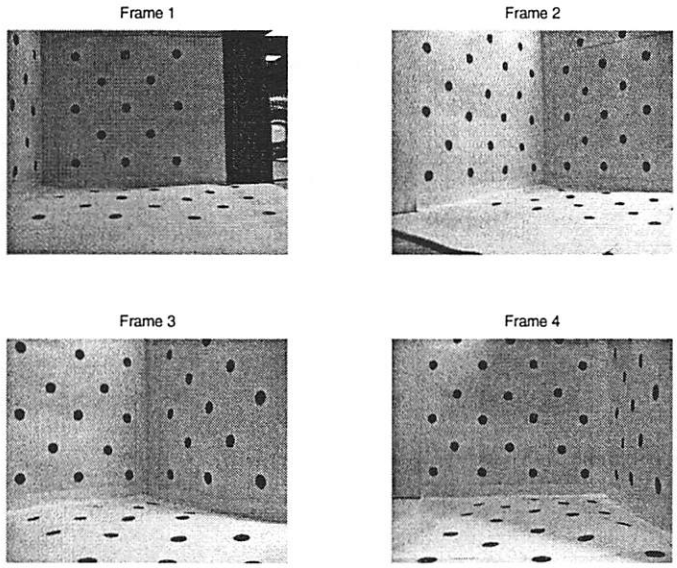

Figure 6: Four images of a cubic corner taken by the camera.

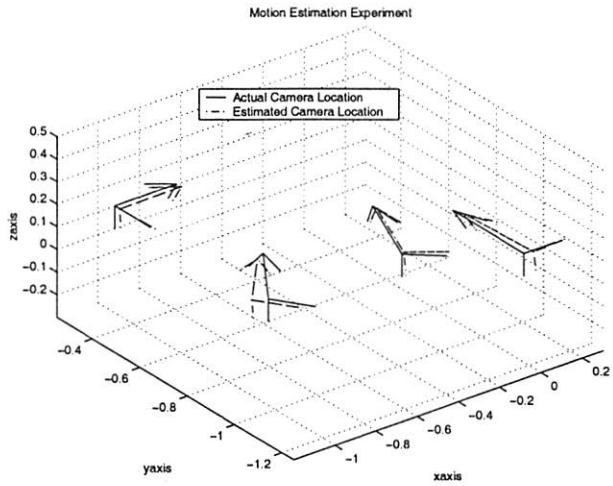

Figure 7: Comparison of estimated and measured camera configuration for the four images.

Table 2: Motion estimate errors in degrees

Motions	Rotation Errors	Translation Errors
Frames 2-1	8.1^o	4.6^o
Frames 3-2	6.3^o	5.8^o
Frames 4-3	4.4^o	4.5^o

expected. We are currently fine-tuning our hardware setup to get better results.

7 Conclusions and Discussions

In this paper, we contend by using (bilinear) epipolar constraint that multilinear constraints need to be properly normalized when used for motion (or structure) estimation. There are several consequences of such a normalization. First, the so obtained objective function is no longer linear hence it does not preserve the tensor structure of multilinear constraints. Second, such a normalization is a natural generalization of the well known normalized epipolar constraint between two images but by no means a trivial sum of them. Third, the normalization not only provides optimal motion (and structure) estimates but, more importantly, reveals certain statistic relationship between epipolar and trilinear constraints – as a necessary complement to the well known algebraic or geometric relationship. We now know that in principle normalized epipolar constraint alone suffices for estimating correct motion (as a statistically stable solution) even in the rectilinear motion case. However, more extensive simulation, experiments and analysis are still needed to evaluate how really practical it is when applied to degenerate cases because it may be less robust to model change. For example, in the case when the noise on the images is no longer isotropic or identically independently distributed, we do not know whether the rectilinear motion can still be well estimated. In a practical implementation, the reader is recommended to extend the idea of normalization in this paper to trilinear constraints or even to an uncalibrated camera.

In this paper, we use the generic Newton's algorithm to minimize the normalized epipolar constraint. One disadvantage is that it is slower than most gradient based algorithms, such as the commonly used Levenberg-Marquardt algorithm. For this reason, we recommend the reader to use those algorithms instead for practical implementations. We here outlined the Newton's algorithm to demonstrate how to compute all the necessary geometric entities associated to the optimization.

References

[1] K. Danillidis. *Visual Navigation.* Lawrence Erlbaum Associates, 1997.

[2] A. Edelman, T. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Analysis Applications*, to appear.

[3] R. Hartley. Lines and points in three views - a unified approach. In *Proceeding of 1994 Image Understanding Workshop*, pages 1006–1016, Monterey, CA USA, 1994. OMNIPRESS.

[4] A. Heyden and K. Åström. Algebraic properties of multilinear constraints. *Mathematical Methods in Applied Sciences*, 20(13):1135–62, 1997.

[5] Y. Ma, J. Košecká, and S. Sastry. Optimization criteria, sensitivity and robustness of motion and structure estimation. In *Proceedings of ICCV workshop on Vision Theory and Algorithm, to appear*, Corfu, Greece, 1999.

[6] Y. Ma, Stefano Soatto, J. Košecká, and S. Sastry. Euclidean reconstruction and reprojection up to subgroups. In *Proceedings of 7th ICCV*, pages 773–80, Corfu, Greece, 1999.

[7] P. F. McLauchlan and D. W. Murry. A unifying framework for structure and motion recovery from image sequences. In *Proceedings of IEEE fifth International Conference on Computer Vision*, pages 314–20, Cambridge, MA USA, 1995. IEEE Com. Soc. Press.

[8] R. M. Murray, Z. Li, and S. Sastry. *A Mathematical Introduction to Robotic Manipulation.* CRC Press, 1993.

[9] J. Oliensis. A multi-frame structure-from-motion algorithm under perspective projection. *In press*, 1999.

[10] A. Shashua. Trilinearity in visual recognition by alignment. In *Proceedings of ECCV, Volume I*, pages 479–484. Springer-Verlag, 1994.

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

[12] S. Soatto, R. Frezza, and P. Perona. Motion estimation via dynamic vision. *IEEE Transactions on Automatic Control*, 41(3):393–413, March 1996.

[13] R. Szeliski and S. B. Kang. Recovering 3D shape and motion from image streams using non-linear least square. *Carnegie Mellon Research Report Series*, 1993.

[14] C. Tomasi and T. Kanade. Shape and motion from image streams: a factorization method. *Cornell TR 92-1270 and Carnegie Mellon CMU-CS-92-104*, 1992.

[15] B. Triggs. Matching constraints and the joint image. In *Proceeding of Fifth International Conference on Computer Vision*, pages 338–43, Cambridge, MA, USA, 1995. IEEE Comput. Soc. Press.

[16] B. Triggs. Factorization methods for projective structure and motion. In *Proceeding of 1996 Computer Society Conference on Computer Vision and Pattern Recognition*, pages 845–51, San Francisco, CA, USA, 1996. IEEE Comput. Soc. Press.

[17] T. Zhang and C. Tomasi. Fast, robust and consistent camera motion estimation. In *Proceeding of CVPR*, 1999.