Optimal Motion Estimation from Multiview Normalized Epipolar Constraint*

René Vidal† Yi Ma‡ Shawn Hsu† Shankar Sastry†

†Electrical Engineering and Computer Sciences
University of California at Berkeley
Berkeley, CA 94720

‡Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Urbana, IL 61801

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 obtain a multiview version of the normalized epipolar constraint of two views. Such a multiview normalized epipolar constraint serves as a statistically optimal objective function for motion (and structure) estimation. Since such a function is defined naturally on a product of Stiefel manifolds, we show how to use geometric optimization techniques to minimize it. We present experimental results on real images to evaluate the proposed algorithm.

1. Introduction

In this paper, we revisit a classic problem in computer vision: *Given a camera undergoing a rigid body motion and observing a cloud of points, recover* **camera motion** *and (Euclidean)* **scene structure** *from their* **correspondences** *among multiple images.*

The problem has been extensively studied in the literature (see, for example, reviews of batch methods [13], recursive methods [8, 12], orthographic case [14] and projective reconstruction [16]). Nevertheless, there are some important issues that have not yet been answered.

First of all, we do not yet have a clear understanding of the relationship between multilinear constraints and the (statistical) optimality of motion and structure estimates. Although we have understood very well the geometric (or algebraic) relationship among multilinear constraints [4, 7, 10, 15] (which will be briefly reviewed in Section 3), when it comes to using them for designing 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*, 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 optimization problem to an unconstrained one? If so, what weight should be assigned to each constraint?*

Secondly, in applications which require high accuracy, noise sensitivity becomes the primary concern [1, 6, 9, 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, evaluated at the maximum, closely approximates the covariance matrix of the estimates. As we will see in Section 5, the multiview normalized epipolar constraint is such a function and we will show how to compute its Hessian. Nevertheless, the sensitivity issue is not a main subject of this paper.

Finally, from an optimization theoretic viewpoint, with such a function we can further understand:

> **(ii)** *What geometric space does the optimization take place on? Is there any generic optimization technique for minimizing such a function?*

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 all 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 multiview normalized epipolar constraint. Question set **(ii)** will be answered in Section 5 where a generic optimization algorithm is explicitly laid out for minimizing the multiview normalized

*This work is supported by ARO MURI grant DAAH04-96-1-0341 and by ONR grant N00014-00-1-0621.

0-7695-1143-0/01 $10.00 © 2001 IEEE

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.

Relations to Previous Work: Our algorithm belongs to the so called *batch methods* for motion and structure recovery from multiple views [13, 14, 16], and is a necessary extension of the unconstrained nonlinear least squares method [13]. We believe that our results, especially the normalized epipolar constraint, may help to improve existing *recursive methods* such as those in [8, 12] if the filter objective function is modified to the one given by us. Moreover, studying the Hessian of such an objective will allow to extend existing sensitivity studies [1, 6] to the multiview case.

2. Notation and Problem Statement

We first introduce some notation which will be frequently used in this paper. We use α, β and λ for scalars, p and q for points in $\Re^3$, $\mathbf{x}$ for image points, $\vec{a}$ and $\vec{\mathbf{x}}$ for vectors and capital letters for matrices. We represent a point $q = [q_1, q_2, q_3]^T \in \Re^3$ in **homogeneous coordinates** as $\underline{q} = [q_1, q_2, q_3, 1]^T \in \Re^4$. Also, given a vector $p = [p_1, p_2, p_3]^T \in \Re^3$, we define $[p]_\times \in so(3)$ (the space of skew symmetric matrices in $\Re^{3\times 3}$) as the matrix generating the cross product, that is, for any two vectors $p, q \in \Re^3$ we have $p \times q = [p]_\times q$.

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

$$SE(3) = \left\{ G = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} \middle| \, p \in \Re^3, R \in SO(3) \right\}, \quad (1)$$

where $SO(3)$ is the space of 3×3 rotation matrices. Let $q(t), t \in \Re$ be the coordinates of q with respect to the camera frame at time t. Then the coordinate transformation $G(t) \in SE(3)$ between $q(t)$ and $q(t_0)$ is given by:

$$\underline{q}(t) = G(t)\underline{q}(t_0) = [(R(t)q(t_0) + p(t))^T \ 1]^T. \quad (2)$$

Without loss of generality, we may assume that $q(t_0)$ are the coordinates of q with respect to a pre-defined inertial frame.

Define $P = [I_{3\times 3}, 0_{3\times 1}] \in \Re^{3\times 4}$ to be the **projection matrix** and $N \subset \Re^3$ to be the imaging surface. Then, the image $\mathbf{x} = (x, y, z)^T \in N$ of a point $q \in \Re^3$ is in general assumed to satisfy the following equation:

$$\lambda \mathbf{x} = P\underline{q}. \quad (3)$$

where $\lambda > 0$ encodes the (unknown positive) **scale** of the point q with respect to its image $\mathbf{x}$. For instance, $\lambda = q_3$ for perspective projection and $\lambda = ||q||$ for spherical projection. If the imaging surface has variable curvature, λ can be more involved. Combining (2) and (3), we have the imaging model for a moving camera:

$$\lambda(t)\mathbf{x}(t) = PG(t)\underline{q}. \quad (4)$$

Problem Statement: Given a set of corresponding image points $\mathbf{x}_1^i, \mathbf{x}_2^i, \ldots, \mathbf{x}_m^i \in N$ of a 3D point $q^i, i = 1, \ldots, n$, with respect to m camera frames (at m unknown locations or time instances), recover the relative motions among the m camera frames and then the 3D locations of the n points with respect to the m camera frames.

To be consistent with the notation, we always use the superscript to enumerate the n different points. We omit the superscript when we refer to a generic single point. The subscript is always used to enumerate the m different camera frames. According to the problem statement, in (4), except for the fact that $\mathbf{x}$ is measured and P is a constant matrix, everything else, *i.e.*, λ, q and G, is unknown and entitled to be recovered from the measured $\mathbf{x}$. As we will soon see, due to some constraints that multiple images of a 3D point must satisfy, the problem of recovering the camera motion G and that of recovering the 3D location of the point q can be very much decoupled. Furthermore, once the camera motion is known, determining the 3D locations of all the feature points is a much simpler problem. Hence, in this paper, we will focus on the problem of recovering camera motion. Once the motion is well estimated, a good reconstruction of 3D structure can also be obtained.

3. Multilinear Constraints

Denote the relative motion between the k^{th} and j^{th} frames as $G_{kj} = (R_{kj}, p_{kj}) \in SE(3), 1 \leq j, k \leq m$. For $i = 1, \ldots, n$, 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 (4) we have:

$$\begin{bmatrix} \mathbf{x}_1^i & 0 & \cdots & 0 \\ 0 & \mathbf{x}_2^i & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{x}_m^i \end{bmatrix} \begin{bmatrix} \lambda_1^i \\ \lambda_2^i \\ \vdots \\ \lambda_m^i \end{bmatrix} = \begin{bmatrix} PG_{11} \\ PG_{21} \\ \vdots \\ PG_{m1} \end{bmatrix} \underline{q}^i, \quad (5)$$

which we rewrite in a more compact notation as:

$$\mathbf{X}^i \vec{\lambda}^i = A\underline{q}^i. \quad (6)$$

We call $A = [\vec{a}_1, \vec{a}_2, \vec{a}_3, \vec{a}_4] \in \Re^{3m\times 4}$ the **motion matrix**. We then have the well-known results:

Proposition 1 (Multilinear Constraint) *Given m images $\{\mathbf{x}_j \in \Re^3\}_{j=1}^m$ of a point q, and the matrix A of relative motions between camera frames, the columns $\{\vec{\mathbf{x}}_j \in \Re^{3m}\}_{j=1}^m$ of matrix $\mathbf{X}$ satisfy the following wedge product equation:*

$$\vec{a}_1 \wedge \vec{a}_2 \wedge \vec{a}_3 \wedge \vec{a}_4 \wedge \vec{\mathbf{x}}_1 \wedge \ldots \wedge \vec{\mathbf{x}}_m = 0. \quad (7)$$

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[1], 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 (on the images $\mathbf{x}_j$'s) involving more than four images are (algebraically) dependent on the trilinear and bilinear ones [4]. It has also been shown that trilinear and quadrilinear constraints are algebraically dependent on bilinear ones when the optical centers of the camera do not lie on a straight line [7]. This degenerate case is also called **rectilinear motion** and is illustrated geometrically in Figure 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 - the "incidental" condition. As a consequence of this interpretation of multilinear constraints, in order for 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 with the ray (o_3, q). The coplanar condition between the ray (o_4, q) and the ray (o_1, q) is redundant.

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 which nonetheless automatically satisfy the incidental condition. In general, it is the coefficients of all the multilinear constraints that contain information about the motion matrix A - in the two view case, these coefficients are exactly the essential matrix. As for relationships among all coefficients, it is also known that the following statement is true [7]:

Proposition 2 (Geometric Dependency) *If the kernels of all the matrices $PG_{k1}, k = 1, \ldots, m$ are linearly dependent, then the coefficients of trilinear or quadrilinear constraints are functions of those of all bilinear constraints.*

It is easy to see that the kernel of the matrix PG_{k1} is spanned by the vector $[-p_{k1}^T R_{k1}, 1]^T \in \Re^4$. Note that $-R_{k1}^T p_{k1} \in \Re^3$ is exactly the optical center of the camera with respect to the initial coordinate frame. Then for all the kernels to be linearly dependent, the optical centers of camera frames 2 to m must all be the same. Therefore, as long as the multiple images are taken at different locations, whatever can be recovered from trilinear constraints (using image correspondences) must be recoverable from epipolar constraints. As we know, epipolar constraints cannot determine the relative scale of translation for rectilinear motion, so neither can trilinear constraints. In Section 6 we will present an experiment showing that statistically this relative scale can still be estimated if we normalize our objective function correctly with respect to a given noise model.

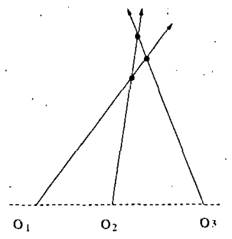

Figure 1: Degeneracy: Centers of camera lie on a straight line. Coplanar constraints are not sufficient to uniquely determine the intersection hence trilinear constraints are needed.

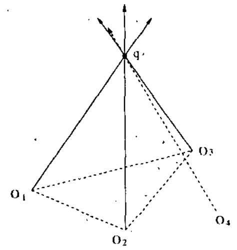

Figure 2: Sufficiency: Centers of camera and the point are not coplanar. Three (bilinear) coplanar constraints are sufficient to uniquely determine the intersection.

4. Multiview Normalized Epipolar Constraint

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. In this paper we will assume that the centers of the camera do not lie on a straight line, unless otherwise stated (Comment 5 will discuss the degenerate case). Therefore, the minimal set will be the set of $2m - 3$ pairwise epipolar constraints among three consecutive images. In this section, we show how to use these constraints to derive a clean form of an statistically optimal objective function for motion (and structure) recovery.

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 $q \in \Re^3$ with respect to frames j and k are related by (see (2)):

$$q_k = R_{kj} q_j + p_{kj}. \tag{8}$$

Let us denote by $E_{jk} = R_{kj}^T [p_{kj}]_\times \in \Re^{3\times3}$ the essential matrix associated with the camera motion between the k^{th} and j^{th} frames, then in the absence of noise, image points $\mathbf{x}_j^i$ satisfy the epipolar constraints:

$$\mathbf{x}_j^{iT} E_{jk} \mathbf{x}_k^i = 0. \tag{9}$$

[1] We recall that in $\Re^k$ the wedge product of ℓ vectors is equal to zero if and only if the subspace generated by the ℓ vectors is of dimension less than ℓ. In other words, all the $\ell \times \ell$ minors of the $k \times \ell$ matrix with columns consisting of those vectors must be zero. Here we have $k = 3m$ and $\ell = m + 4$.

In the 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 want to minimize the objective:

$$F(\mathcal{G}, \tilde{\mathbf{x}}) = \sum_{i=1}^{n} \sum_{j=1}^{m} \|\tilde{\mathbf{x}}_j^i - \mathbf{x}_j^i\|^2 \tag{10}$$

subject to the constraints:

$$\tilde{\mathbf{x}}_j^{iT} E_{j,j+1} \tilde{\mathbf{x}}_{j+1}^i = 0, \quad \tilde{\mathbf{x}}_k^{iT} E_{k,k+2} \tilde{\mathbf{x}}_{k+2}^i = 0, \quad \tilde{\mathbf{x}}_\ell^{iT} e_3 = 1, \tag{11}$$

where $e_3 = (0,0,1)^T \in \Re^3, 1 \leq j \leq m-1, 1 \leq k \leq m-2, 1 \leq \ell \leq m$ and $1 \leq i \leq n$. The first two constraints are epipolar constraints among three consecutive images. The last constraint is for the imaging model of perspective projection.[2] Using **Lagrangian multipliers**, the above constrained optimization problem is equivalent to minimizing:

$$\sum_{i=1}^{n} \sum_{j=1}^{m} \Big(\|\tilde{\mathbf{x}}_j^i - \mathbf{x}_j^i\|^2 + \sum_{k=j+1 \leq m}^{j+2} \alpha_{jk}^i \tilde{\mathbf{x}}_j^{iT} E_{jk} \tilde{\mathbf{x}}_k^i + \beta_j^i (\tilde{\mathbf{x}}_j^{iT} e_3 - 1) \Big), \tag{12}$$

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

$$2(\tilde{\mathbf{x}}_j^i - \mathbf{x}_j^i) + \sum_{k=j+1 \leq m}^{j+2} \alpha_{jk}^i E_{jk} \tilde{\mathbf{x}}_k^i + \sum_{k=j-2 \geq 1}^{j-1} \alpha_{kj}^i E_{kj}^T \tilde{\mathbf{x}}_k^i + \beta_j^i e_3 = 0, \tag{13}$$

for all $i = 1, \ldots, n$, $j = 1, \ldots, m$. Multiplying the above equation by $[e_3]_\times^T [e_3]_\times$ to eliminate β_j^i, we obtain:

$$2(\mathbf{x}_j^i - \tilde{\mathbf{x}}_j^i) = [e_3^T]_\times [e_3]_\times \Big(\sum_{k=j+1 \leq m}^{j+2} \alpha_{jk}^i E_{jk} \tilde{\mathbf{x}}_k^i + \sum_{k=j-2 \geq 1}^{j-1} \alpha_{kj}^i E_{kj}^T \tilde{\mathbf{x}}_k^i \Big), \tag{14}$$

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 $\tilde{\mathbf{x}}^i, \mathbf{x}^i, \Delta\mathbf{x}^i \in \Re^{3m}$ associated with the i^{th} point as $\tilde{\mathbf{x}}^i = [\tilde{\mathbf{x}}_1^{iT}, \ldots, \tilde{\mathbf{x}}_m^{iT}]^T$, $\mathbf{x}^i = [\mathbf{x}_1^{iT}, \ldots, \mathbf{x}_m^{iT}]^T$, $\Delta\mathbf{x}^i = \mathbf{x}^i - \tilde{\mathbf{x}}^i$, the vector of all Lagrangian multipliers as:

$$\vec{\alpha}^i = [\alpha_{12}^i, \alpha_{13}^i, \ldots, \alpha_{m-2,m}^i, \alpha_{m-1,m}^i]^T \in \Re^{2m-3}, \tag{15}$$

and the block diagonal matrix $D \in \Re^{3m \times 3m}$ having $[e_3^T]_\times [e_3]_\times$ as diagonal blocks.

For $m \geq 3$, we define matrices $E = E(m) \in \Re^{3m \times 3(2m-3)}$ and $\tilde{X}^i = \tilde{X}^i(m) \in \Re^{3m \times (2m-3)}$ recursively as:

$$E(m) = \left[\begin{array}{c|c} E(m-1) & 0_{(3m-9)\times 6} \\ 0_{3\times 3(2m-5)} & E_m \end{array}\right],$$
$$\tilde{X}^i(m) = \left[\begin{array}{c|c} \tilde{X}^i(m-1) & 0_{(3m-9)\times 2} \\ 0_{3\times(2m-5)} & \tilde{X}_m^i \end{array}\right],$$

with

$$E(2) = \begin{bmatrix} E_{12} \\ E_{12}^T \end{bmatrix}, \quad E_m = \begin{bmatrix} E_{m-2,m} & 0_{3\times 3} \\ 0_{3\times 3} & E_{m-1,m} \\ E_{m-2,m}^T & E_{m-1,m}^T \end{bmatrix},$$
$$\tilde{X}^i(2) = \begin{bmatrix} \tilde{\mathbf{x}}_2^i \\ \tilde{\mathbf{x}}_1^i \end{bmatrix}, \quad \tilde{X}_m^i = \begin{bmatrix} \tilde{\mathbf{x}}_m^i & 0_{3\times 1} \\ 0_{3\times 1} & \tilde{\mathbf{x}}_m^i \\ \tilde{\mathbf{x}}_{m-2}^i & \tilde{\mathbf{x}}_{m-1}^i \end{bmatrix}.$$

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

$$E(m) \cdot \tilde{X}^i(m) = \left[\begin{array}{c|c} E(m-1) \cdot \tilde{X}^i(m-1) & 0_{(3m-9)\times 2} \\ 0_{3\times(2m-5)} & E_m \cdot \tilde{X}_m^i \end{array}\right], \tag{16}$$

with

$$E(2) \cdot \tilde{X}^i(2) = \begin{bmatrix} E_{12} \tilde{\mathbf{x}}_2^i \\ E_{12}^T \tilde{\mathbf{x}}_1^i \end{bmatrix},$$
$$E_m \cdot \tilde{X}_m^i = \begin{bmatrix} E_{m-2,m} \tilde{\mathbf{x}}_m^i & 0_{3\times 1} \\ 0_{3\times 1} & E_{m-1,m} \tilde{\mathbf{x}}_m^i \\ E_{m-2,m}^T \tilde{\mathbf{x}}_{m-2}^i & E_{m-1,m}^T \tilde{\mathbf{x}}_{m-1}^i \end{bmatrix}. \tag{17}$$

Using this notation, (14) can be rewritten as:

$$2\Delta\mathbf{x}^i = DE \cdot \tilde{X}^i \vec{\alpha}^i. \tag{18}$$

Note that D is a projection matrix, *i.e.*, $D^2 = D$. All the constraints in (11) then can be rewritten compactly as two matrix equations:

$$\tilde{\mathbf{x}}^{iT} E \cdot \tilde{X}^i = 0, \quad D\Delta\mathbf{x}^i = \Delta\mathbf{x}^i. \tag{19}$$

The first equation is simply a matrix expression of all the epipolar constraints. Thus we can solve (18) for $\vec{\alpha}^i$:

$$\vec{\alpha}^i = 2\left((E \cdot \tilde{X}^i)^T DE \cdot \tilde{X}^i \right)^{-1} (E \cdot \tilde{X}^i)^T \mathbf{x}^i, \tag{20}$$

given that the matrix $(E \cdot \tilde{X}^i)^T DE \cdot \tilde{X}^i \in \Re^{2m-3 \times 2m-3}$ is invertible. This is the case not only for general motion but

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

also for the rectilinear motion, except for points on the line containing all the optical centers.

Substituting (18) and (20) into the objective function, we obtain the following expression for $F(\mathcal{G}, \tilde{\mathbf{x}})$:

$$\sum_{i=1}^{n} \mathbf{x}^{iT} E \cdot \tilde{X}^i \left((E \cdot \tilde{X}^i)^T D E \cdot \tilde{X}^i\right)^{-1} (E \cdot \tilde{X}^i)^T \mathbf{x}^i. \quad (21)$$

For $m = 2$ the objective function reduces to:

$$F(\mathcal{G}, \tilde{\mathbf{x}}) = \sum_{i=1}^{n} \frac{(\mathbf{x}_1^{iT} E_{12} \tilde{\mathbf{x}}_2^i + \tilde{\mathbf{x}}_1^{iT} E_{12} \mathbf{x}_2^i)^2}{\|[e_3]_\times E_{12} \tilde{\mathbf{x}}_2^i\|^2 + \|[e_3]_\times E_{12}^T \tilde{\mathbf{x}}_1^i\|^2}. \quad (22)$$

which is the sum of the normalized epipolar constraints for two views. Hence, the terms on (21) are exactly multiview versions of the **crossed normalized epipolar constraints** for two views [6].

In order to minimize $F(\mathcal{G}, \tilde{\mathbf{x}})$, we need to iterate between the camera motion $\mathcal{G}$ and triangulated structure using a multiview version of the **optimal triangulation** procedure proposed in [6]. This procedure consist of (1) initialize $\tilde{\mathbf{x}} = \mathbf{x}$, (2) compute the motion $\mathcal{G}$ with $\tilde{\mathbf{x}}$ fixed, (3) compute the structure with $\mathcal{G}$ fixed , (4) goto (2) until $(\mathcal{G}, \tilde{\mathbf{x}})$ converge.

In this paper, we will only demonstrate how to obtain optimal motion estimates. We then obtain a new function of the camera motion only, $F_n(\mathcal{G}) = F(\mathcal{G}, \tilde{\mathbf{x}})$ with $\tilde{\mathbf{x}}$ fixed. In the absence of noise, each term of $F_n(\mathcal{G})$ should be:

$$\mathbf{x}^{iT} E \cdot X^i \left((E \cdot X^i)^T D E \cdot X^i\right)^{-1} (E \cdot X^i)^T \mathbf{x}^i = 0, \quad (23)$$

where X^i is obtained from $\tilde{X}^i$ by replacing $\tilde{\mathbf{x}}_j^i$ by the known $\mathbf{x}_j^i$. We call this the **multiview normalized epipolar constraint**. This is a natural generalization of the normalized epipolar constraint in the two view case [6]. Thus $F_n(\mathcal{G})$ can be regarded as a statistically adjusted objective function for directly estimating the camera motion.

Comment 1 (Bilinear vs. Trilinear Constraints) *It is true that one can also use a set of independent trilinear constraints to replace those in (11) 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 of a geometric structure as the bilinear ones. This makes the associated optimization problem harder to describe, even though it is essentially an equivalent optimization problem.*

Comment 2 (Calibrated vs. Uncalibrated Camera) *In the uncalibrated case, nothing substantial will change in the above derivation, except that essential matrices need to be replaced by fundamental matrices and that the camera intrinsic parameters will introduce 5 new unknowns.*

5. Geometric Optimization Methods

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 minimize F_n (see Comment 3). In this section, we show how to apply newly developed geometric optimization techniques [2, 11] to solve this problem. Here we 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 is determined by relative rotations and translations:

$$\begin{aligned} \mathcal{R} &= [R_{21}, R_{32}, \ldots, R_{m,m-1}] \in SO(3)^{m-1}, && (24)\\ \mathcal{P} &= [p_{21}^T, p_{32}^T, \ldots, p_{m,m-1}^T]^T \in \Re^{3m-3}. && (25)\end{aligned}$$

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 $\lambda \neq 0$. Therefore, $F_n(\mathcal{R}, \mathcal{P})$ 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 $6m - 7$. 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:

$$\begin{aligned} \mathcal{X}_\mathcal{R} &= [R_{21}[\omega_{21}]_\times, \ldots, R_{m,m-1}[\omega_{m,m-1}]_\times], && (26)\\ \mathcal{X}_\mathcal{P} &= [\mathcal{X}_{21}^T, \ldots, \mathcal{X}_{m,m-1}^T]^T, && (27)\end{aligned}$$

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

$$\Phi(\mathcal{X}, \mathcal{X}) = \sum_{i=1}^{m-1} \omega_{i+1,i}^T \omega_{i+1,i} + \mathcal{X}_\mathcal{P}^T \mathcal{X}_\mathcal{P}. \quad (28)$$

As in the two view case [6], 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 for 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 $\mathbf{g} \in \Re^{6m-7}$ with its i^{th} entry given by $(\mathbf{g})_i = dF_n(\mathcal{B}^i)$. Compute the matrix $\mathbf{H} \in \Re^{(6m-7)\times(6m-7)}$ with its $(i,j)^{th}$ entry given by $(\mathbf{H})_{i,j} = HessF_n(\mathcal{B}^i, \mathcal{B}^j)$. Compute the vector $\delta = -\mathbf{H}^{-1}\mathbf{g} \in \Re^{6m-7}$.*

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 δ. Update the point $(\mathcal{R}, \mathcal{P})$ along the geodesic to* $\exp(\Delta)$.

3. *Go to step 1 if $\|\mathbf{g}\| \geq \epsilon$ for some pre-specified $\epsilon > 0$.*

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

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

$$\mathcal{B}^{3i-3+j} = \left([0, \ldots, 0, R_{i+1,i}[e_j]_\times, 0, \ldots, 0], \mathbf{0}\right) \tag{29}$$

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

$$\mathcal{B}^{3m-3+i} = \left(\mathbf{0}, V_{\mathcal{P}}^i\right), \quad \text{for } 1 \leq i \leq 3m-4. \tag{30}$$

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 (26) and (27) respectively, the geodesic $(\mathcal{R}(t), \mathcal{P}(t)) = \exp(\mathcal{X}t), t \in \Re$ is given by:

$$\mathcal{R}(t) = (R_{21}e^{t[\omega_{21}]_\times}, \ldots, R_{m,m-1}e^{t[\omega_{m,m-1}]_\times}), \tag{31}$$

$$\mathcal{P}(t) = \mathcal{P}\cos(\|\mathcal{X}_{\mathcal{P}}\|\, t) + \mathcal{X}_{\mathcal{P}}\sin(\|\mathcal{X}_{\mathcal{P}}\|\, t)/\|\mathcal{X}_{\mathcal{P}}\|. \tag{32}$$

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 the computation of directional derivatives along geodesics on M. Then we have:

$$dF_n(\mathcal{X}) = \left.\frac{dF_n(\mathcal{R}(t), \mathcal{P}(t))}{dt}\right|_{t=0}, \tag{33}$$

$$\text{Hess}F_n(\mathcal{X}, \mathcal{X}) = \left.\frac{d^2F_n(\mathcal{R}(t), \mathcal{P}(t))}{dt^2}\right|_{t=0}. \tag{34}$$

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

$$\begin{aligned}\text{Hess}F_n(\mathcal{X}, \mathcal{Y}) = \frac{1}{4}(&\text{Hess}F_n(\mathcal{X}+\mathcal{Y}, \mathcal{X}+\mathcal{Y}) - \\ &\text{Hess}F_n(\mathcal{X}-\mathcal{Y}, \mathcal{X}-\mathcal{Y})).\end{aligned} \tag{35}$$

According to its definition, $\text{grad}F_n \in T_{(\mathcal{R},\mathcal{P})}M$ is given by:

$$dF_n(\mathcal{X}) = \Phi(\text{grad}F_n, \mathcal{X}), \quad \forall \mathcal{X} \in T_{(\mathcal{R},\mathcal{P})}M, \tag{36}$$

which is 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 the 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[3] and satisfies:

$$\text{Hess}F_n(\Delta, \mathcal{X}) = \Phi(-\text{grad}F_n, \mathcal{X}), \ \forall \mathcal{X} \in T_{(\mathcal{R},\mathcal{P})}M. \tag{37}$$

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:

$$E_{j,j+1} = R_{j+1,j}^T[p_{j+1,j}]_\times, \tag{38}$$

$$E_{j,j+2} = E_{j,j+1}R_{j+2,j+1}^T + R_{j+1,j}^T E_{j+1,j+2}. \tag{39}$$

Hence the computation can be further reduced to derivatives of essential matrix $E_{j,j+1}$ only. For $\mathcal{X} \in T_{(\mathcal{R},\mathcal{P})}M$ of the form (26) and (27), by direct computation, we have:

$$\begin{aligned}dE_{j,j+1}(\mathcal{X}) = {} & [\omega_{j+1,j}^T]_\times R_{j+1,j}^T[p_{j+1,j}]_\times \\ & + R_{j+1,j}^T[\mathcal{X}_{j+1,j}]_\times,\end{aligned} \tag{40}$$

$$\begin{aligned}\text{Hess}E_{j,j+1}(\mathcal{X}, \mathcal{X}) = {} & [\omega_{j+1,j}]_\times^2 R_{j+1,j}^T[p_{j+1,j}]_\times \\ & + 2[\omega_{j+1,j}]_\times T R_{j+1,j}^T[\mathcal{X}_{j+1,j}]_\times \\ & - \mathcal{X}_{j+1,j}^T\mathcal{X}_{j+1,j}R_{j+1,j}^T[p_{j+1,j}]_\times\end{aligned} \tag{41}$$

for $j = 1, \ldots, m-1$. Note that these formulas are consistent with the corresponding ones in the two view case [6]. Thus we now have all the necessary ingredients for implementing the proposed optimization scheme.

Comment 3 (Riemannian vs. Euclidean Newton) *We could have used the standard (Euclidean) Newton's algorithm, instead. However, since M is not Euclidean, at each iteration the new motion estimates are not necessarily in M. Therefore, it is necessary to project those estimates to M, which not only introduces additional computation but also deteriorates the convergence of the algorithm. In fact, convergence is only guaranteed for the exact Newton's algorithm.*

Comment 4 (Newton vs. Levenberg-Marquardt) *Since we have shown how to compute the gradient and the Hessian of F_n and the geodesics of M, the reader can use any (Riemannian) gradient or Hessian based optimization algorithm, for example Levenberg - Marquardt. In practice, since the computation of the Hessian is costly (95% of the computing time in our implementation of the Newton's algorithm), the reader is recommended to use either a gradient based method or to approximate the Hessian by some form of the gradient. Here, we computed the Hessian anyway since it approximates the covariance matrix of the estimates, which would be useful for future sensitivity analysis of motion estimation in the multiview case.*

6. Experiments on Real Images

In this section we present two experiments. The first one considers an indoor sequence, with the camera undergoing rectilinear motion. The second one involves an outdoor scene with generic motion.

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

In order to work with real images, we need to calibrate the camera, track a set of feature points and establish their correspondences across multiple frames. We calibrated the camera from a set of planar feature points using Zhang's technique [18]. For feature tracking and correspondence, we adapted the algorithm from [19].

The multiview algorithm is then initialized with estimates from the conventional eight-point linear algorithm for two views. Since the translation estimates of the linear algorithm are given up to scale only, for the multiview case an initialization of the relative scale between consecutive translations is required. This is done by triangulation since the directions of the translations are known. For example, the relative scale between p_{21} and p_{32} is $\sin(\alpha)/\sin(\gamma)$ where α is the angle between p_{31} and $R_{21}p_{21}$ and γ is the angle between p_{23} and $R_{13}p_{13}$.

The estimated motion is then compared with the ground truth data. Error measure for translation is the angle between p and $\tilde{p}$ in degrees where $\tilde{p}$ is an estimate of the true p. Error measure for rotation is $\arccos\left(\frac{tr(R\tilde{R}^T)-1}{2}\right)$ in degrees where $\tilde{R}$ is an estimate of the true R.

Camera motions are 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. n of such symbols connected by hyphens specify a sequence of consecutive motions.

6.1. Indoor Rectilinear Motion Sequence

We use 4 images of an indoor scene, with the motion of the camera in a straight line (rectilinear motion) along the Z-axis (see Figure 3). The relative scales between consecutive translations are 2:1 and 1:2, respectively. Even though the motion is rectilinear, relative scales still can be initialized by triangulation, because image measurements are noisy.

Table 1 shows the error between the estimated motion and the actual motion of the camera. It can be observed that the algorithm is able to recover the correct motion and that rotation estimates tend to be more accurate than translation estimates. Table 2 shows the error of the relative scales between consecutive translations. We can see that the scale is estimated with an error below 7%. This shows that it is possible to use bilinear constraints only to estimate motion, even in the case of rectilinear motion.

Comment 5 (Rectilinear Motion) *The experiment reveals an interesting situation: When we formulate the recovery problem using the Lagrangian method, it is necessary and sufficient that the set of constraints on images be algebraically independent. The sufficiency is clearly violated when the motion becomes rectilinear. However, the geometric dependency guarantees that if the image measurements are very close to the true ones, one should be able obtain a close estimate of the true motion from epipolar constraints only. Such an estimate can be interpreted as a "limit" of a sequence of estimates of generic configurations. Therefore,* in the presence of noise, *we do not really need trilinear constraints to estimate motion (including relative scales) correctly even in the rectilinear motion case. Nevertheless, we believe that more theoretical analysis is required to confirm this experimental results.*

6.2. Outdoor Generic Motion Sequence

This sequence consists of 4 images of an outdoor environment, with the camera undergoing motion in the YY-YX-YY (rotation-translation) axes. The relative scale between all the translations is 1:1. The correspondences are shown in Figure 4. The results are shown in Tables 3 and 4.

We can see that the algorithm is able to recover the correct motion. However, the estimates are in general worse than those of the indoor experiment. This is not unexpected. First, the feature points from the indoor sequence are in general closer to the camera. Therefore, even a small amount of motion would cause a noticeable change in the position of the feature points. However, when the points are far away, even a large motion would not cause a significant change in the relative location of these points. Secondly, the conditions of an outdoor environment are more volatile. For example, the leaves on the trees as well as the grass on the lawn can shift positions (due to wind, shadows, etc) from image to image, independent of the camera motion.

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. Third, the normalization not only provides optimal motion (and structure) estimates but, more importantly, reveals certain non-trivial relationship between epipolar and trilinear constraints – as a necessary complement to the well known algebraic or geometric dependency. We now know that, in principle, normalized epipolar constraint alone suffices for estimating correct motion including the relative translation scale even in the rectilinear motion case. However, more extensive simulation, experiments and theoretical analysis are still needed to evaluate how practical the algorithm is when applied to degenerate cases, because it may be very sensitive to noise. In a practical implementation, the reader is also recommended to extend the idea of normalization in this paper to trilinear constraints or even to an uncalibrated camera.

Table 1: Motion estimate errors in degrees

Frames	Rotation Errors	Translation Errors
1-2	0.78°	9.0°
2-3	1.94°	2.8°
3-4	0.91°	1.7°

Table 2: Scale estimate error

Translations	Scale Error
1-2	6.58%
2-3	1.52%

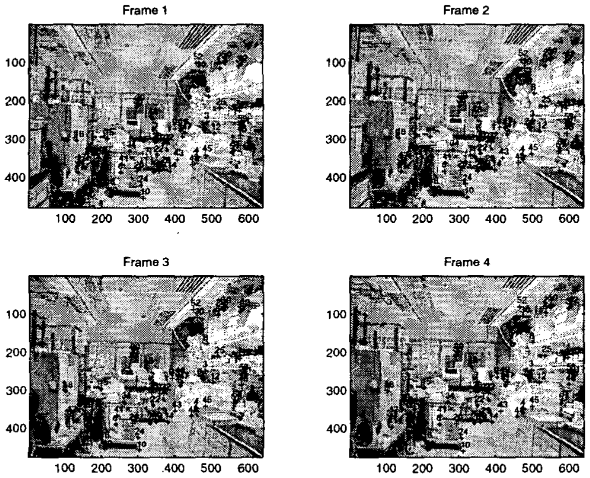

Figure 3: Indoor rectilinear motion image sequence

Table 3: Motion estimate errors in degrees

Frames	Rotation Errors	Translation Errors
1-2	5.1°	18.9°
2-3	4.9°	1.9°
3-4	5.1°	14.5°

Table 4: Scale estimate error

Translations	Scale Error
1-2	9.0%
2-3	7.1%

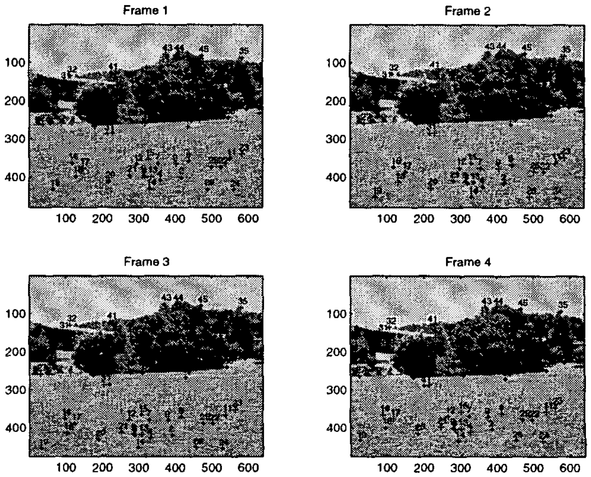

Figure 4: Outdoor generic motion image sequence

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*, 20(2):303–353, 1999.

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

[4] A. Heyden and K. Åström. Algebraic properties of multilinear constraints. *Math. Methods in Applied Sciences*, 20(13):1135–1162, 1997.

[5] K. Kanatani. *Statistical Optimization for Geometric Computation: Theory and Practice*. Elsevier Science, 1996.

[6] 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 Algorithms*, pages 9–16, 1999.

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

[8] P. McLauchlan and D. Murry. A unifying framework for structure and motion recovery from image sequences. In *Proceedings of 5th ICCV*, pages 314–20. IEEE Comp. Soc. Press, 1995.

[9] J. Oliensis. A multi-frame structure-from-motion algorithm under perspective projection. *IJCV*, 34(2-3):163–192, 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, 1996.

[13] R. Szeliski and S. B. Kang. Recovering 3D shape and motion from image streams using non-linear least square. *CMU 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 *Proceedings of 5th ICCV*, pages 338–43. IEEE Comp. Soc. Press, 1995.

[16] B. Triggs. Factorization methods for projective structure and motion. In *Proceedings of CVPR*, pages 845–51. IEEE Comp. Soc. Press, 1996.

[17] T. Zhang and C. Tomasi. Fast, robust and consistent camera motion estimation. In *Proceeding of CVPR*, pages 164–170, 1999.

[18] Z. Zhang. A flexible new technique for camera calibration. *Microsoft Technical Report MSR-TR-98-71*, 1998.

[19] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78(1-2):87–119, 1995.