EUCLIDEAN STRUCTURE AND MOTION FROM IMAGE SEQUENCES

by

Yi Ma, Jana Košecká, and Shankar Sastry

Memorandum No. UCB/ERL M98/38

23 June 1998

EUCLIDEAN STRUCTURE AND MOTION FROM IMAGE SEQUENCES

by

Yi Ma, Jana Košecká, and Shankar Sastry

Memorandum No. UCB/ERL M98/38

23 June 1998

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Euclidean Structure and Motion From Image Sequences *

Yi Ma Jana Košecká Shankar Sastry

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

June 23, 1998

Abstract

In this report we address a problem of Euclidean structure and motion recovery from image sequences and propose a linear method for determining the Euclidean motion and structure information up to a single universal scale regardless of the projection model.

We formulate the problem in the "joint image space" and first review the existing multilinear constraints between m-images of n-points using exterior algebraic notation. It is well known that the projective constraints capture the information about the motion between individual frames and are used to recover it up to a scale. We show how the structural scale information which is lost during the projection process can be recovered using additional Euclidean constraints and propose a linear algorithm for obtaining compatible scales of the joint image matrix entries. We discuss further issues dealing with the uniqueness of the recovery and occlusion.

The presented theory and algorithms are developed for both the discrete and differential case. We outline how the approach can be extended for the hybrid case where for particular image locations both optical flow information and corresponding points in the consecutive frames are available.

Key words: structure from motion, multi-frame, multilinear constraint, Euclidean invariant, differential case, hybrid case.

Introduction

The problem of structure and motion recovery from a image sequence has been in the mainstream of computer vision research for several decades. The original approaches assumed calibrated camera systems focused primarily on recovering structure and motion from two frames and were formulated as minimization problems of nonlinear objective function [12, 11]. Different instances of the objective function represented different error metrics and consequently different parameterizations of the problem. In case the solutions were initialized sufficiently close to the true optimum, these approaches provided good results, but did fail starting with arbitrary guesses. Several natural efforts

*Computer vision research report. This work is supported by ARO under the MURI grant DAAH04-96-1-0341.

for obtaining reasonable initial estimates as well as understanding of proper parameterization of the objective function improved performance of these of techniques [2, 23, 24].

Making simplifying assumptions about the camera model, considering either orthographic [25] or paraperspective model [19] led to a recasting of the motion and shape recovery problem, rendering the problem as a linear one. This made the appealing linear algebraic techniques applicable to the problem and opened an avenue for solving the problem in a multi-frame setting. The motion and shape were extracted using factorization of the large measurement matrix. These efforts were pursued with some hope that by increasing the number of measurements the quality of the recovery process improves, attaining some "global optimum" over the extended sequence of frames. In spite of the impressive results for certain types of scene configurations they fell short for the general setting due to the use of approximate camera models. The techniques were consequently used for providing initial estimates for the nonlinear optimization [2].

The interest in the active vision systems with changing intrinsic and extrinsic parameters initiated an ascent of the uncalibrated methods and gave rise to many new algorithms for recovering both structure and motion [10, 3, 32, 13, 17]. Within the uncalibrated setting the notions of projective and affine structure recovery have been used with the justification that these representations are suitable for certain types of tasks [20]. Another line of work explored the problem of camera self-calibration in the projective case making the Euclidean reconstruction from uncalibrated cameras possible [14, 10]. The agenda of the uncalibrated case gave rise to the stratification of various representations of the three-dimensional structures [4] as well as canonical representations of the constraints between images. The m-frame, n-point motion and structure recovery problem has been explored in the projective setting [28]. The existing projective constraints between images have been characterized [6] and (linear) algorithms for projective structure recovery (structure recovery up to arbitrary projective transformation) developed [29]. In this formulation additional metric constraints which are nonlinear in their nature, can be used in order to recover the unknown projective transformation.

We look at the problem of Euclidean structure and motion recovery from image sequences assuming that intrinsic camera parameters are known. We formulate the problem in the 'joint image space' and first review the existing projective constraints between m-images of n-points using notations and techniques from exterior algebra. The projective constraints decouple the motion information from the information about the structure of the scene and are used for motion recovery from image measurements directly. In order to obtain the Euclidean structure we study existing intrinsic and extrinsic Euclidean invariants which lead to constraints used for recovery of unknown structural scales lost in the projection process. We propose a linear algorithm for computing the motion and the structure up to a single universal scale and address issues dealing with uniqueness of this recovery process and occlusion. We present a formulation of the problem and the algorithm for the differential case and outline how the approach can be extended for hybrid cases where for particular image locations both optical flow information as well as corresponding points in the consecutive images are available.

1 Camera Motion and Projection Models

In this section, we represent the motion of a moving camera as a 3D rigid body motion and introduce the projection models we use for the later structure and motion reconstruction algorithm.

We first introduce some notation which will be frequently used in this paper (the notation is consistent to that in Murray, Li and Sastry [18]). Given a vector $p = (p_1, p_2, p_3)^T \in \mathbb{R}^3$, we define $\hat{p} \in so(3)$ (the space of skew symmetric matrices in $\mathbb{R}^{3\times3}$) by:

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

It then follows from the definition of cross-product of vectors that, for any two vectors $p, q \in \mathbb{R}^3$:

$$p \times q = \hat{p}q. \tag{2}$$

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:

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

where $SO(3)$ is the space of 3×3 rotation matrices (unitary matrices with determinant $+1$). An element $g(t)$ in this group is used to represent the 3D translation and orientation (the displacement) of the camera coordinate frame F_t at time t relative to its initial coordinate frame F_{t_0} at time t_0 (see Figure 1). By an abuse of notation, the group element $g(t)$ serves as both a specification of

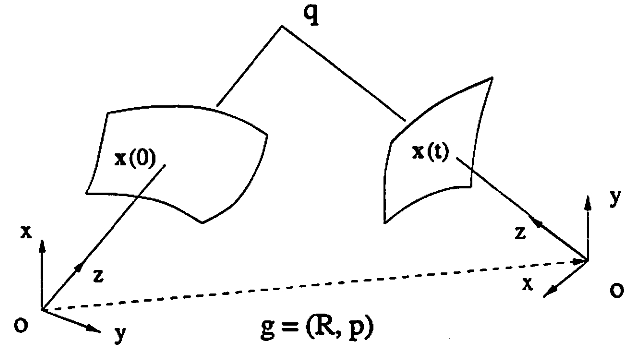

Figure 1: Coordinate frames for specifying rigid body motion of a camera.

the configuration of the camera and as a transformation taking the coordinates of a point from F_{t_0} to F_t. 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 by $g \sim (R, p)$ as a shorthand.

It is convenient to represent a point q in the 3 dimensional Euclidean space in the homogeneous coordinates as

$$q = (q_1, q_2, q_3, 1)^T \in \mathbb{R}^4.$$

The set of all such points can also be identified as the subset of $\mathbb{P}^3$ excluding the **plane at infinity**, *i.e.* the plane consisting of all points with coordinates $(q_1, q_2, q_3, 0)$. 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:

$$q(t) = g(t)q(t_0). \tag{4}$$

In the 3 dimensional representation $\tilde{q} = (q_1, q_2, q_3)^T \in \mathbb{R}^3$, the above coordinate transformation is equivalent to:

$$\tilde{q}(t) = R(t)\tilde{q}(t_0) + p(t). \tag{5}$$

Assume that the camera frame is chosen such that the optical center of the camera, denoted by o, is the same as the origin of the frame. Then the image of a point q in the scene is the point where the ray $< o, q >$ intersects the imaging surface. A sphere or a plane is usually used to model the imaging surface. They are called **spherical projection** and **perspective projection** respectively. However, here we do not assume any specific property of the imaging surface: it could be any smooth 2 dimensional surface with which any ray $< o, q >$ intersects at only one point; or in other words, it can be regarded as (part of) the 2 dimensional projective space $\mathbb{P}^2$. We call this type of imaging as **projective imaging**. The theory and algorithms to be developed will hold for the most general cases of the projective imaging. However, **orthographic projection** is not a case of the projective imaging hence it will not be studied in this paper.

Define the **projection matrix** $P \in \mathbb{R}^{3\times 4}$ to be:

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{6}$$

In this paper we always use bold letters to denote image points. Then, in homogeneous coordinates, the image $\mathbf{x} = (x, y, z)^T \in \mathbb{R}^3$ of $q \in \mathbb{R}^4$ is given by:

$$\lambda \mathbf{x} = Pq. \tag{7}$$

where $\lambda \in \mathbb{R}^+$ encodes the (positive) depth information and we call λ to be the **scale** of the point q with respect to its image $\mathbf{x}$. For instances, $\lambda = q_3$ for perspective projection and $\lambda = \sqrt{q_1^2 + q_2^2 + q_3^2}$ for spherical projection. If the imaging surface has variable curvature, λ can be more involved.

> By **Euclidean structure and motion from image sequences**, we mean the problem of reconstructing the Euclidean transformation g and the depth information λ from the image measurements $\mathbf{x}$.

There are two fundamental transformations in vision: projection and Euclidean rigid body motion. It is well known that motion recovery can be decoupled from the (Euclidean) structure by using the epipolar constraints. However, the scale of the motion cannot be determined by using these constraints alone. Further, when one wants to reconstruct 3D structure of the scene, *i.e.* the scale for each 3D point, Euclidean constraints need to be exploited. In the following, we will study in detail both projective and Euclidean constraints. We then show that these constraints can be used to recover the Euclidean structure and motion up to a universal scale.

In this paper we will address the n-point m-frame problem:

> Reconstructing the relative Euclidean transformations between m image frames and the Euclidean coordinates for n points fixed in the world using their projections in the m images.

We will discuss later how to generalize all the ideas to the differential case and hybrid cases.

Consider n points in the world with homogeneous coordinates $q^1, q^2, \ldots, q^n \in \mathbb{P}^3$ with respect to some inertial coordinate frame. To be consistent in notation, we always use the superscript $j \in \mathbb{N}$ of q^j to enumerate different points. Each of the n points $\{q^j\}_{j=1}^n$ has its corresponding images $\mathbf{x}_1^j, \mathbf{x}_2^j, \ldots, \mathbf{x}_m^j \in \mathbb{R}^3, 1 \leq j \leq n$, with respect to the camera frames at m different positions. The subscript i is always used to enumerate the m camera frames. Denote the relative motion (transformation) between the k^{th} and i^{th} frames as $g_{ki} \sim (R_{ki}, p_{ki}) \in SE(3), 1 \leq i, k \leq m$; and $g_i \sim (R_i, p_i)$ is a shorthand for $g_{i,i-1}, i = 1, \ldots, m$ (with $g_{10} = g_1$). Without loss of generality, we may assume $g_1 = I$. Also, for $i = 1, \ldots, m; j = 1, \ldots, n$, let λ_i^j be the scale of the point q^j with respect to its i^{th} image $\mathbf{x}_i^j$. From (7), we have:

$$\begin{pmatrix} \lambda_1^1\mathbf{x}_1^1 & \lambda_1^2\mathbf{x}_1^2 & \cdots & \lambda_1^n\mathbf{x}_1^n \\ \lambda_2^1\mathbf{x}_2^1 & \lambda_2^2\mathbf{x}_2^2 & \cdots & \lambda_2^n\mathbf{x}_2^n \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_m^1\mathbf{x}_m^1 & \lambda_m^2\mathbf{x}_m^2 & \cdots & \lambda_m^n\mathbf{x}_m^n \end{pmatrix} = \begin{pmatrix} Pg_1 \\ Pg_2g_1 \\ \vdots \\ Pg_m \cdots g_1 \end{pmatrix} \left(q^1, q^2, \ldots, q^n\right). \tag{8}$$

Then our goal is to reconstruct the relative motions $\{g_i\}_{i=1}^m$ and the scales $\{\lambda_i^j\}_{i=1,j=1}^{m,n}$ from image measurements $\{\mathbf{x}_i^j\}_{i=1,j=1}^{m,n}$. To simplify the notation, define matrices $\mathbf{Y}, \mathbf{X} \in \mathbb{R}^{3m\times n}$ and $\Lambda \in \mathbb{R}^{m\times n}$ to be:

$$\mathbf{Y} = (\lambda_i^j\mathbf{x}_i^j), \quad \mathbf{X} = (\mathbf{x}_i^j), \quad \Lambda = (\lambda_i^j), \quad 1 \leq i \leq m, 1 \leq j \leq n.$$

The matrix $\mathbf{X}$ will be called **image matrix**, Λ the **scale matrix** and $\mathbf{Y}$ the **scaled image matrix.** Imposing the **positive depth** constraint, the entries of the scale matrix Λ are always positive. Define the matrices $A \in \mathbb{R}^{3m\times 4}$ and $Q \in \mathbb{R}^{4\times n}$ to be:

$$A = \begin{pmatrix} Pg_1 \\ Pg_2g_1 \\ \vdots \\ Pg_m \cdots g_1 \end{pmatrix}, \quad Q = \left(q^1, q^2, \ldots, q^n\right). \tag{9}$$

Notice matrix A has four columns. We denote them as

$$A = (a_1, a_2, a_3, a_4)$$

with each column vector $a_k \in \mathbb{R}^{3m}$, $1 \leq k \leq 4$. Matrix A only depends on the relative motions between image frames. We will refer to the matrix A as the **motion matrix**. The motion matrix is a natural generalization of the **essential matrix** in the 2-frame case. The matrix Q will be called the **coordinate matrix** since it is just the collection of all the homogeneous coordinates of 3D points. We will use the convention that the last entry of the homogeneous coordinates is always 1. Hence the last row of matrix Q is always $(1, \ldots, 1)$.

Using the new notation, we can write equation (8) in shorthand:

$$\Lambda \odot \mathbf{X} = AQ, \tag{10}$$

where $\odot$ means component-wise multiplication except that λ_i^j multiplies 3 rows. Unless otherwise stated, we treat the matrix $\mathbf{X}$ and $\mathbf{Y}$ as $m \times n$ (instead of $3m \times n$) matrices with $\mathbf{x}_i^j \in \mathbb{R}^3$ and $\lambda_i^j\mathbf{x}_i^j \in \mathbb{R}^3$ as the $(i,j)^{th}$ entry, respectively.

Definition 1 *Given an image matrix* $\mathbf{X}$*, a scale matrix* Λ *with positive entries is called* **compatible** *with* $\mathbf{X}$ *if there exist a motion matrix* A *and a coordinate matrix* Q *such that* $\Lambda \odot \mathbf{X} = AQ$*.*

Then, if one can find a compatible scale matrix Λ for an image matrix $\mathbf{X}$, the rows of the equation (8) give:

$$(\lambda_i^1 \mathbf{x}_i^1, \lambda_i^2 \mathbf{x}_i^2, \ldots, \lambda_i^n \mathbf{x}_i^n) = P g_i \cdots g_1 (q^1, q^2, \ldots, q^n). \tag{11}$$

Let $(R, p) \sim g = g_i \cdots g_1$ and $\tilde{q}^j \in \mathbb{R}^3, 1 \leq j \leq n$ be the 3D coordinates of $\{q^j\}_{j=1}^n$, we have:

$$(\lambda_i^1 \mathbf{x}_i^1, \lambda_i^2 \mathbf{x}_i^2, \ldots, \lambda_i^n \mathbf{x}_i^n) = R(\tilde{q}^1, \tilde{q}^2, \ldots, \tilde{q}^n) + p.$$

In other words, for each $i = 1, \ldots, m$, the entries $\{\lambda_i^j \mathbf{x}_i^j\}_{j=1}^n$ exactly give the 3D coordinates of points $\{q^j\}_{j=1}^n$ with respect to the i^{th} frame. The n 3D points $\{q^j\}_{j=1}^n$ are then reconstructed up to some Euclidean transformation $g \sim (R, p)$. Further, the relative motions $\{g_i\}_{i=1}^m$ hence $\{g_{ki}\}_{k=1,i=1}^{m,m}$ can be recovered from the motion matrix A. Therefore **the problem of reconstructing Euclidean structure and motion is really a problem of finding a compatible scale matrix Λ for a given image matrix $\mathbf{X}$.** The main purpose of this paper is to show that under certain conditions the compatible scale matrix Λ is unique up to a scale, and there are efficient algorithms to compute it.

2 Projective Constraints and Motion Recovery

In this section, we study constraints in the images which are invariant under projective transformations, called **projective constraints**. Since these constraints are invariant under projective transformation, they can only be used to recover the projective structure of the n points (Triggs [29]), not the Euclidean structure. However, as we will soon see, these constraints decouple the motion estimation problem from the (Euclidean) structure. The results to be presented here follow from the work by Triggs and Faugeras *et al* [30, 28, 6]. However, as a brief **review** of these theories, we here give a simpler representation using **exterior algebra** (or **Grassmann algebra**) notation, and one will soon see, these results can also be easily generalized to the differential case and some hybrid cases in later sections.

The projective constraints on multiple images of a point actually state a very **simple fact:**

> The image of a four dimensional vector space (*i.e.* the homogeneous space representing all the 3D points) under a linear transformation is at most four dimensional.

Notice that each column of the scaled image matrix $\mathbf{Y}$ is in the four dimensional subspace of $\mathbb{R}^{3m}$ spanned by the four columns of the motion matrix A, *i.e.* $a_1, \ldots, a_4$. Now for the i^{th} image $\mathbf{x}_i \in \mathbb{R}^3$ of the point q, we construct the vector $\tilde{\mathbf{x}}_i \in \mathbb{R}^{3m}$ associated to $\mathbf{x}_i$ by filling zeros in components not belonging to the i^{th} image:

$$\tilde{\mathbf{x}}_i = (0, \ldots, 0, \mathbf{x}_i^T, 0, \ldots, 0)^T \in \mathbb{R}^{3m}, \quad 1 \leq i \leq m.$$

We then have:

Theorem 1 (Projective Constraints) *Consider m images $\{\mathbf{x}_i\}_{i=1}^m \in \mathbb{R}^3$ of a point q, and the motion matrix for the relative motions between image frames is $A = (a_1, a_2, a_3, a_4) \in \mathbb{R}^{3m\times 4}$. Then the associated vectors $\{\tilde{\mathbf{x}}_i\}_{i=1}^m \in \mathbb{R}^{3m}$ satisfy the following wedge product equation:*

$$a_1 \wedge a_2 \wedge a_3 \wedge a_4 \wedge \tilde{\mathbf{x}}_1 \wedge \ldots \wedge \tilde{\mathbf{x}}_m = 0. \tag{12}$$

The wedge product equation (12) is an alternative way of saying that the vectors $\tilde{\mathbf{x}}_i, 1 \leq i \leq m$ are in the four dimensional subspace of $\mathbb{R}^{3m}$ spanned by the vectors a_1, a_2, a_3, a_4.

Proof: Let $\lambda_i, i = 1, \ldots, m$ be the scale of q with respect to its i^{th} image $\mathbf{x}_i$. By (8), the vector $Y = \sum_{i=1}^m \lambda_i \tilde{\mathbf{x}}_i$ is in span$\{a_1, a_2, a_3, a_4\}$, *i.e.* Y, a_1, a_2, a_3, a_4 are linearly dependent. This gives

$$a_1 \wedge a_2 \wedge a_3 \wedge a_4 \wedge Y = 0.$$

Without loss of generality, we may assume $\lambda_1 = 1$. Then, $\tilde{\mathbf{x}}_1 = Y - \sum_{i=2}^m \lambda_i \tilde{\mathbf{x}}_i$. Substituting this back into the wedge product on the left hand side of (12), it yields:

$$\begin{aligned}
& a_1 \wedge a_2 \wedge a_3 \wedge a_4 \wedge \left(Y - \sum_{i=2}^m \lambda_i \tilde{\mathbf{x}}_i\right) \wedge \tilde{\mathbf{x}}_2 \wedge \ldots \wedge \tilde{\mathbf{x}}_m \\
= \;& a_1 \wedge a_2 \wedge a_3 \wedge a_4 \wedge Y \wedge \tilde{\mathbf{x}}_2 \wedge \ldots \wedge \tilde{\mathbf{x}}_m - \\
& a_1 \wedge a_2 \wedge a_3 \wedge a_4 \wedge \left(\sum_{i=2}^m \lambda_i \tilde{\mathbf{x}}_i\right) \wedge \tilde{\mathbf{x}}_2 \wedge \ldots \wedge \tilde{\mathbf{x}}_m \\
= \;& 0.
\end{aligned}$$

Thus the wedge product constraint (12) holds. ∎

The above wedge product constraint is exactly the constraint given by Triggs [28]. It gives constraints on the multiple images of a single point. $\wedge_{k=1}^4 a_k \wedge_{i=1}^m \tilde{\mathbf{x}}_i$ is an element in $\bigwedge^{m+4}(\mathbb{R}^{3m})$, the space of all skew-symmetric elements in $\otimes_{i=1}^{m+4}\mathbb{R}^{3m}$, or in other words, the space of all $(m+4)$-forms of $\mathbb{R}^{3m}$. So $\wedge_{k=1}^4 a_k \wedge_{i=1}^m \tilde{\mathbf{x}}_i$ has the common form:

$$\sum_{1\leq i_1 < i_2 < \cdots < i_{m+4} \leq 3m} f_{i_1, i_2, \ldots, i_{m+4}} (e_{i_1} \wedge \ldots \wedge e_{i_{m+4}}) \tag{13}$$

where $\{e_l\}_{l=1}^{3m}$ is a standard basis for $\mathbb{R}^{3m}$. Notice that it has $\begin{pmatrix} 3m \\ m+4 \end{pmatrix}$ linear independent terms. By the wedge product constraint (12), all the coefficients f's have to be zero. This gives the same number of homogeneous constraints of degree $m+4$ in terms of the entries of a_k's and $\tilde{\mathbf{x}}_i$'s. However, since many of the entries of $\tilde{\mathbf{x}}_i$ are zeros, most of the homogeneous constraints will be trivial or reducible.

Let each image point $\mathbf{x}_i = (x_i, y_i, z_i)$. Then each $\tilde{\mathbf{x}}_i$ has three linearly independent terms:

$$\tilde{\mathbf{x}}_{i+1} = x_{i+1}e_{3i+1} + y_{i+1}e_{3i+2} + z_{i+1}e_{3i+3}, \quad i = 0, \ldots, m-1.$$

The wedge product is just:

$$\wedge_{k=1}^4 a_k \wedge_{i=1}^m \tilde{\mathbf{x}}_i = \wedge_{k=1}^4 a_k \wedge_{i=0}^{m-1} (x_{i+1}e_{3i+1} + y_{i+1}e_{3i+2} + z_{i+1}e_{3i+3}). \tag{14}$$

Notice that any non-trivial term $f_{i_1,i_2,\ldots,i_{m+4}}$ contains at least one non-zero entry from each $\tilde{\mathbf{x}}_i$. Further, any particular f is always **multi-linear** in all involved $\mathbf{x}_i$'s; if a non-trivial f depends on exactly one entry from $\tilde{\mathbf{x}}_i$, the involved entry of $\mathbf{x}_i$ can be reduced when considering $f = 0$. Therefore, $f = 0$ only imposes constraints on $\mathbf{x}_i$ which has more than two entries showing up in the same f. Grouping the $m + 4$ indices: $i_1, \ldots, i_{m+4}$ into m non-empty sets, then the numbers of elements in the sets have three types (up to a permutation of the m sets): $(3, 3, 1, \ldots, 1)$, $(3, 2, 2, 1 \ldots, 1)$ and $(2, 2, 2, 2, 1 \ldots, 1)$. Any term f of these three types of indices has respectively two, three or four $\mathbf{x}_i$'s with more than two entries involved. The corresponding constraints imposed by $f = 0$ are called **bilinear, trilinear** or **quadrilinear** constraints in the $\mathbf{x}_i$'s, respectively.

These constraints given by the wedge product equation are invariant under projective transformations of the n points. To see this, for any $T \in GL(4, \mathbb{R})$ the group of all 4×4 invertible real matrices, let $A' = AT^{-1}$ and $Q' = TQ$, we still have $\Lambda \odot \mathbf{X} = A'Q'$. The new wedge product constraint given is just:

$$\begin{aligned} & a_1' \wedge a_2' \wedge a_3' \wedge a_4' \wedge \tilde{\mathbf{x}}_1 \wedge \ldots \wedge \tilde{\mathbf{x}}_m \\ = \; & \det(T^{-1}) \cdot a_1 \wedge a_2 \wedge a_3 \wedge a_4 \wedge \tilde{\mathbf{x}}_1 \wedge \ldots \wedge \tilde{\mathbf{x}}_m \end{aligned}$$

Since $\det(T^{-1}) \neq 0$, this wedge product gives the same set of constraints. This is also why we name these constraints as **projective constraints**.

These projective constraints are natural generalization of the epipolar constraint for the 2-frame case. Here we check that in the 2-frame case the above projective constraint indeed gives the epipolar constraint. In the 2-frame case we have:

$$A = \begin{pmatrix} Pg_1 \\ Pg_2g_1 \end{pmatrix}.$$

If $g_1 \neq 0$, define $Q' = g_1 Q$ and $A' = Ag_1^{-1}$. Thus, without loss of generality, we may assume $g_1 = I$. We then have:

$$A = \begin{pmatrix} I & 0 \\ R_2 & p_2 \end{pmatrix}.$$

The wedge product constraint gives:

$$a_1 \wedge \ldots \wedge a_4 \wedge \tilde{\mathbf{x}}_1 \wedge \tilde{\mathbf{x}}_2 = \det \begin{pmatrix} I & 0 & \mathbf{x}_1 & 0 \\ R_2 & p_2 & 0 & \mathbf{x}_2 \end{pmatrix} e_1 \wedge \ldots \wedge e_6 = 0.$$

Then we have:

$$\begin{aligned} & \det \begin{pmatrix} I & 0 & \mathbf{x}_1 & 0 \\ R_2 & p_2 & 0 & \mathbf{x}_2 \end{pmatrix} = 0 \quad \Leftrightarrow \quad \det \begin{pmatrix} I & 0 & \mathbf{x}_1 & 0 \\ 0 & p_2 & -R_2\mathbf{x}_1 & \mathbf{x}_2 \end{pmatrix} = 0 \\ \Leftrightarrow \quad & \det\left(p_2, -R_2\mathbf{x}_1, \mathbf{x}_2\right) = 0 \quad \Leftrightarrow \quad \mathbf{x}_1^T R_2^T \hat{p}_2 \mathbf{x}_2 = 0. \end{aligned}$$

Thus the projective constraint is exactly the same as the epipolar constraint in the 2-frame case. Also, as pointed out by Triggs and Faugeras *et al* [28, 6] that the quadrilinear constraints are not independent of the bilinear and trilinear ones. Thus, we do not need to consider quadri-linear constraints. The bilinear and trilinear constraints then give a complete set of constraints for the m images of a point. Notice that all the projective constraints only depend on the entries of the motion matrix A, not the scale matrix Λ at all. Given the matrix of image points $\mathbf{X} = (\mathbf{x}_i^j)$, one can use these projective constraints to recover the unknown motion parameters (entries in the motion matrix A) up to a scale in the translational motion.

A big advantage of using the projective constraints is that motion is decoupled from structure. Hence they can be used for motion estimation without knowing structure. For example, in the 2-frame case, one can use the epipolar constraint to recover the relative motion g between the two image frames (with p up to a scale) if one has eight pairs of image points: $\{(\mathbf{x}_1^j, \mathbf{x}_2^j)\}_{j=1}^8$.

However, since projective constraint is invariant under projective transformation, it loses information about the (Euclidean) metric. Therefore, they cannot give a unique recovery of the translational motion nor the 3D structure. The recovery of structure can be done only up to an arbitrary projective transformation (Triggs [29] and Hartley [8]). In the next section, we will study invariant constraints in the Euclidean space under the Euclidean transformation, which will later be used in algorithms for reconstructing the scale information.

Although the bilinear and trilinear constraints together give a complete set of constraints, for motion recovery purpose, the bilinear epipolar constraints are already sufficient. The bilinear epipolar constraints are also better understood and easier to use than the trilinear ones. Motion estimation schemes based on epipolar constraints have been well established. Since most of the motion estimation schemes using bilinear constraints only need (eight) image correspondences between two image frames while those using trilinear ones require (seven) image correspondences among three image frames [22, 7], algorithms based on bilinear constraints alone will be faster and easier to implement. Therefore, **in this paper, we only study the theory of reconstruction based on using the bilinear epipolar constraints only.** For a study and usage of the trilinear constraints for reconstruction purposes, one may refer to Shashua [22] and Hartley [7]. As a summary of the above discussions, we write down all the bilinear **epipolar constraints** we have on the image matrix $\mathbf{X} = (\mathbf{x}_i^j)$ of the m-frame n-point problem:

$$\mathbf{x}_i^{jT} R_{ki}^T \hat{p}_{ki} \mathbf{x}_k^j = 0, \quad 1 \leq j \leq n,\ 1 \leq i < k \leq m, \tag{15}$$

where $g_{ki} \sim (R_{ki}, p_{ki})$ represents the relative motion between the k^{th} and i^{th} frames.

Given $n \geq 8$ image correspondences in general positions, the relative motion between the k^{th} and i^{th} image frames can be determined by solving the **MMSE problem:**

$$\min_{R_{ki} \in SO(3), p_{ki} \in S^2} V(R_{ki}, p_{ki}) = \sum_{j=1}^{n} (\mathbf{x}_i^{jT} R_{ki}^T \hat{p}_{ki} \mathbf{x}_k^j)^2$$

Using the linear algorithm (Toscani or Faugeras [27]) or the nonlinear algorithm (Ma, Košecká and Sastry [16]), (R_{ki}, p_{ki}) can be recovered with p_{ki} up to a scale. From now on, unless otherwise stated, **we assume points considered are always in general position such that the motion recovery algorithms give a unique solution**[1]. For a study of a degenerate case when the 3D points are co-planar, one may refer to Faugeras [5].

3 Euclidean Constraints and Structure Reconstruction

From the preceding section, we see that the projective constraints do not depend on the scale $\Lambda = (\lambda_i^j)$ of the structure nor the scale of the translational motion p. In order to recover the

[1]The motion estimation algorithm typically determines (R_{ki}, p_{ki}) up to four ambiguous solutions. Imposing the positive depth constraint, there is only one compatible with the images.

scale information, one has to work in a metric space and exploit constraints associated to the metric. In this section, we study all the Euclidean invariants and constraints for the n-point m-frame problem under a unified framework. In this framework, one will see the reconstruction of the structural scales and the (translational) motion scales are essentially the same problem. Conditions for unique reconstruction of the structure and motion will be derived.

3.1 Euclidean Invariants and Constraints

For an n-point and m-frame problem, the n points $\{q^j\}_{j=1}^n$ and the m optical centers $\{o^i\}_{i=1}^m$ determines the overall Euclidean configuration. Any Euclidean invariant relates to the relative positions of $n+m$ points as we will explain below.

From the motion recovery algorithms, the relative translational motion $p_{ki}, 1 \leq i, k \leq m$ among the m image frames is only determined up to a scale. Denote the scale associated to p_{ki} as $\gamma_{ki} \in \mathbb{R}^+$ for all $1 \leq i, k \leq m$. Knowing the scale γ_{ki}, then $g_{ki} \sim (R_{ki}, \gamma_{ki} p_{ki})$ gives the true Euclidean transformation from i^{th} frame to the k^{th} frame. In particular, when $k = i$, the transformation is just $(I, 0)$ hence $\gamma_{ii} = 0$ for all $1 \leq i \leq m$.

Notice that each vector p_{ki} can be regarded as the "image" of the i^{th} optical center o^i with respect to the k^{th} image frame. We may identify the m optical centers $\{o^i\}_{i=1}^m$ as m new 3D points $q^{n+i} = o^i$ for $1 \leq i \leq m$. Then the images of these m points with respect to the m image frames are just $\mathbf{x}_k^{n+i} = p_{ki}, 1 \leq i, k, \leq m$. The scale of these image points are denoted by $\lambda_k^{n+i}, 1 \leq i, k \leq m$. These scales are exactly the same as the scales of translational motions $\gamma_{ki}, 1 \leq i, k \leq m$. Then for the m optical centers we have two equivalent representations:

$$
\begin{array}{rcll}
o^i & \longleftrightarrow & q^{n+i}, & 1 \leq i \leq m, \\
p_{ki} & \longleftrightarrow & \mathbf{x}_k^{n+i}, & 1 \leq i, k \leq m, \\
\gamma_{ki} & \longleftrightarrow & \lambda_k^{n+i}, & 1 \leq i, k \leq m.
\end{array}
$$

The left hand side is associated to the notion of motion and the right hand side is associated to the notion of structure.

For all the $n+m$ points $\{q^j\}_{j=1}^{n+m}$, we have the **extended image matrix** $\mathbb{X} \in \mathbb{R}^{3m \times (n+m)}$:

$$
\mathbb{X} = \left(\mathbf{x}_i^j\right)_{i=1,j=1}^{m,n+m} = \begin{pmatrix}
\mathbf{x}_1^1 & \mathbf{x}_1^2 & \cdots & \mathbf{x}_1^n & 0 & p_{12} & \cdots & p_{1m} \\
\mathbf{x}_2^1 & \mathbf{x}_2^2 & \cdots & \mathbf{x}_2^n & p_{21} & 0 & \cdots & p_{2m} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
\mathbf{x}_m^1 & \mathbf{x}_m^2 & \cdots & \mathbf{x}_m^n & p_{m1} & p_{m2} & \cdots & 0
\end{pmatrix}.
$$

Then the reconstruction is to determine all the scales in the **extended scaled image matrix** $\mathbb{Y} \in \mathbb{R}^{3m \times (n+m)}$:

$$
\mathbb{Y} = \left(\lambda_i^j \mathbf{x}_i^j\right)_{i=1,j=1}^{m,n+m} = \begin{pmatrix}
\lambda_1^1 \mathbf{x}_1^1 & \lambda_1^2 \mathbf{x}_1^2 & \cdots & \lambda_1^n \mathbf{x}_1^n & 0 & \gamma_{12} p_{12} & \cdots & \gamma_{1m} p_{1m} \\
\lambda_2^1 \mathbf{x}_2^1 & \lambda_2^2 \mathbf{x}_2^2 & \cdots & \lambda_2^n \mathbf{x}_2^n & \gamma_{21} p_{21} & 0 & \cdots & \gamma_{2m} p_{2m} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
\lambda_m^1 \mathbf{x}_m^1 & \lambda_m^2 \mathbf{x}_m^2 & \cdots & \lambda_m^n \mathbf{x}_m^n & \gamma_{m1} p_{m1} & \gamma_{m2} p_{m2} & \cdots & 0
\end{pmatrix}.
$$

Without loss of generality, we can always choose $\|p_{ki}\| = \|p_{ik}\|$ hence $\gamma_{ki} = \gamma_{ik}$. The i^{th} "row" $(\lambda_i^j \mathbf{x}_i^j)_{j=1}^{n+m}$ of the extended scaled image matrix $\mathbb{Y}$ gives the 3D coordinates of the $n+m$ points

$\{q^j\}_{j=1}^{n+m}$ with respect to the i^{th} image frame. Each such a row reconstructs the 3D structure of the overall n-point m-frame configuration up to a Euclidean transformation.

All the Euclidean invariants associated to the m-frame n-point problem are invariants related to the quantities in the extended scaled image matrix $\mathbb{Y}$. These invariants give constraints on the unknown scales $\{\lambda_i^j\}_{i=1,j=1}^{m,n+m}$. The scales can then be reconstructed by solving the equations given by these constraints. Notice that the special Euclidean group $SE(3)$ is a semi-direct product of $SO(3)$ and $\mathbb{R}^3$. The subgroup $\mathbb{R}^3$ is a normal subgroup of $SE(3)$ and acts transitively on $\mathbb{R}^3$ itself. The basic invariants of $\mathbb{R}^3$ preserved under the action on itself is just the **vector**, *i.e.* the difference between the coordinates of two points in $\mathbb{R}^3$. Therefore, the invariants of $\mathbb{R}^3$ under the full group action $SE(3)$ are the invariants of the space of 3 dimensional vectors under the group action $SO(3)$.

The difference between the j^{th} and l^{th} columns of the matrix $\mathbb{Y}$ are the representations of the vector $q^j - q^l$ with respect to the m image frames. From the above discussion, all the intrinsic invariants of the n-point m-frame problem are just the invariants associated to the $(n+m)(n+m-1)$ vectors among these $n+m$ points. Different representations of the same vector with respect to different image frames only differ by orthogonal transformations, *i.e.* elements in $SO(3)$. The following theorem, Theorem 2.9 from Weyl [31], characterizes all the invariants of orthogonal group $SO(n)$.

Theorem 2 (Invariants of $SO(n)$) *For a n dimensional real vector space, a complete table of typical basic invariants of the orthogonal group consists of (1) the inner product $< u, v >$ and (2) the determinant of n vectors* $\det[u^1, \dots, u^n]$.

According to this theorem, the set of all invariants is just the algebra generated by these two types of basic invariants. In the 3 dimensional case, life is easier. The two basic types of invariants are just: the inner product between two vectors and the determinant of three vectors. We then have:

Corollary 1 *The set of all intrinsic invariants associated to the n-point m-frame problem is the $\mathbb{R}$-algebra generated by all the inner products between two vectors and determinants of three vectors associated with the $n+m$ points $\{q^j\}_{j=1}^n$ and $\{o^i\}_{i=1}^m$.*

In general, $SE(3)$ is just a subgroup of $GL(4)$, hence the set of invariants of $GL(4)$ is a subset of that of $SE(3)$. In this sense, for the n-point m-frame problem, the set of all projective invariants (constraints) is just a subset of all the Euclidean invariants (constraints). The study of projective invariants is important, since it provides some constraints relevant to a particular problem: for example, the motion recovery problem is elegantly decoupled from the structure by considering the projective constraints only. On the other hand, projective constraints are not adequate for structure reconstruction and Euclidean constraints need also to be exploited.

Notice all the constraints obtained by using these intrinsic Euclidean invariants will be either **quadratic** (from inner products) or **cubic** (from determinants) in the unknown scales λ's and γ's. That is, there are **no intrinsic Euclidean constraints which are linear in the scales.**

In order to obtain **linear** constraints, one has to explicitly exploit the relative motions between image frames. The obtained constraints will be called **extrinsic Euclidean constraints.** For example, for $1 \leq i, k \leq m, 1 \leq j \leq n$, the relation between the representations of the vector $q^j - o^i$

in the i^{th} and k^{th} image frames are given by:

$$\begin{aligned} & g_{ki*}(q^j - o^i) = g_{ki}(q^j) - g_{ki}(o^i) \quad \Leftrightarrow \quad R_{ki}(\lambda_i^j \mathbf{x}_i^j - 0) = \lambda_k^j \mathbf{x}_k^j - \lambda_k^{n+i} \mathbf{x}_k^{n+i} \\ \Leftrightarrow \quad & R_{ki}(\lambda_i^j \mathbf{x}_i^j - 0) = \lambda_k^j \mathbf{x}_k^j - \gamma_{ki} p_{ki} \quad \Leftrightarrow \quad \lambda_k^j \mathbf{x}_k^j = R_{ki} \lambda_i^j \mathbf{x}_i^j + \gamma_{ki} p_{ki}. \end{aligned} \tag{16}$$

This is just the Euclidean transformation of the $3D$ coordinates of the point q^j between the i^{th} and k^{th} frames. This constraint explicitly relies on the knowledge of R_{ki} and p_{ki} (up to a scale).

From the above discussion, we have the following observations:

1. The problem of recovering the scales of motion is essentially the same as one of reconstructing structural scales. The scales of motion and structure could (and should) be reconstructed in a uniform framework.
2. There are only **quadratic** and **cubic** intrinsic constraints on scales. To obtain **linear** constraints, one has to explicitly rely on knowledge of relative motion.
3. Although the general theory of invariants gives all the intrinsic invariants/constraints in the scales, it does not directly give the conditions for uniqueness of the scale reconstruction.
4. There is redundancy in the set of all Euclidean constraints, as in the projective case. We will want to pick a convenient but sufficient subset for reconstructing the scales.

These issues will be addressed in the following sections.

3.2 Conditions for a Unique Reconstruction

In this section, we study the problem: given the relative motions $(R_{ki}, p_{ki}), 1 \leq i, k \leq m$ with p_{ki}'s defined up to a scale, what are the conditions for the n-point m-frame problem to have a unique reconstruction, *i.e.* a unique scale matrix Λ compatible with the given image matrix $\mathbf{X}$. Notice that the equation (8) is homogeneous, the scale matrix Λ can only be recovered up to a **universal scale.** We then treat two scale matrices $\Lambda' = \sigma\Lambda, \sigma \in \mathbb{R}^+$ as equivalent; and we will talk about the uniqueness modulo this equivalence.

Definition 2 *A configuration of the n-point m-frame problem is called* **critical** *if any of the vectors* $o^i - o^k, 1 \leq i < k \leq m$, *i.e. the epipoles, lines up with any vector* $q^j - o^k, 1 \leq k \leq m, 1 \leq j \leq n$.

Notice that, following this definition, zero translation always induces a critical configuration. The set of all critical configurations is a zero-measure set of the overall configuration space; the set of non-critical configurations is dense in the overall configuration space. For a non-critical configuration, the two vectors p_{ki} and $\mathbf{x}_k^j$ are linearly independent for all $1 \leq i, k \leq m, 1 \leq j \leq n$. In this case, the structure can be reconstructed uniquely.

Theorem 3 (Sufficient Conditions for a Unique Reconstruction) *Consider a non-critical n-point m-frame problem. Given the relative motions* $(R_{ki}, p_{ki}), 1 \leq i < k \leq m$ *with* p_{ki}*'s defined up to a scale, for the image matrix* $\mathbf{X} \in \mathbb{R}^{3m \times n}$ *of these n points in the m image frames, its compatible scale matrix* $\Lambda \in \mathbb{R}^{m \times n}$ *is unique up to a scale.*

Proof: Suppose Λ' is another compatible scale matrix. Re-scale Λ' by λ_1^1/λ'^1_1. For this scale matrix Λ', it suffices to prove that $\lambda_k^j = \lambda'^j_k$ for all $1 \le j \le n$ and $1 \le k \le m$. From the constraints (16), for all $2 \le k \le m, 2 \le j \le n$ we have:

$$R_{k1}(\lambda_1^1 \mathbf{x}_1^1 - \lambda_1^j \mathbf{x}_1^j) = \lambda_k^1 \mathbf{x}_k^1 - \lambda_k^j \mathbf{x}_k^j$$
$$R_{k1}(\lambda'^1_1 \mathbf{x}_1^1 - \lambda'^j_1 \mathbf{x}_1^j) = \lambda'^1_k \mathbf{x}_k^1 - \lambda'^j_k \mathbf{x}_k^j$$

After the rescaling, $\lambda_1^1 = \lambda'^1_1$. Subtracting these two equations, we have:

$$(\lambda'^j_1 - \lambda_1^j) R_{k1} \mathbf{x}_1^j = (\lambda_k^1 - \lambda'^1_k)\mathbf{x}_k^1 - (\lambda_k^j - \lambda'^j_k)\mathbf{x}_k^j. \tag{17}$$

For the constraint (16), when $i = 1$, applying cross product with p_{k1} to both sides, it yields:

$$\lambda_k^j \mathbf{x}_k^j \times p_{k1} = \lambda_1^j R_{ki} \mathbf{x}_1^j \times p_{k1},$$
$$\lambda'^j_k \mathbf{x}_k^j \times p_{k1} = \lambda'^j_1 R_{ki} \mathbf{x}_1^j \times p_{k1}.$$

Apply cross product with p_{k1} to both sides of (17) and use the above identities we obtain:

$$(\lambda_k^1 - \lambda'^1_k)(\mathbf{x}_k^1 \times p_{k1}) = 0.$$

Since $\mathbf{x}_k^1 \times p_{k1} \neq 0$ because the configuration is non-critical, we have $\lambda_k^1 - \lambda'^1_k = 0$ and (17) becomes:

$$(\lambda'^j_1 - \lambda_1^j) R_{k1} \mathbf{x}_1^j = (\lambda'^j_k - \lambda_k^j)\mathbf{x}_k^j.$$

By the assumption of non-criticality, $\mathbf{x}_k^j$ and p_{k1} are linearly independent. Since $\lambda_k^j \mathbf{x}_k^j = \lambda_1^j R_{k1} \mathbf{x}_1^j + \gamma_k^1 p_{k1}$ we have $R_{k1}\mathbf{x}_1^j$ and $\mathbf{x}_k^j$ are linearly independent. Therefore, $\lambda'^j_k - \lambda_k^j = 0$ and $\lambda'^j_1 - \lambda_1^j = 0$. Combining with the previous result that $\lambda_k^1 - \lambda'^1_k = 0$, we have

$$\lambda'^j_k = \lambda_k^j, \quad 1 \le k \le m, 1 \le j \le n.$$

That is $\Lambda' = \Lambda$. ■

The scale matrix Λ is thus recovered up to a universal (positive) scale, say $\sigma \in \mathbb{R}^+$. With respect to this scale, the scale of the relative translational motion is uniquely recovered through (16):

$$\gamma_k^i p_{ki} = \lambda_k^j \mathbf{x}_k^j - \lambda_i^j R_{ki} \mathbf{x}_i^j. \tag{18}$$

Corollary 2 *Consider the setup with assumptions of Theorem 3. The structural scales of the n points and the scale of relative (translational) motions among the m frames are uniquely determined up to a universal scale.*

Combined with the motion estimation algorithms, Theorem 3 and Corollary 2 tell us that as long as eight points and two image frames form a non-critical configuration, the compatible scales of these points are uniquely determined, as is the scale of the relative motion between these two frames.

Theorem 3 gives a sufficient condition for the n-point m-frame problem to have a unique solution. This condition may be too restrictive in practice. We need to study the necessary and sufficient conditions for the uniqueness. For a general n-point m-image configuration, it is reasonable to

assume that for any point $q^j, 1 \leq j \leq n$ there are at least two image frames in which q^j is not critical. Otherwise, the camera simply translates along the straight line determined by the optical center and this point. There is no way to tell the distance of such a point. A point q which does not satisfy this assumption is called a **singular point** with respect to the m image frames considered.

Definition 3 *Two frames are called* **strongly connected** *if they both have images for at least eight points in general positions which form a non-critical configuration with the two frames.*

Definition 4 Connectedness *is the smallest equivalence relation generated on the m frames by the strong connectedness.*

In the language of group theory, if the i^{th} and k^{th} frames are strongly connected, we treat it as a 2-cycle (k, i) in the permutation group S_m of the m image frames as m elements. Then the connectedness relation is represented by the subgroup of S_m generated by all the 2-cycles that represent strong connectednesses. For an instance, there is only one equivalence class if and only if this subgroup acts on m elements transitively.

Theorem 4 (Necessary and Sufficient Condition for Unique Reconstruction) *For an n-point m-image problem in a general configuration, the structure and the relative motion are determined up to a universal scale if and only if the n points are non-singular and the m frames are in a single equivalence class with respect to the connectedness.*

Proof: The necessity is obvious since the relative motion between different equivalence classes cannot be uniquely determined by the motion recovery algorithms (yet the 3D structure of the n points can still possibly be reconstructed up to an arbitrary Euclidean transformation), and singular points have to be ruled out. We prove the sufficiency here. First use all strongly connected relations to determine the relative motions, and then determine the structural scales of all the associated image points and the scales of the translational motions between strongly connected frames. The relative motion g_{ki} between any two frames is then uniquely determined by those known from the strong connectednesses. Now only the scales of points which are not in any of the strongly connected relations are not yet determined. Say the scale of point q^j is not yet determined. Since it is non-singular, there exist two frames, say i^{th} and k^{th} frames in which q^j is not critical. Then the vectors p_{ki} and $\mathbf{x}_k^j$ are linearly independent. The equation:

$$\lambda_k^j \mathbf{x}_k^j = \lambda_i^j R_{ki} \mathbf{x}_i^j + \gamma_{ki} p_{ki}$$

uniquely determines the two scale λ_k^j and λ_i^j since γ_k^i is already known now. Then the scales of point q^j with respect to frame i and k are reconstructed, so are the scales with respect to other frames. In fact, if a non-singular point has images in all of the m frames, it must be in one of the strongly connected relations we have used above to determine g_{ki}. ■

The proof also implies that we do not have to require that every point has images in all m image frames. Occlusion is allowed as long as the point is non-singular in the frames in which it has images. We thus have:

Theorem 5 (Reconstruction with Occlusion) *For an n-point m-image problem with occlusion, the structure and the relative motion are recoverable up to a universal scale if and only if the*

n points are non-singular in the frames in which they have images and the m frames are in a single equivalence class with respect to the connectedness.

Then one needs at least two (non-singular) images of a point q to reconstruct its (relative) 3D scale. This theorem also implies that in the n-point m-frame setting, a lot of the image data is actually redundant for determining the structural scales. One application of this theorem is that one may reconstruct the 3D structure of an opaque object by taking pictures from all over the places without knowing the position of the camera and worrying about the occlusion. The uniqueness is guaranteed as long as connectedness between all the pictures is satisfied.

Theorem 3, 4 and 5 establish the basic theory of Euclidean structure and motion reconstruction. In the absence of noise, they guarantee a unique solution of the motion matrix A and the scale matrix Λ from the image $\mathbf{X}$. In practice however, the image measurements $\mathbf{X}$ are always noisy, due to quantization errors, thermal noise in the CCD array, or the errors in image correspondences. In the next section, we will develop efficient algorithms for estimating the scale and motion matrices from a noisy image matrix.

4 Algorithms

In this section, we study numerical algorithms for estimating the motion and scale matrices from possibly noisy image matrices. For a given noisy image matrix $\mathbf{X}$, it is possible that there is no scale matrix Λ which is exactly compatible with it. In this case, we choose to pick an estimate of $\Lambda \in \mathbb{R}^{m\times n}$ which minimizes the mean squared error. We here propose several algorithms derived from the reconstruction theory developed in the preceding sections.

4.1 A Preliminary Algorithm

For the n-point m-frame problem, consider the scaled image matrix:

$$\mathbf{Y} = \begin{pmatrix} \lambda_1^1\mathbf{x}_1^1 & \lambda_1^2\mathbf{x}_1^2 & \cdots & \lambda_1^n\mathbf{x}_1^n \\ \lambda_2^1\mathbf{x}_2^1 & \lambda_2^2\mathbf{x}_2^2 & \cdots & \lambda_2^n\mathbf{x}_2^n \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_m^1\mathbf{x}_m^1 & \lambda_m^2\mathbf{x}_m^2 & \cdots & \lambda_m^n\mathbf{x}_m^n \end{pmatrix} \in \mathbb{R}^{3m\times n}.$$

The **preliminary algorithm works for the case that the configuration between any two consecutive frames (rows of the matrix Y) is not critical.** The reconstruction can then be done by recovering relative motions and scales only between pairs of consecutive frames (rows of $\mathbf{Y}$).

For $2 \le i \le m$, suppose that the relative motion between the i^{th} and the $(i-1)^{th}$ frames is (R_i, p_i) with p_i defined up to a unknown scale γ_i (note that R_i, p_i and γ_i are shorthand for $R_{i,i-1}, p_{i,i-1}$ and $\gamma_{i,i-1}$ respectively), then from (16), we have the **row scale constraints:**

$$\begin{aligned} &\lambda_i^j\mathbf{x}_i^j = R_i\lambda_{i-1}^j\mathbf{x}_{i-1}^j + \gamma_i p_i \quad \Leftrightarrow \quad \lambda_i^j\mathbf{x}_i^j \times p_i = \lambda_{i-1}^j(R_i\mathbf{x}_{i-1}^j) \times p_i \\ \Leftrightarrow \quad &\lambda_{i-1}^j\hat{p}_i R_i\mathbf{x}_{i-1}^j - \lambda_i^j\hat{p}_i\mathbf{x}_i^j = 0. \end{aligned}$$

Obviously row scale constraints determine relative scales $\lambda_i^j/\lambda_{i-1}^j, 2 \le i \le m, 1 \le j \le n$ between rows of the scale matrix Λ.

Also from (16), we have the **column scale constraints:**

$$\begin{aligned} & \lambda_i^j \mathbf{x}_i^j - R_i \lambda_{i-1}^j \mathbf{x}_{i-1}^j = \lambda_i^{j+1} \mathbf{x}_i^{j+1} - R_i \lambda_{i-1}^{j+1} \mathbf{x}_{i-1}^{j+1} = \gamma_i^{i-1} p_i \\ \Leftrightarrow\; & \lambda_{i-1}^j \left[(\lambda_i^j / \lambda_{i-1}^j) \mathbf{x}_i^j - R_i \mathbf{x}_{i-1}^j \right] = \lambda_{i-1}^{j+1} \left[(\lambda_i^{j+1} / \lambda_{i-1}^{j+1}) \mathbf{x}_i^{j+1} - R_i \mathbf{x}_{i-1}^{j+1} \right] \end{aligned}$$

Then knowing the relative row scales $\lambda_i^j/\lambda_{i-1}^j$ and $\lambda_i^{j+1}/\lambda_{i-1}^{j+1}$, the column scale constraints determine the relative scales $\lambda_{i-1}^{j+1}/\lambda_{i-1}^j, 2 \leq i \leq m, 1 \leq j \leq n-1$ between columns of Λ. Notice the column scale constraints are equivalent to:

$$\lambda_i^j \mathbf{x}_i^j - \lambda_i^{j+1} \mathbf{x}_i^{j+1} = R_i \left(\lambda_{i-1}^j \mathbf{x}_{i-1}^j - \lambda_{i-1}^{j+1} \mathbf{x}_{i-1}^{j+1} \right).$$

The left hand side and the right hand side are the expressions for the same vector $q^j - q^{j+1}$ with respect to the $(i-1)^{th}$ frame and the i^{th} frame, respectively. The row and column constraints are geometrically interpreted in Figure 2.

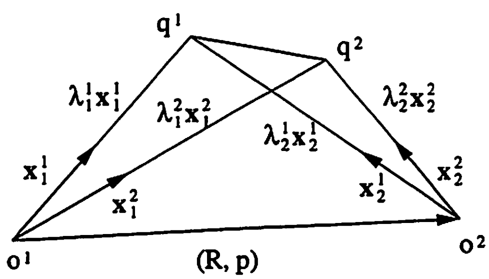

Figure 2: Geometric interpretation of row and column constraints: the row constraint (for the point q^1) is that the cross products $(o^2 - o^1) \times (q^1 - o^1)$ and $(o^2 - o^1) \times (q^1 - o^2)$ give the same vector with the length proportional to the area of the triangle formed by (q^1, o^1, o^2); the column constraint (for the points q^1 and q^2) is essentially the fact that the length of the vector $q^1 - q^2$ is preserved under the Euclidean transformation $(R, p) \in SE(3)$ *i.e.* different expressions for the vector $q^1 - q^2$ with respect to different frames only differ by the relative rotation $R \in SO(3)$ between the frames.

For a given (noisy) image matrix $\mathbf{X} = (\mathbf{x}_i^j) \in \mathbb{R}^{3m \times n}, 1 \leq i \leq m, 1 \leq j \leq n$, the Preliminary Algorithm goes as follows:

Preliminary Algorithm:

- Compute the relative motion (R_i, p_i) between the $(i-1)^{th}$ and i^{th} frames for $2 \leq i \leq m$ by solving the MMSE estimation problem:

$$\min_{R_i \in SO(3), p_i \in S^2} V(R_i, p_i) = \sum_{j=1}^{n} (\mathbf{x}_{i-1}^{jT} R_i^T \hat{p}_i \mathbf{x}_i^j)^2$$

  This can be solved by the nonlinear optimization algorithm on a Stiefel manifold $SO(3) \times S^2$ (Ma, Košecká and Sastry [16]) or by the linear algorithm (Toscani and Faugeras [27]). Imposing the positive depth constraint, pick the only pair (R_i, p_i) which matches the image data from four ambiguous solutions given by the foregoing motion recovery algorithms.

- Use the **row scale constraints:**

$$\lambda_{i-1}^j \hat{p}_i R_i \mathbf{x}_{i-1}^j - \lambda_i^j \hat{p}_i \mathbf{x}_i^j = 0, \quad 2 \leq i \leq m, 1 \leq j \leq n$$

to solve for all relative row scales $\lambda_i^j/\lambda_{i-1}^j, 2 \le i \le m, 1 \le j \le n$ between consecutive rows of the scale matrix Λ. They are well determined by the constraints since $\hat{p}_i \mathbf{x}_i^j \neq 0$ because the configuration is not critical.

- Use the **column scale constraints:**

$$\lambda_{i-1}^j \left[(\lambda_i^j/\lambda_{i-1}^j)\mathbf{x}_i^j - R_i \mathbf{x}_{i-1}^j \right] = \lambda_{i-1}^{j+1} \left[(\lambda_i^{j+1}/\lambda_{i-1}^{j+1})\mathbf{x}_i^{j+1} - R_i \mathbf{x}_{i-1}^{j+1} \right]$$

  to calculate the relative column scales $\lambda_{i-1}^{j+1}/\lambda_{i-1}^j, 2 \le i \le m, 1 \le j \le n-1$ of the scale matrix Λ. This is well determined since the translation is always nonzero. We then have m estimates for each relative column scale λ^{j+1}/λ^j. They should be the same in the absence of noise. A reasonable estimate of the relative column scale between the $(j+1)^{th}$ and j^{th} column is the mean:

$$\lambda^{j+1}/\lambda^j = \frac{1}{m} \sum_{i=1}^{m} \left(\lambda_i^{j+1}/\lambda_i^j \right), \quad 1 \le j \le n-1.$$

- Set $\lambda_1^1 = 1$ and then the relative row and column scales calculated above uniquely determine the scale matrix Λ.

- The translational motion $\gamma_i p_i$ is re-estimated by the mean:

$$\gamma_i p_i = \frac{1}{n} \sum_{j=1}^{n} \left(\lambda_i^j \mathbf{x}_i^j - \lambda_{i-1}^j R_i \mathbf{x}_{i-1}^j \right), \quad 1 \le i \le m.$$

- Recover the motion matrix A from $(R_i, p_i), 1 \le i \le m$, the scale image matrix $\mathbf{Y}$ from $\mathbf{X}$ and Λ. Then any row of $\mathbf{Y}$ reconstruct the 3D structure of the n points q^j up to a Euclidean motion and an overall scale.

Although the preliminary algorithm is conceptually simple, it has some major drawbacks:

1. it does not apply to the cases where the non-criticality between consecutive frames is violated;
2. (relative) structural scales are estimated locally and the overall estimation may not be globally optimal;
3. the structural scales and translational motion scales are estimated separately; this is not in the spirit of our former observations, which reveal that the reconstruction of structural scales and translational motion scales is essentially the same thing.

For example, difficulties occur when two consecutive rows form a critical configuration but the overall configuration satisfies conditions given in Theorem 4. Also, when a point is singular with respect to two consecutive frames, the corresponding row scale constraint is ill-conditioned. Further, different rows of $\mathbf{Y}$ may give inconsistent reconstructions of the 3D structure of the n points. We thus need a version of **triangulation** (Hartley [9]) for the m-frame case, *i.e.* a consistent reconstruction of the 3D structure using information from all m images.

4.2 Main Algorithm

The main algorithm to be proposed here is to estimate the structural scales and (translational) motion scales altogether. Further more, we will consider triangulation among the m image frames so as to obtain a consistent reconstruction.

To simplify the notation, we first assume the configuration between two consecutive rows is non-critical and we will drop this requirement later on. Notice all the row and column constraints used in the Preliminary Algorithm essentially come from the same constraints (16):

$$\lambda_i^j \mathbf{x}_i^j = \lambda_{i-1}^j R_i \mathbf{x}_{i-1}^j + \gamma_i p_i, \quad 2 \leq i \leq m, 1 \leq j \leq n \tag{19}$$

In the absence of noise, **the preliminary algorithm actually constructively proved that the row and column constraints uniquely determine all the unknown scales up to a universal scale, as do the constraints given by the above equations (19).**

Notice that knowing $\{(R_i, p_i)\}_{i=2}^m$, the equations given by (19) are linear in both the structural scales λ's and the motion scales γ's. The estimation of these scales can be formulated as a standard **LLSE estimation problem.** Arrange all the unknown scales in (19) into an **extended scale vector** $\vec{\lambda}$:

$$\vec{\lambda} = (\lambda_1^1, \ldots, \lambda_1^n, \lambda_2^1, \ldots, \lambda_2^n, \ldots, \lambda_m^1, \ldots, \lambda_m^n, \gamma_2, \ldots, \gamma_m)^T \in \mathbb{R}^{mn+m-1}.$$

Then all the constraints given by (19) can be expressed in a single linear equation:

$$M\vec{\lambda} = 0 \tag{20}$$

where M is a matrix depending on $\{(R_i, p_i)\}_{i=2}^m$ and $\{\mathbf{x}_i^j\}_{i=1,j=1}^{m,n}$. There are totally $3n(m-1)$ (scalar) linear equations given by (19) hence M is a $3n(m-1) \times (mn+m-1)$. Since $3n(m-1) - (mn+m-1) = (2n-1)(m-1) - n$, as long as $m, n \geq 2$, we have more equations than unknowns. Since in our case $n \geq 8, m \geq 2$, the problem of solving $\vec{\lambda}$ from (20) is usually over-determined. For (20) to have a unique solution, the matrix M needs to have rank $mn + m - 2$.

In the absence of noise, if consecutive pairs of rows are non-critical, the equations given by (19) uniquely determine all the unknown scales. In this case, the matrix M should have exactly rank $mn + m - 2$ and the linear equation (20) has a unique (up to scale) solution for $\vec{\lambda}$ (as previously pointed out, the preliminary algorithm in fact gives a constructive proof for the uniqueness). In the presence of noise, the LLSE estimate of $\vec{\lambda}$ is just the eigenvector of $M^T M$ corresponding to the smallest eigenvalue. However, the obtained reconstruction $\mathbf{Y}$ may still suffer from inconsistency. In order to **triangulate** among the m images, we convert the m images of each point $q^j, 1 \leq j \leq n$ to the same image frame, say the last or the m^{th} image frame by the Euclidean transformations:

$$\lambda_i^j \mathbf{x}_i^j \mapsto \mathbf{z}_i^j = R_{mi} \lambda_i^j \mathbf{x}_i^j + \gamma_{mi} p_{mi}, \quad 1 \leq i \leq m, 1 \leq j \leq n.$$

Then $\mathbf{z}_i^j, 1 \leq i \leq m, 1 \leq j \leq n$ is the **transformed image** of the j^{th} point q^j from the i^{th} image frame to the m^{th} image frame. However, the relative motions $(R_{mi}, p_{mi}), 1 \leq i \leq m$ are usually not estimated directly. They may be given in terms of relative motions between consecutive frames by:

$$R_{mi} = \prod_{k=i}^{m-1} R_{k+1}, \quad \gamma_{mi} p_{mi} = \sum_{k=i}^{m-1} R_{m,k+1} \gamma_{k+1} p_{k+1} = \sum_{k=i}^{m-1} \gamma_{k+1} R_{m,k+1} p_{k+1}.$$

Notice the expression for the term $\gamma_{mi}p_{mi}, 1 \leq i \leq m-1$ is linear in $\gamma_k, i+1 \leq k \leq m$. With respect to the m^{th} frame, we now have m copies of 3D coordinates for each point $q^j, 1 \leq j \leq n$:

$$(\mathbf{z}_1^j, \ldots, \mathbf{z}_m^j) = \left(\lambda_1^j R_{m1}\mathbf{x}_1^j + \sum_{k=1}^{m-1} \gamma_{k+1} R_{m,k+1} p_{k+1} \,, \ldots, \, \lambda_m^j \mathbf{x}_m^j \right).$$

In the absence of noise, due to the Euclidean transformation $\lambda_m^j \mathbf{x}_m^j = R_{mi}\lambda_i^j \mathbf{x}_i^j + \gamma_{mi}p_{mi}$, all the m copies of coordinates should be equal to each other:

$$\mathbf{z}_i^j = \mathbf{z}_k^j, \quad 1 \leq i \leq k \leq m, 1 \leq j \leq n. \tag{21}$$

This gives an equivalent set of constraints to those given by (19). Notice that the transformed images $\{\mathbf{z}_i^j\}_{i=1,j=1}^{m,n}$ are still linear in the unknown scales λ's and γ's. The constraints given by (21) are therefore linear equations in terms of the unknown scales.

With respect to the m^{th} frame, a reasonable estimate of the 3D coordinates of the point q^j will be the mean:

$$\bar{\mathbf{z}}^j = \frac{1}{m}\sum_{i=1}^{m} \mathbf{z}_i^j, \quad 1 \leq j \leq n.$$

We then have three equivalent sets of **linear** constraints on scales:

$$\{\lambda_i^j \mathbf{x}_i^j = \lambda_{i-1}^j R_i \mathbf{x}_{i-1}^j + \gamma_i p_i\}_{i=2,j=1}^{m,n} \quad \Leftrightarrow \quad \{\mathbf{z}_i^j = \mathbf{z}_k^j\}_{i=1,k=1,j=1}^{m,m,n} \quad \Leftrightarrow \quad \{\mathbf{z}_i^j = \bar{\mathbf{z}}^j\}_{i=1,j=1}^{m,n}.$$

Since these three sets of constraints are algebraically equivalent, if any one of them uniquely determines the scales, so will the other two. However, in the presence of noise, using different sets of constraints will give different LLSE estimations. If we write all the linear constraints given by $\{\mathbf{z}_i^j - \bar{\mathbf{z}}^j = 0\}_{i=1,j=1}^{m,n}$ in matrix form:

$$\tilde{M}\vec{\lambda} = 0$$

then there exists a matrix W such that $WM = \tilde{M}$. In the presence of noise, M and $\tilde{M}$ are usually not singular. Then the eigenvectors corresponding to the smallest eigenvalues of M^TM and $\tilde{M}^T\tilde{M} = M^TW^TWM$ may not be the same, hence minimization of $\|M\vec{\lambda}\|^2$ and $\|\tilde{M}\vec{\lambda}\|^2$ will give different LLSE estimates of $\vec{\lambda}$.

For the purpose of **triangulating** the m images, it is natural to minimize the following objective function:

$$\min_{\|\vec{\lambda}\|=1} V(\vec{\lambda}) = \sum_{j=1}^{n}\sum_{i=1}^{m}(\mathbf{z}_i^j - \bar{\mathbf{z}}^j)^2 = \|\tilde{M}\vec{\lambda}\|^2.$$

That is to minimize the Euclidean distances from the reconstructed m copies to their mean. Clearly, this problem is also an LLSE problem and can be solved efficiently. Then the mean $\{\bar{\mathbf{z}}^j\}_{j=1}^n$ gives a unbiased reconstruction of the n points $\{q^j\}_{j=1}^n$ (with respect to the m^{th} frame).

Notice that, in the above discussion, the assumption that consecutive pairs of image frames are non-critical is really not necessary. The ideas apply to any configuration which satisfies the uniqueness conditions given in Theorem 4. Below is the summary of the main algorithm.

Main Algorithm: Discrete Case

- For $1 \leq i < k \leq m$, check for the i^{th} and k^{th} frames if the conditions for the eight-point motion recovery algorithms to have a unique solution are satisfied for the available image correspondences. If so, recover the relative motion (R_{ki}, p_{ki}) as in the Preliminary Algorithm. Recover at least $m-1$ (independent) relative motions such that all the m frames is in a single connected class.

- With respect to the last image frame, the m^{th} image frame, solve the LLSE estimation problem

$$\min_{\|\vec{\lambda}\|=1} V(\vec{\lambda}) = \sum_{j=1}^{n}\sum_{i=1}^{m}(\mathbf{z}_i^j - \bar{\mathbf{z}}^j)^2 = \|\tilde{M}\vec{\lambda}\|^2.$$

  for all the available constraints. The solution is unique if and only if the matrix $\tilde{M}^T\tilde{M}$ has a unique smallest eigenvalue.

- Recover the scales of the translational motions and the scale matrix Λ from $\vec{\lambda}$. The scaled image matrix $\mathbf{Y}$ is simply $\Lambda \odot \mathbf{X}$. The triangulated structure estimate (with respect to the m^{th} frame) is given by $\{\bar{\mathbf{z}}^j\}_{j=1}^n$.

In this algorithm, the roles of projective and Euclidean constraints are very clear: the projective constraints recover the motion up to a scale; the Euclidean constraints handle the scale information (both for the structure and the motion).

4.3 Iterative Algorithm

One may have noticed in the above algorithm the close relation between motion and structure. In fact, after obtaining the scaled image matrix $\mathbf{Y}$, one can re-estimate both R_{ki} and p_{ki}, $1 \leq i, k \leq m$. This is an MMSE estimation problem with the linear constraints:

$$\lambda_k^j \mathbf{x}_k^j = R_{ki}\lambda_i^j \mathbf{x}_i^j + p_{ki}, \quad 1 \leq i, k \leq m, 1 \leq j \leq n.$$

and the constraints $R_{ki}^T R_{ki} = I$. It can be solved as an optimization problem on the manifold $SO(3) \times \mathbb{R}^3$. With the newly estimated relative motions, one can run the Main Algorithm again and re-estimate the structural scales and translational motion scales. Keep repeating this procedure until the difference between two consecutive estimates of the structure or the motion is small enough. This is then a Gauss-Newton type iteration scheme and the goal is to search for a global optimal structure and motion estimates. Once it converges, it will certainly give a better estimation than the initial guess. Of course, this approach is computationally costly.

The above algorithm only needs to be slightly modified to work for the **case with occlusion** (not like the factorization method). In a dynamic setting, like real-time vision, these algorithms can also be adjusted to **recursive versions**.

5 Differential Case

The differential case is a limiting case of the discrete case. In this section, we study the differential version of some of the constraints from previous sections. Some of these differential constraints have already been used in computer vision to recover motion or structure.

5.1 Projective Constraints and Motion Recovery

At time $t = t_0$, differentiating the equation (7) $(m-1)$ times, we obtain the equation for higher order derivatives of the optical flow at one point:

$$\begin{pmatrix} \mathbf{x} & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ \dot{\mathbf{x}} & \mathbf{x} & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \cdots & \cdots & \vdots \\ \mathbf{x}^{(i)} & \vdots & c_j^i \mathbf{x}^{(i-j)} & \ddots & \ddots & \cdots & \vdots \\ \vdots & \vdots & \vdots & \cdots & \ddots & \ddots & \vdots \\ \mathbf{x}^{(m-2)} & \cdots & \cdots & \cdots & \cdots & \mathbf{x} & 0 \\ \mathbf{x}^{(m-1)} & \cdots & \cdots & \cdots & \cdots & \cdots & \mathbf{x} \end{pmatrix} \begin{pmatrix} \lambda \\ \dot{\lambda} \\ \vdots \\ \lambda^{(i)} \\ \vdots \\ \lambda^{(m-2)} \\ \lambda^{(m-1)} \end{pmatrix} = \begin{pmatrix} Pg \\ P\dot{g} \\ \vdots \\ Pg^{(i)} \\ \vdots \\ Pg^{(m-2)} \\ Pg^{(m-1)} \end{pmatrix} q.$$

where $c_j^i = \begin{pmatrix} i \\ j \end{pmatrix} \in \mathbb{Z}^+$ for $0 \leq j \leq i \leq (m-1)$. The quantities $\mathbf{x}^{(i)}, 0 \leq i \leq (m-1)$ are the i^{th} order derivatives of the image point. If we define $c_j^i = 0$ for $i < j$, the $(i,j)^{th}$ entry (in fact a tuple) of the first matrix in the above equation has the unified form $c_j^i \mathbf{x}^{(i-j)}$, $0 \leq i, j \leq (m-1)$. We may define matrices $\mathbf{U} \in \mathbb{R}^{3m \times m}, B \in \mathbb{R}^{3m \times 4}$:

$$\mathbf{U} = (c_j^i \mathbf{x}^{(i-j)}), \quad B = (Pg^{(i)}), \quad 0 \leq i, j \leq (m-1). \tag{22}$$

Let $\tilde{\mathbf{u}}_i \in \mathbb{R}^{3m}$ be the i^{th} column of the matrix $\mathbf{U}$ and $b_1, b_2, b_3, b_4 \in \mathbb{R}^{3m}$ be the four columns of the matrix B. We then have the differential version of the Theorem 1.

Theorem 6 *Consider the image $\mathbf{x}(t) \in \mathbb{R}^3$ of a point q under the camera motion $g(t) \in SE(3)$. Then for the matrices $\mathbf{U} \in \mathbb{R}^{3m \times m}$ and $B \in \mathbb{R}^{3m \times 4}$ defined in (22), the column vectors $\{\tilde{\mathbf{u}}_i\}_{i=1}^m \in \mathbb{R}^{3m}$ of the matrix $\mathbf{U}$ and $b_1, b_2, b_3, b_4 \in \mathbb{R}^{3m}$ of the matrix B satisfy the following wedge product equation:*

$$b_1 \wedge b_2 \wedge b_3 \wedge b_4 \wedge \tilde{\mathbf{u}}_1 \wedge \ldots \wedge \tilde{\mathbf{u}}_m = 0. \tag{23}$$

This wedge equation contains all the projective invariants associated with the motion of the image of a single point. The proof is essentially the same as that of the Theorem 1. One would see that most of the constraints given by the wedge product involve high order derivatives of the optical flows or the structural scales. Due to numerical accuracy, they are not very useful for reconstruction purpose. However, constraints involving the first derivative have been widely used. These are simply the bilinear constraints on optical flows, which are a differential version of the bilinear epipolar constraints in the discrete case.

Without loss of generality, we may assume $g(t_0) = I$. Then $\dot{g}$ has the **twist form**:

$$\dot{g} = \begin{pmatrix} \hat{\omega} & v \\ 0 & 0 \end{pmatrix}$$

where $\omega \in \mathbb{R}^3$ is the angular velocity and $v \in \mathbb{R}^3$ the linear velocity. Then, in the case that $m = 2$, the wedge product equation gives:

$$
\begin{aligned}
& b_1 \wedge \ldots \wedge b_4 \wedge \tilde{\mathbf{u}}_1 \wedge \tilde{\mathbf{u}}_2 = \det \begin{pmatrix} Pg & \mathbf{x} & 0 \\ P\dot{g} & \dot{\mathbf{x}} & \mathbf{x} \end{pmatrix} e_1 \wedge \ldots \wedge e_6 = 0 \\
\Leftrightarrow \quad & \det \begin{pmatrix} Pg & \mathbf{x} & 0 \\ P\dot{g} & \dot{\mathbf{x}} & \mathbf{x} \end{pmatrix} = 0 \quad \Leftrightarrow \quad \det \begin{pmatrix} I & 0 & \mathbf{x} & 0 \\ \hat{\omega} & v & \dot{\mathbf{x}} & \mathbf{x} \end{pmatrix} = 0 \\
\Leftrightarrow \quad & \det \begin{pmatrix} I & 0 & \mathbf{x} & 0 \\ 0 & v & \dot{\mathbf{x}} - \hat{\omega}\mathbf{x} & \mathbf{x} \end{pmatrix} = 0 \quad \Leftrightarrow \quad \det(v, \dot{\mathbf{x}} - \hat{\omega}\mathbf{x}, \mathbf{x}) = 0 \\
\Leftrightarrow \quad & \dot{\mathbf{x}}^T \hat{v} \mathbf{x} + \mathbf{x}^T \hat{v} \hat{\omega} \mathbf{x} = 0.
\end{aligned}
$$

This is exactly the differential version of the **epipolar constraint**, or the **bilinear constraint** (see for example Ma and Košecká and Sastry [15]). This equation holds for all the n image points:

$$\dot{\mathbf{x}}^{jT} \hat{v} \mathbf{x}^j + \mathbf{x}^{jT} \hat{v} \hat{\omega} \mathbf{x}^j = 0, \quad 1 \leq j \leq n.$$

If one has $n \geq 8$ points in general position, the relative motion (ω, v) can be determined by solving the **MMSE estimation problem:**

$$\min_{\omega \in \mathbb{R}^3, v \in S^2} V(\omega, v) = \sum_{j=1}^{n} (\dot{\mathbf{x}}^{jT} \hat{v} \mathbf{x}^j + \mathbf{x}^{jT} \hat{v} \hat{\omega} \mathbf{x}^j)^2.$$

Using the linear or nonlinear algorithms on $\mathbb{R}^3 \times S^2$ (Ma, Kosecka and Sastry [15, 16]), the motion (ω, v) can be recovered with v up to a scale.

5.2 Euclidean Constraints and Structure Reconstruction

As we have seen in the discrete case, the purpose of exploiting Euclidean constraints is to reconstruct the scales of the motion and structure. In the differential case, the scale information is encoded in $\lambda^j, \dot{\lambda}^j, 1 \leq j \leq n$ for the structure of the n points, and $\eta \in \mathbb{R}^+$ for the linear velocity v. The differential version of the constraint (16) is just:

$$\dot{\lambda}^j \mathbf{x}^j + \lambda^j \dot{\mathbf{x}}^j = \hat{\omega}(\lambda^j \mathbf{x}^j) + \eta v \quad \Leftrightarrow \quad \dot{\lambda}^j \mathbf{x}^j + \lambda^j (\dot{\mathbf{x}}^j - \hat{\omega}\mathbf{x}^j) - \eta v = 0, \quad 1 \leq j \leq n \tag{24}$$

Known $\mathbf{x}, \dot{\mathbf{x}}, \omega$ and v, these constraints are all linear in $\lambda^j, \dot{\lambda}^j, 1 \leq j \leq n$ and η. Also, if $\mathbf{x}^j, 1 \leq j \leq n$ are linearly independent of v, *i.e.* the feature points do not line up with the direction of translation, these linear constraints are not degenerate hence the unknown scales are determined up to a universal scale. As in the discrete case, we call a configuration **critical** if there is any $\mathbf{x}^j, 1 \leq j \leq n$ which lines up with the translational direction v. In fact, this is the limiting case of the critical configuration defined in the discrete case.

We can arrange all the scale quantities into a single vector $\vec{\lambda}$:

$$\vec{\lambda} = (\lambda^1, \ldots, \lambda^n, \dot{\lambda}^1, \ldots, \dot{\lambda}^n, \eta)^T \in \mathbb{R}^{2n+1}.$$

For n optical flows, $\vec{\lambda}$ is a $2n+1$ dimensional vector. (24) gives $3n$ (scalar) linear equations. The problem of solving $\vec{\lambda}$ from (24) is usually over-determined. As in the discrete case, it is easy to check that in the absence of noise the set of equations given by (24) uniquely determine $\vec{\lambda}$ if the configuration is non-critical. As in the discrete case, we can write all the equations in the matrix form:

$$M\vec{\lambda} = 0$$

with $M \in \mathbb{R}^{3n\times(2n+1)}$ being a matrix depending on ω, v and $\{\mathbf{x}^j, \dot{\mathbf{x}}^j\}_{j=1}^n$. Then in the presence of noise, the LLSE estimate of $\vec{\lambda}$ is just the eigenvector of $M^T M$ corresponding to the smallest eigenvalue.

Notice that the rate of scales $\{\dot{\lambda}^j\}_{j=1}^n$ are also estimated. This piece of information has been ignored in most of previous structure from motion algorithms. However, it turns out to be a very important piece of information. If we do the above estimation for a time interval, say (t_0, t_f), then we obtain the estimation $\vec{\lambda}(t)$ as a function of time t. But the estimation of $\vec{\lambda}(t)$ at each time t is only determined up to an arbitrary scale. Hence $\rho(t)\vec{\lambda}(t)$ is also a valid estimation for any positive function $\rho(t)$. However, since $\rho(t)$ is multiplied to both $\lambda(t)$ and $\dot{\lambda}(t)$. Their ratio:

$$r(t) = \dot{\lambda}(t)/\lambda(t)$$

is independent of the choice of $\rho(t)$ at each time t. Notice $\frac{d}{dt}(\ln\lambda) = \dot{\lambda}/\lambda$. Let the logarithm of the structural scale λ to be $y = \ln\lambda$. Then a time-consistent estimation $\lambda(t)$ needs to satisfy the following ordinary differential equation, we call it the **dynamic scale ODE**:

$$\dot{y}(t) = r(t).$$

Given $y(t_0) = y_0 = \dot{\lambda}(t_0)/\lambda(t_0)$, solve this ODE and and obtain $y(t)$ for $t \in [t_0, t_f]$. Then the time-consistent scale $\lambda(t)$ is simply given by:

$$\lambda(t) = \exp(y(t)).$$

Thus, all the scales estimated at different times are with respect to the scales at time t_0. One can also normalize all the scales with respect to those at time t_f by setting the final value $y(t_f)$ and then integrating the ODE backwards. Therefore, in the differential case, we are able to recover all the scales as a function of time up to a universal scale. Notice that in particular the (relative) scales of the translational motion v are fully recovered, which is very important to many applications in mobile robot navigation. From the above discussion, we have a differential version of Theorem 4:

Theorem 7 (Uniqueness of Reconstruction: Differential Case) *Consider a moving camera and $n \geq 8$ points in general positions. If the configuration is non-critical for the time interval (t_0, t_f), then the Euclidean structure of the n points and motion of the camera as a function of time can be reconstructed up to a universal scale.*

In the differential case, the idea of **triangulation** is essentially the same: try to find a consistent reconstruction of the Euclidean structure from all the structure estimated over time. However, it is much harder to implement in a practical algorithm since it involves integration of the motion $(\omega(t), v(t))$ unless we have an estimation of the transformation $g(t) \sim (R(t), p(t))$ from other sources. The issue of estimating the velocity and the transformation together will be addressed in section 6 which deals with hybrid settings.

Main Algorithm II: Differential Case

- At each time $t \in (t_0, t_f)$, compute the motion (ω, v) by solving the MMSE estimation problem:

$$\min_{\omega \in \mathbb{R}^3, v \in S^2} V(\omega, v) = \sum_{j=1}^{n} (\dot{\mathbf{x}}^{jT} \hat{v} \mathbf{x}^j + \mathbf{x}^{jT} \hat{v} \hat{\omega} \mathbf{x}^j)^2.$$

  This can be solved by the linear or nonlinear algorithms (Ma, Kosecka and Sastry [15, 16]).

- At each time t, solve the LLSE estimation problem

$$\min_{\|\vec{\lambda}\|=1} V(\vec{\lambda}) = \sum_{j=1}^{n} [\dot{\lambda}^j \mathbf{x}^j + \lambda^j (\dot{\mathbf{x}}^j - \hat{\omega} \mathbf{x}^j) - \eta v]^2 = \|M\vec{\lambda}\|^2$$

  for all available constraints from (24).

- Obtain $\vec{\lambda}(t)$ as a function of t for $t \in [t_0, t_f]$. Calculate the ratio function $r(t) = \dot{\lambda}/\lambda$ and solve the dynamic scale ODE to get $y(t)$. Normalize all the scales in $\vec{\lambda}(t)$ with respect to the time-consistent scale $\lambda(t) = \exp(y(t))$. Notice one only needs to do this for the scale of one point, say $\lambda^j(t)$. The structural scales of the n points and scales of the translational motion $v(t)$ are then reconstructed from the vector $\vec{\lambda}$ up to a universal scale.

In practice, the ratio function $r(t)$ may not be available for all the times $t \in [t_0, t_f]$. One can use some simple interpolation schemes to recover $r(t)$, hence the time-consistent scale $\lambda(t)$. It is up to the user to adjust the algorithm appropriately for the specific applications.

Comments 1 *In both the discrete and differential cases, the proposed algorithms reconstruct both the Euclidean structure and motion up to a single universal scale. These algorithms provide any vision-based autonomous agent, for example an autonomous mobile robot, with* **complete information** *about its surrounding environment and its ego-motion relative to the environment. The universal scale is not important since it only scales up or down the overall configuration space (as a Riemannian manifold). All the intrinsic geometric (including metric) properties of the space are preserved. In this sense, no information is really lost through a vision system.*

6 Hybrid Reconstruction of Structure and Motion

We now study the cases where both point correspondences and optical flow measurements are available. Such cases are referred to as **hybrid**. In practical systems the quality of the motion/structure estimates naturally depend on the quality of the measurements. Large motions, occlusions, reflectance variations, aliasing etc. affect negatively the quality of the flow estimates as well as the point correspondences. Therefore it is of interest to study the case when both types of measurements are used for motion and structure estimation. In the following section we present theoretical analysis of such scenario and propose corresponding algorithms.

6.1 Hybrid Case I

Suppose one point q is projected on all m image frames (in discrete positions) and its optical flows on these frames are also measured. We refer to this case as **hybrid case I.** It is a natural combination (a "direct sum") of the discrete case and the differential case we studied in the preceding sections. For this case, we have:

$$\begin{pmatrix} \mathbf{x}_1 & 0 & 0 & \cdots & 0 \\ \dot{\mathbf{x}}_1 & \mathbf{x}_1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \mathbf{x}_m & 0 \\ 0 & \cdots & 0 & \dot{\mathbf{x}}_m & \mathbf{x}_m \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \dot{\lambda}_1 \\ \vdots \\ \lambda_m \\ \dot{\lambda}_m \end{pmatrix} = \begin{pmatrix} Pg_1 \\ P\dot{g}_1 \\ \vdots \\ Pg_m \\ P\dot{g}_m \end{pmatrix} q.$$

In general, $\dot{g}_i, 1 \le i \le m$ have the form:

$$\dot{g}_i = g_i\hat{\xi}_i = g\begin{pmatrix} \hat{\omega}_i & v_i \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} R_i\hat{\omega}_i & R_i v_i \\ 0 & 0 \end{pmatrix}$$

Similarly, we define the matrices $\mathbf{H} \in \mathbb{R}^{6m\times 2m}$ and $C \in \mathbb{R}^{6m\times 4}$ to be:

$$\mathbf{H} = \begin{pmatrix} \mathbf{x}_1 & 0 & 0 & \cdots & 0 \\ \dot{\mathbf{x}}_1 & \mathbf{x}_1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \mathbf{x}_m & 0 \\ 0 & \cdots & 0 & \dot{\mathbf{x}}_m & \mathbf{x}_m \end{pmatrix}, \quad C = \begin{pmatrix} Pg_1 \\ Pg_1\hat{\xi}_1 \\ \vdots \\ Pg_m \\ Pg_m\hat{\xi}_m \end{pmatrix}.$$

Let $\tilde{\mathbf{h}}_i \in \mathbb{R}^{6m}$ to be the i^{th} column of the matrix $\mathbf{H}$ and c_1, c_2, c_3, c_4 be the four columns of the matrix C. We then have the following result:

Theorem 8 *For the hybrid case I, the vectors $\{\tilde{\mathbf{h}}_i\}_{i=1}^{2m}$ and $\{c_i\}_{i=1}^{4}$ defined as above satisfy the following wedge product equation:*

$$c_1 \wedge c_2 \wedge c_3 \wedge c_4 \wedge \tilde{\mathbf{h}}_1 \wedge \ldots \wedge \tilde{\mathbf{h}}_{2m} = 0.$$

Obviously, this wedge product equation gives all the discrete projective constraints (bilinear, trilinear and quadrilinear ones) given by the wedge product in Theorem 1; it also gives all the differential (bilinear) epipolar constraints we used in the differential case. Further, some new constraints are given by this wedge product. These constraints involving both velocity $\{(\omega_i, v_i)\}_{i=1}^{m}$ and transformation $\{(R_i, p_i)\}_{i=1}^{m}$ are called **hybrid constraints**. In fact all the constraints given by the wedge product equation are the same as that all the $(2m+4)\times(2m+4)$ minors of the $6m \times (2m+4)$ matrix $(C, \mathbf{H})$ are degenerate (*i.e.* the determinant is zero). All the non-trivial constraints given by these minors will be homogeneous equations in terms of the entries of $\{(\mathbf{x}_i, \dot{\mathbf{x}}_i)\}_{i=1}^{m}$. According to the structure of the matrix $\mathbf{H}$, **the degree of these homogeneous (hybrid) constraints is from degree 2 to degree 8.**

Without loss of generality, we will assume that consecutive frames are non-critical. Then the homogeneous constraints above determine the motion $\{(\omega_i, v_i)\}_{i=1}^{m}$ and $\{(R_i, p_i)\}_{i=1}^{n}$ with translational motion $\{v_i\}_{i=1}^{m}$ and $\{p_i\}_{i=1}^{m}$ up to unknown scales. In order to reconstruct the structural scales and the scales of motions, one needs to use the following set of Euclidean constraints from both the discrete case and the differential case:

$$\lambda_i^j \mathbf{x}_i^j - \lambda_{i-1}^j R_i \mathbf{x}_{i-1}^j - \gamma_i p_i = 0, \quad 2 \le i \le m, 1 \le j \le n$$
$$\dot{\lambda}_i^j \mathbf{x}_i^j + \lambda_i^j(\dot{\mathbf{x}}_i^j - \hat{\omega}_i \mathbf{x}_i^j) - \eta_i v_i = 0, \quad 1 \le i \le m, 1 \le j \le n.$$

As long as the discrete case and differential case respectively have unique solutions, the overall hybrid case has a unique solution (up to a universal scale). The estimation is simply an LLSE problem.

In particular, the scales of velocities at a particular time can be uniquely recovered with respect to the transformation between the current image frame and a reference image frame. This is very important for applications such as mobile robot navigation since a consistent estimation of the displacements and velocities can be obtained. Notice that, in the i^{th} image frame, we certainly can measure optical flows for points which do not have projections in the other image frames at all. Their structural scales can also be determined with respect to the same universal scale. Then **the occlusion is certainly not a problem at all in the hybrid case I.** In fact, if regarding the optical flows as the limit of image correspondences between two image frames very close in time, this is essentially another way of stating Theorem 5.

Notice that in the hybrid case I, the quantities $\{\dot{\lambda}_i^j\}_{i=1,j=1}^{m,n}$ are not quite useful since we are not measuring the optical flows in a continuous fashion. So one can get rid of them by applying cross product with $\{\mathbf{x}_i^j\}_{i=1,j=1}^{m,n}$ to the differential Euclidean constraints:

$$\dot{\lambda}_i^j \mathbf{x}_i^j + \lambda_i^j(\dot{\mathbf{x}}_i^j - \hat{\omega}_i \mathbf{x}_i^j) - \eta_i v_i = 0 \quad \Leftrightarrow \quad \lambda_i^j(\dot{\mathbf{x}}_i^j - \hat{\omega}_i \mathbf{x}_i^j) \times \mathbf{x}_i^j - \eta_i v_i \times \mathbf{x}_i^j = 0.$$

Then the number of states in the associated LLSE estimation problem can be reduced. This is essentially the bilinear constraint used by some researchers in the structure from motion algorithms using optical flow, see for example [23].

6.2 Hybrid Case II

For the **hybrid case II**, we consider the situation that optical flows of n points are measured for the whole time interval $[t_0, t_f]$, the correspondences of the n points between the initial (1^{th}) and final (2^{nd}) image frames are also given. Essentially, for reconstructing the Euclidean scales, the hybrid case II can use the same set of Euclidean constraints as hybrid case I (with the differential ones over the whole time interval). The dynamic scale ODE is needed to recover the time consistency of scales as in the pure differential case.

However, for estimating the motion $(\omega(t), v(t))$ and the displacement $(R(t_f), p(t_f))$ between the two frames, instead of using the eight point motion estimation algorithms for each time t separately, we can formulate it as a standard optimal control problem as follows.

Regard the velocities $u = (\omega, v) \in \mathbb{R}^6$ as the control inputs to the following dynamic system on the Lie group $SE(3)$:

$$\dot{g} = g\hat{u}, \quad \text{where} \quad \hat{u} = \begin{pmatrix} \hat{\omega} & v \\ 0 & 0 \end{pmatrix}.$$

Now define the **final-state cost function:**

$$\phi(g(t_f)) = \sum_{j=1}^{n} w^j (\mathbf{x}^j(t_0)^T R^T \hat{p} \mathbf{x}^j(t_f))^2,$$

and the **Lagrangian:**

$$L(u, t) = \sum_{j=1}^{n} w^j (\dot{\mathbf{x}}^{jT} \hat{v} \mathbf{x}^j + \mathbf{x}^{jT} \hat{v} \hat{\omega} \mathbf{x}^j)^2 + w_\omega \omega^T \omega + w_v v^T v, \tag{25}$$

where $w^j, w_\omega, w_v > 0$ are weights for different measurements. Then **the motion recovery problem is naturally equivalent to an optimal control problem on the Lie group $SE(3)$** (see [26, 21] for examples of optimal control on Lie groups): the optimal motion $u^* = (\omega^*, v^*)$ is the optimal control law for the dynamic system on $SE(3)$:

$$\dot{g} = g\hat{u},$$

subject to the constraint on the final state $g(t_f)$:

$$\psi(g(t_f)) = p^T p = 1,$$

which minimizes the objective:

$$J(u(\cdot)) = \phi(g(t_f)) + \int_{t_0}^{t_f} L(u,t)\, dt.$$

The solution for this optimal control problem gives a more time-consistent estimation of the motion. Many numerical algorithms exist for solving this type of optimal control problems (one can refer to Bryson and Ho's [1] for a detailed discussion on these algorithms). Motion estimation schemes which consider dynamic consistency are usually referred to as **dynamic motion estimation schemes.** A more detailed study of this approach will be presented in another paper.

In the case that the image correspondences between the initial and final image frames are not available, the above optimal control problem simply becomes one without a penalty on the final state. The solution is still well determined. Notice that, by solving these optimal control problems, one automatically obtains both the optimal velocity $(\omega(t), v(t))$ and the transformation $g(t) \sim (R(t), p(t))$ as a function of time t. As we mentioned in the differential case section, such a $g(t)$ can be used for triangulation purposes.

7 Conclusions and Discussions

In this paper, the problem of Euclidean structure and motion recovery from multiple frames is thoroughly studied for the calibrated camera case. Two types of constraints are presented in this paper: constraints on motion parameters and constraints on (Euclidean) structural parameters. The algorithms for motion and structure recovery are naturally derived from these constraints. The geometric intuitions associated with the problem are nicely revealed through the clean roles that these two types of constraints have played in the proposed algorithms. Such geometric intuitions usually are very elusive in the uncalibrated camera case when the projective geometry is applied. On the other hand, a better understanding of the uncalibrated case should be based on a better understanding of the calibrated case, which is simpler but crucial.

From our experience, we notice that most vision problems, although difficult, are over determined problems. That is, the existence of (unique) solutions is usually guaranteed, but there are always much more equations than needed to solve the parameters. This is probably also the reason why there are always multiple algorithms or approaches to solve the same vision problem. However, it is always hard to argue whether an approach is better or not. We believe a good approach not only is based on a solid mathematical understanding but also is an approach which can be easily applied to general cases. This is the reason we always study the discrete and differential (and even hybrid) cases together.

Notice that at least in the calibrated camera case, one does not need the projective geometry framework at all. In fact, the Euclidean setup seems more suitable for the problem. Differential geometric properties associated with the structure and motion recovery problem can be more easily and clearly revealed. The study of the uncalibrated camera case within this framework is currently under way. We are also in the process of carrying out the experimental evaluation of the proposed algorithms.

References

[1] Arthur E. Bryson and Yu-Chi Ho. *Applied Optimal Control.* Blaisdell Publishing Company, 1969.

[2] S. Christy and R. Horaud. Euclidean shape and motion from multiple perspective views by affine iterations. *IEEE Transactions on PAMI*, 18(11):1098–1104, 1996.

[3] Oliver Faugeras. What can be seen in 3d with and uncalibrated stereo rig. In *Proceedings of 2nd European Conference on Computer Vision*, pages 563–578. Springer-Verlag, 1992.

[4] Oliver Faugeras. Stratification of three-dimensional vision: projective, affine, and metric representations. *Journal of the Optical Society of America*, 12(3):465–84, 1995.

[5] Olivier Faugeras, Francis Lustman, and Giorgio Toscani. Motion and structure from motion from point and line matches. In *Proceeding of IEEE First International Conference on Computer Vision*, pages 25–34, London, England, 1987. IEEE Comput. Soc. Press.

[6] Olivier Faugeras and Bernard Mourrain. On the geometry and algebra of the point and line correspondences between N images. In *Proceeding of Fifth International Conference on Computer Vision*, pages 951–6, Cambridge, MA, USA, 1995. IEEE Comput. Soc. Press.

[7] 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.

[8] R. Hartley, R. Gupta, and T. Chang. Stereo from uncalibrated cameras. In *Proceeding of Conference on Computer Vision and Pattern Recognition*, pages 761–4, Urbana-Champaign, IL, USA, 1992. IEEE Comput. Soc. Press.

[9] R. Hartley and Peter Sturm. Triangulation. In *Proceeding of 1994 Image Understanding Workshop*, pages 957–966, Monterey, CA USA, 1994. OMNIPRESS.

[10] R. I. Hartley. Euclidean reconstruction from uncalibrated views. In J. L. Mundy, A. Zisserman, D. Forsyth, and J.-O. Berlin, editors, *Applications of Invariance in Computer Vision.*, pages 237–56, 1994.

[11] B. P. Horn. Relative orientation. *International journal of Computer Vision*, 4(59), 1990.

[12] N. Ahuja J. Weng and T.S. Huang. Optimal motion and structure estimation. *IEEE Transactions on PAMI*, 15(9):843–884, 1993.

[13] J. J. Konderink and A. J. van Dorn. Affine structure from motion. *Journal of Optical Society of America*, 8(2):337–385, 1991.

[14] Q.-T. Luong and O. Faugeras. Self-calibration of a moving camera from point correspondences and fundamental matrices. *International Journal of Computer Vision*, vol.22(3):261–89, 1997.

[15] Yi Ma, Jana Košecká, and Shankar Sastry. Motion recovery from image sequences: Discrete viewpoint vs. differential viewpoint. In *Proceeding of European Conference on Computer Vision, or UC Berkeley Memorandum No. UCB/ERL M98/11*, 1998.

[16] Yi Ma, Jana Košecká, and Shankar Sastry. Optimal motion from image sequences: A Riemannian viewpoint. In *Proceeding of the Conference on Mathematical Theory of Networks and Systems*, 1998.

[17] Philip F. McLauchlan and David W. Murray. A unifying framework for structure and motion recovery from image sequences. In *Proceedings of IEEE fifth Internation Conference on Computer Vision*, pages 314–20, Cambridge, MA USA, 1995. IEEE Com. Soc. Press.

[18] Richard M. Murray, Zexiang Li, and Shankar Sastry. *A Mathematical Introduction to Robotic Manipulation.* CRC Press, 1993.

[19] C.J. Poelman and T. Kanade. A paraperspective factorization method for shape and motion recovery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3):206–18, 1997.

[20] L. Robert, C. Zeller, O. Faugeras, and M. Hebert. Applications of nonmetric vision to some visually guided tasks. In I. Aloimonos, editor, *Visual Navigation*, pages 89–135, 1996.

[21] Shankar Sastry and Richard Montgomery. The structure of optimal control for a steering problem. In *Proceeding of the IFAC Workshop on Nonlinear Control*, pages 135–40, 1992.

[22] Amnon Shashua. Multiple-view geometry and photometry. In *Recent Developments in Computer Vision, ACCV95*, pages 395–404, Singapore, 1995. Springer.

[23] Stefano Soatto and Roger Brockett. Optimal and suboptimal structure from motion. In *Proceedings of IEEE International Conference on Computer Vision*, 1997.

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

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

[26] Claire Tomlin, Yi Ma, and Shankar Sastry. Free flight in 2000: Games on Lie groups. In *Submitted to 1998 IEEE Conference on Decision & Control (CDC)*, 1998.

[27] G. Toscani and O. D. Faugeras. Structure and motion from two noisy perspective images. *Proceedings of IEEE Conference on Robotics and Automation*, pages 221–227, 1986.

[28] Bill 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.

[29] Bill 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.

[30] Bill Triggs. The geometry of projective reconstruction I: Matching constraints and the joint image. *International Journal of Computer Vision*, to appear.

[31] Hermann Weyl. *The Classical Groups: Their Invariants and Representations.* Princeton University Press, 1946.

[32] Z. Zhang, Q.-T. Luong, and O. Faugeras. Motion of an uncalibrated stereo rig: self-calibration and metric reconstruction. *IEEE Transactions on Robotics and Automation*, 12(1):103–13, 1996.