June 2001

UILU-ENG-01-2215
DC-220

# RANK CONDITIONS OF THE MULTIPLE VIEW MATRIX IN MULTIPLE VIEW GEOMETRY

**Yi Ma, Kun Huang, Rene Vidal, J. Košecká and Shankar Sastry**

*Coordinated Science Laboratory*
University of Illinois at Urbana-Champaign
1308 West Main Street, Urbana, IL 61801

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB NO. 0704-0188*



| 1. AGENCY USE ONLY *(Leave blank)* | 2. REPORT DATE | 3. REPORT TYPE AND DATES COVERED |
|---|---|---|
| | June 2001 | |

4. TITLE AND SUBTITLE
Rank Conditions of the Multiple View Matrix in Multiple View Geometry

5. FUNDING NUMBERS

6. AUTHOR(S)
Yi Ma, Kun Huang, Rene Vidal, Jana Košecká, and Shankar Sastry

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
**Coordinated Science Laboratory**
**University of Illinois at Urbana-Champaign**
**1308 West Main Street**
**Urbana, Illinois 61801-2307**

8. PERFORMING RGANIZATION REPORT NUMBER
**UILU-ENG-2215**
**(DC-220)**

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
UIUC ECE startup fund,
ARO DAAD19-00-1-0466,
NSF SBC-MIT-5710000330,
ONR N00014-00-1-0621

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
**The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official position, policy or decision, unless so designated by other documentation**

12a. DISTRIBUTION/AVAILABILITY STATEMENT
**Approved for public release; distribution unlimited.**

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This technical report is a comprehensive collection of four self-contained technical papers which I have jointly written with my PhD student Kun Huang (UIUC ECE), PhD student Rene Vidal (UCB EECS), professor Jana Košecká (GMU CS) and professor Shankar Sastry (UCB EECS). It consists of a coherent treatment of multiple view geometry from a linear algebraic viewpoint. In particular, a newly introduced concept of multiple view matrix and its associated rank deficiency condition have been extensively studied for the purpose of 2-D to 3-D reconstruction. The proposed framework provides a brand new approach to multiple view geometry, which is independent of previous approaches based on projective geometry, tensor analysis or algebraic geometry, and which has, nonetheless, demonstrated significant theoretical and algorithmic advantages.

The technical report adopts a homogeneous terminology - which makes it slightly different from each original paper. Each chapter corresponds to an original paper and it is still kept rather self-contained in this report. Although we are still in the process of grasping the full implication of the developed theory and algorithms and carrying out more experiments on real images, the report is contrived for the purpose of communicating among researchers who share the same interest in multiple view geometry and would like to try to extend the theory and apply to other applications.

14. SUBJECT TERMS
Multiple view matrix, multilinear constraints, rank deficiency condition, feature matching, motion and structure recovery

15. NUMBER OF PAGES
78

16. PRICE CODE

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| UNCLASSIFIED | UNCLASSIFIED | UNCLASSIFIED | UL |

# Rank Conditions of the Multiple View Matrix

in Multiple View Geometry

Yi Ma
Kun Huang
René Vidal
Jana Košecká
Shankar Sastry

June 18, 2001

# Contents

# Preface

*"It is good that I did not let myself be influenced."*
— Ludwig Wittgenstein

This technical report is a comprehensive collection of four self-contained technical papers which I have jointly written with my PhD student Kun Huang (UIUC ECE), PhD student René Vidal (UCB EECS), professor Jana Košecká (GMU CS) and professor Shankar Sastry (UCB EECS). It consists of a coherent treatment of multiple view geometry from a linear algebraic viewpoint. In particular, a newly introduced concept of *multiple view matrix* and its associated *rank deficiency condition* have been extensively studied for the purpose of 2-D to 3-D reconstruction. The proposed framework provides a brand new approach to multiple view geometry, which is independent of previous approaches based on projective geometry, tensor analysis or algebraic geometry, and which has, nonetheless, demonstrated significant theoretical and algorithmic advantages.

The technical report adopts a homogeneous terminology - which makes it slightly different from each original paper. Each chapter corresponds to an original paper and it is still kept rather self-contained in this report. Although we are still in the process of grasping the full implication of the theory and algorithms developed so far and carrying out more experiments on real images, the report is contrived for the purpose of communicating among researchers who share the same interest in multiple view geometry and would like to try to extend the theory and apply to other applications.

I would like to thank professor Robert Fossum at UIUC Mathematics department for his kind help and useful suggestions during the preparation of some of these chapters - including proofreading some of them. I thank professor P. R. Kumar at UIUC ECE department for his support for Kun Huang and his encouragement on carrying out research along this line. I also thank professor Stefano Soatto at UCLA CS department who has given me valuable advice on the presentation of the papers.

This research is primarily supported by the UIUC ECE department startup fund; Kun Huang is supported by U.S. Army Research Office under Contract DAAD19-00-1-0466, and the National Science Foundation KDI initiative under subcontract SBC-MIT-5710000330; René Vidal is supported by ONR grant N00014-00-1-0621; and Jana Košecká is supported by the GMU CS department startup fund.

*Yi Ma*
Dept. of Electrical & Computer Eng.
Univ. of Illinois at Urbana-Champaign
yima@uiuc.edu

# Chapter 1

# The Multiple View Matrix for Point Features

*Yi Ma, René Vidal, Kun Huang, Shankar Sastry*
*Submitted to CVPR01, May 18th.*

## Abstract

*A new rank deficiency condition is presented for multiple images of a point. We show that a set of m images corresponds to a unique 3-D pre-image if and only if the so-called multiple view matrix $M_p \in \mathbb{R}^{3(m-1)\times 2}$ is of rank 1. If the rank is always 0, then the pre-image is only determined up to a line on which all the camera centers must lie. This condition is shown to be equivalent to all the multilinear constraints, but it tremendously simplifies the derivation and proof of all the algebraic relationships among bilinear, trilinear and quadrilinear constraints. The rank deficiency condition gives rise to a set of natural linear algorithms for purposes such as matching feature points and motion estimation from images of multiple points. These linear algorithms use all available data simultaneously without specifying a particular choice of a set of pairwise, triple-wise or quadruple-wise images. We present simulation and experimental results that validate the proposed algorithms.*
**Key words:** Multiple view matrix, multilinear constraints, rank deficiency condition, feature matching, motion recovery.

## 1.1 Introduction

Multiple view geometry has been extensively studied for the past decade and is now considered as a well-established area in computer vision (see [8]). Although algebraic and geometric relationships among constraints governing multiple images have been explored for numerous applications, theoretical and algorithmic aspects of multiple view geometry seem to grow in different directions. In theory, it is well-known that the relationship among multiple images can be reduced to that among two, three, or four images (or views) at a time. Hence the study of the associated bifocal, trifocal and quadrifocal tensors has been of primary interest in the past few years. However, there is yet no clear consensus on how to systematically and simultaneously exploit those constraints among pairwise, triple-wise or quadruple-wise images for a consistent 3-D reconstruction from multiple (typically more than four) images. Many existing algorithms depend on a particular choice of a (sufficient but minimal) set of cascaded pairwise, triple-wise or even quadruple-wise constraints. Given that such a choice is by no means unique – in fact, the number of choices grows exponentially along with the number of images – performance of such algorithms is very hard to evaluate or justify. Hence in many practical algorithms, "global" and "direct" methods, such as *factorization method*, are sought instead in order to use all data simultaneously. Such global algorithms, however, rely very little on the difficult theory of multilinear constraints. So why is there such a separation of theory and algorithm? Is there any

way of restating the relationship among multiple images that is equivalent to the multilinear constraints but much easier to use for global reconstruction? This chapter will show how the theory of multilinear constraints and the algorithms for global reconstruction can be closely related to the *rank deficiency condition* of the so-called *multiple view matrix* $M_p$.

There is probably another practical reason why we have not seen the theory of multilinear constraints being applied to a broader range of applications. There is not yet any characterization of these constraints which is simple enough for people who are not necessarily experts in tensorial algebra or algebraic geometry to fully understand it and use it. We will show that the rank deficiency condition is a simpler and more unifying characterization of the relationship among multiple images of a 3-D point and their dependency. This condition does not discriminate Euclidean, affine or projective camera models, is mathematically equivalent to all the multilinear constraints, and gives rise to algorithms that use all the data simultaneously without specifying a particular set of pairwise, triple-wise or quadruple-wise images. These algorithms include: matching feature points, mapping images to a new view, camera motion estimation and so on. Since the rank deficiency of the multiple view matrix is a purely linear algebraic condition, the so-developed algorithms are mostly *linear*. In particular, we will show that, regardless of the type of camera model (Euclidean, affine or projective), the motion and structure reconstruction problem from multiple images is "almost globally factorizable", which means that, except for the initialization step, the algorithm is based upon a singular value decomposition (SVD) of all the data simultaneously. This extends previous factorization algorithms known for orthographic, affine and projective camera models [18, 26, 28] to any perspective camera model.

Significance of this rank deficiency condition is far-reaching. It not only serves as a bridge between theory and algorithm for multiple view geometry, but also changes how constraints among multiple images should be described. Instead of using algebraic equations as we were used to do, a more efficient and concise way to capture those constraints is to impose a rank deficiency condition on the multiple view matrix $M_p$. In the next chapter "The Multiple View Matrix for Line Features", we show that the same idea generalizes to line (curve and surface) features as well. An exactly parallel set of theorems and algorithms are developed for line features. Combining the point and line cases, our results in fact can be directly extended to the study of multiple images of any 3-D objects. Hence the rank deficiency condition indeed provides a unified framework for the study of multiple view geometry.

**Relation to previous work:** This work provides a new perspective to the multiple view geometry which used to be based on the study of multilinear constraints for point or line and their dependency [2, 6, 7, 10, 9, 27]. It is almost impossible to give a complete list of references that are related to this subject. However, a comprehensive account can be found in [3, 33, 8]. Previous work mainly focuses on characterizing the geometry among pairwise [16], triple-wise [1, 20] or quadruple-wise [27, 22] views as well as numerous algorithms associated to it. Our work shows that a *global* characterization of multiple images together is indeed possible. Furthermore, a unified theoretical treatment for point and line (and even curve and surface) is also possible, in addition to previous effort in algorithm development [7]. The reader may find that some of the (theoretical) results presented in this chapter may be equivalent to some previously known facts. But we believe it is the first time that all the existing and new results can be easily obtained in such a concise and unified framework. As a result, new linear algorithms can be obtained to replace old ones. All the proofs and algorithms are done using linear algebra only, and in particular we *do not* use any tensorial notation or algebraic geometry.

**How this chapter is organized:** Section 1.2 formulates and briefly reviews the origin of multilinear constraints among multiple images of a 3-D point $p$. Section 1.3 introduces the multiple view matrix $M_p$ associated with multiple images of the point $p$ and demonstrates that the rank deficiency condition of matrix $M_p$ implies all multilinear constraints between arbitrary views of the point. We also state the conditions on the uniqueness and degeneracy of the pre-image of multiple views in terms of the $M_p$ matrix and provide a clear geometric interpretation for the $M_p$ matrix itself in Section 1.4. As natural applications of the rank deficiency condition, Section 1.5 outlines two linear algorithms: one for point feature matching and another for motion and structure estimation. In Sections 1.6 and 1.7 the proposed algorithms are evaluated by both simulation and experiments. Section 1.8 concludes the chapter.

## 1.2 Multiple views of a point

An image $\mathbf{x}(t) = [x(t), y(t), z(t)]^T \in \mathbb{R}^3$ (in homogeneous coordinates) of a point $p \in \mathbb{E}^3$, with homogeneous coordinates $\mathbf{X} = [X, Y, Z, W]^T \in \mathbb{R}^4$ relative to a fixed world coordinate frame, taken by a moving camera satisfies the following relationship

$$\lambda(t)\mathbf{x}(t) = A(t)Pg(t)\mathbf{X}, \tag{1.1}$$

where $\lambda(t) \in \mathbb{R}_+$ is the (unknown) depth of the point $p$ relative to the camera frame, $A(t) \in SL(3)$ is the camera calibration matrix (at time $t$), $P = [I, 0] \in \mathbb{R}^{3\times 4}$ is the constant projection matrix and $g(t) \in SE(3)$ is the coordinate transformation from the world frame to the camera frame at time $t$. In the above equation, all $\mathbf{x}, \mathbf{X}$ and $g$ are in homogeneous representation. The reader may have noticed that here we allow the calibration matrix $A$ to change in time. This is partly because such a generalization does not make more difficult the development of the results in this chapter. In addition, it encompasses a richer class of practical situations when some camera intrinsic parameters such as the focal length may indeed change from image to image.

In a realistic situation, we usually only obtain "sampled" images of $\mathbf{x}(t)$ at some time instances, say $t_1, t_2, \ldots, t_m \in \mathbb{R}$. For simplicity we denote

$$\lambda_i = \lambda(t_i), \quad \mathbf{x}_i = \mathbf{x}(t_i), \quad \Pi_i = A(t_i)Pg(t_i).$$

The matrix $\Pi_i$ is then a $3 \times 4$ matrix which relates the $i^{th}$ image $\mathbf{x}_i$ of the point $p$ to its world coordinates $\mathbf{X}$ by

$$\boxed{\mathbf{x}_i\lambda_i = \Pi_i\mathbf{X}} \tag{1.2}$$

for $i = 1, \ldots, m$. In the above equations, except for the $\mathbf{x}_i$'s, everything else is unknown and subject to recovery. However, solving for the $\lambda_i$'s, $\Pi_i$'s and $\mathbf{X}$ simultaneously from such equations is extremely difficult. A traditional way to simplify the task is to decouple the recovery of the matrices $\Pi_i$'s from that of $\lambda_i$'s and $\mathbf{X}$. For that purpose, let us rewrite the system of equations (1.2) in a single matrix form

$$\begin{array}{ccc} & \mathcal{I}\vec{\lambda} & = \Pi\mathbf{X} \\ \begin{bmatrix} \mathbf{x}_1 & 0 & \cdots & 0 \\ 0 & \mathbf{x}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{x}_m \end{bmatrix} & \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_m \end{bmatrix} = \begin{bmatrix} \Pi_1 \\ \Pi_2 \\ \vdots \\ \Pi_m \end{bmatrix} \mathbf{X}. \end{array} \tag{1.3}$$

For obvious reasons, we will call $\vec{\lambda} \in \mathbb{R}^m$ the *depth scale vector*, and $\Pi \in \mathbb{R}^{3m\times 4}$ the *projection matrix* associated to the *image matrix* $\mathcal{I} \in \mathbb{R}^{3m\times m}$.

In order to eliminate the unknowns $\vec{\lambda}$ and $\mathbf{X}$, we consider the following matrix in $\mathbb{R}^{3m\times(m+4)}$

$$N_p := [\Pi,\ \mathcal{I}] = \begin{bmatrix} \Pi_1 & \mathbf{x}_1 & 0 & \cdots & 0 \\ \Pi_2 & 0 & \mathbf{x}_2 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ \Pi_m & 0 & \cdots & 0 & \mathbf{x}_m \end{bmatrix}. \tag{1.4}$$

Since it is clear that there exists $v := [\mathbf{X}^T, -\vec{\lambda}^T]^T \in \mathbb{R}^{m+4}$ in the null space of $N_p$, equation (1.3) is equivalent to:

$$\boxed{\text{rank}(N_p) \leq m + 3} \tag{1.5}$$

**Remark 1 (Null space of $N_p$).** *Even though equations (1.3) and (1.5) are equivalent, if rank$(N_p) \leq m+2$ then equation $N_p v = 0$ has more than one solution. In the next section, we will show that rank$(N_p)$ is either $m + 2$ or $m + 3$ and that the first case happens if and only if the point being observed and all the camera centers lie on the same line.*

**Remark 2 (Positive depth).** *Even if rank$(N_p) = m + 3$, there is no guarantee that $\vec{\lambda}$ in (1.3) will have positive entries.*[1] *In practice, if the point being observed is always in front of the camera, then $\mathcal{I}$, $\Pi$ and $\mathbf{X}$ in (1.3) will be such that the entries of $\vec{\lambda}$ are positive. Since the solution to $N_p v = 0$ is unique and $v := [\mathbf{X}^T, -\vec{\lambda}^T]^T$ is a solution, then the last $m$ entries of $v$ have to be of the same sign.*

It is known from linear algebra that the matrix $N_p$ is rank deficient if and only if all its minors (of $(m+4) \times (m+4)$ submatrices) are zero. In this way, we may obtain numerous polynomial equations (in the entries of the projection matrix $\Pi$ and images $\mathbf{x}_1, \ldots, \mathbf{x}_m$) in the general form:

$$f(\Pi, \mathbf{x}_1, \ldots, \mathbf{x}_m) = 0. \tag{1.6}$$

The polynomial $f(\cdot)$ is always linear in each $\mathbf{x}_i$, $i = 1, \ldots, m$. It is known in the computer vision literature that any such a polynomial can be reduced to those involving $2, 3$ or $4$ images at at time from $\mathbf{x}_1, \ldots, \mathbf{x}_m$. These are the well-known bilinear, trilinear or quadrilinear constraints among multiple images. Correspondingly the coefficients of these constraints give rise to the so-called bifocal, trifocal and quadrifocal tensors. However, the existing derivation and formulation of the multilinear constraints and multifocal tensors are quite involved and usually make it hard to utilize these constraints in a uniform fashion for practical purposes. We now propose an alternative approach to study the relationship among multiple images whose connection with the multilinear constraint approach will soon becomes clear.

## 1.3 The multiple view matrix and its rank

Without loss of generality, we may assume that the first camera frame is chosen to be the reference frame.[2] That gives the projection matrices $\Pi_i, i = 1, \ldots, m$

$$\Pi_1 = [I, 0], \ \ldots, \ \Pi_m = [R_m, T_m] \quad \in \mathbb{R}^{3\times 4}, \tag{1.7}$$

where $R_i \in \mathbb{R}^{3\times 3}, i = 2, \ldots, m$ represent the first three columns of $\Pi_i$ and $T_i \in \mathbb{R}^3, i = 2, \ldots, m$ is the fourth column of $\Pi_i$. Although we have used the suggestive notation $(R_i, T_i)$ here, they are not necessarily the actual rotation and translation. Only in the case when the camera is perfectly calibrated does $R_i$ correspond to actual camera rotation and $T_i$ to translation.

With the above notation, we eliminate $\mathbf{x}_1$ from the first row of $N_p$ using column manipulation. It is easy to see that $N_p$ has the same rank as the following matrix in $\mathbb{R}^{3m\times(m+4)}$:

$$\begin{bmatrix} I & 0 & 0 & 0 & \cdots & 0 \\ R_2 & T_2 & R_2\mathbf{x}_1 & \mathbf{x}_2 & \ddots & \vdots \\ \vdots & \vdots & \vdots & 0 & \ddots & 0 \\ R_m & T_m & R_m\mathbf{x}_1 & 0 & 0 & \mathbf{x}_m \end{bmatrix} = \left[\begin{array}{c|c} I & 0 \\ \hline R_2 & \\ \vdots & N_p' \\ R_m & \end{array}\right].$$

Hence, the original $N_p$ is rank deficient if and only if the sub-matrix $N_p' \in \mathbb{R}^{3(m-1)\times(m+1)}$ is.

Multiplying $N_p'$ on the left by the following matrix[3]

$$D_p = \begin{bmatrix} \mathbf{x}_2^T & 0 & \cdots & 0 \\ \widehat{\mathbf{x}_2} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \mathbf{x}_m^T \\ 0 & \cdots & 0 & \widehat{\mathbf{x}_m} \end{bmatrix} \in \mathbb{R}^{4(m-1)\times 3(m-1)}$$

[1] In a projective setting, this won't be a problem since all points on the same line through the camera center are equivalent. But it does matter if we want to choose appropriate reference frames such that the depth for the recovered 3-D point is physically meaningful, *i.e.*, positive.

[2] Depending on the context, the reference frame could be either an Euclidean, Affine or Projective reference frame. In any case, the projection matrix for the first image becomes $[I, 0] \in \mathbb{R}^{3\times 4}$.

[3] For $u \in \mathbb{R}^3$, we use $\widehat{u} \in \mathbb{R}^{3\times 3}$ to denote the skew symmetric generating the cross product, *i.e.*, for any vector $v \in \mathbb{R}^3$, we have $\widehat{u}v = u \times v$.

yields the following matrix $D_p N_p'$ in $\mathbb{R}^{4(m-1)\times(m+1)}$

$$\begin{bmatrix} \mathbf{x}_2^T T_2 & \mathbf{x}_2^T R_2 \mathbf{x}_1 & \mathbf{x}_2^T \mathbf{x}_2 & 0 & 0 & 0 \\ \widehat{\mathbf{x}_2} T_2 & \widehat{\mathbf{x}_2} R_2 \mathbf{x}_1 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & 0 & \ddots & 0 & 0 \\ \vdots & \vdots & 0 & 0 & \ddots & 0 \\ \mathbf{x}_m^T T_m & \mathbf{x}_m^T R_m \mathbf{x}_1 & 0 & 0 & 0 & \mathbf{x}_m^T \mathbf{x}_m \\ \widehat{\mathbf{x}_m} T_m & \widehat{\mathbf{x}_m} R_m \mathbf{x}_1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Since $D_p$ is of full rank $3(m-1)$, we have $\text{rank}(N_p') = \text{rank}(D_p N_p')$. Hence the original matrix $N_p$ is rank deficient if and only if the following sub-matrix of $D_p N_p'$ is rank deficient:

$$M_p = \begin{bmatrix} \widehat{\mathbf{x}_2} T_2 & \widehat{\mathbf{x}_2} R_2 \mathbf{x}_1 \\ \widehat{\mathbf{x}_3} T_3 & \widehat{\mathbf{x}_3} R_3 \mathbf{x}_1 \\ \vdots & \vdots \\ \widehat{\mathbf{x}_m} T_m & \widehat{\mathbf{x}_m} R_m \mathbf{x}_1 \end{bmatrix} \in \mathbb{R}^{3(m-1)\times 2}. \tag{1.8}$$

We call $M_p$ the *multiple view matrix* associated to a point feature $p$. More precisely, we have proven the following:

**Theorem 1 (Rank deficiency equivalence condition).** *Matrices $N_p$ and $M_p$ satisfy*

$$\boxed{\textit{rank}(M_p) = \textit{rank}(N_p) - (m+2) \leq 1.} \tag{1.9}$$

*Therefore rank$(N_p)$ is either $m+3$ or $m+2$, depending on whether rank$(M_p)$ is 1 or 0, respectively.*

**Comment 1 (Geometric interpretation of $M_p$).** *Notice that $\widehat{\mathbf{x}}_i T_i$ is the normal to the epipolar plane given by frames 1 and $i$ and so is $\widehat{\mathbf{x}}_i R_i \mathbf{x}_1$. Therefore the rank deficiency condition not only implies that these two normals are parallel (as obvious from the epipolar constraint) but also that the scale between these two possible normal vectors is the same for all frames.*

Since the rank deficiency of $M_p$ is equivalent to that of $N_p$, we conclude that the rank deficiency of $M_p$ is equivalent to all bilinear, trilinear or quadrilinear constraints among the given $m$ images. To see this more explicitly, notice that for $M_p$ to be rank deficient, it is necessary for the any pair of the vectors $\widehat{\mathbf{x}}_i T_i, \widehat{\mathbf{x}}_i R_i \mathbf{x}_1$ to be linearly dependent. This gives us the well-known bilinear (or epipolar) constraints

$$\mathbf{x}_i^T \widehat{T}_i R_i \mathbf{x}_1 = 0. \tag{1.10}$$

between the $i^{th}$ and $1^{st}$ images. Hence the constraint $\text{rank}(M_p) \leq 1$ consistently generalizes the epipolar constraint (for 2 views) to arbitrary $m$ views.

It is also easy to prove the following linear algebraic fact:

**Lemma 1.** *Given* non-zero *vectors $a_1, \ldots, a_n, b_1, \ldots, b_n \in \mathbb{R}^3$, the following matrix is rank deficient*

$$\begin{bmatrix} a_1 & b_1 \\ \vdots & \vdots \\ a_n & b_n \end{bmatrix} \in \mathbb{R}^{3n\times 2} \tag{1.11}$$

*if and only if $a_i b_j^T - b_i a_j^T = 0$ for all $i, j = 1, \ldots, n$.*

Applying this to the matrix $M_p$ in (1.8), we obtain

$$\widehat{\mathbf{x}}_i (T_i \mathbf{x}_1^T R_j^T - R_i \mathbf{x}_1 T_j^T) \widehat{\mathbf{x}_j} = 0. \tag{1.12}$$

Note that this is a matrix equation and it gives a total of $3 \times 3 = 9$ scalar (trilinear) equations in terms of $\mathbf{x}_1, \mathbf{x}_i, \mathbf{x}_j$. These 9 equations are exactly the 9 trilinear constraints that one would obtain from the minors of $N_p$ following the conventional derivation of trilinear constraints.

Notice that the multiple view matrix $M_p$ being rank deficient is equivalent to all its $2 \times 2$ minors having determinant zero. Since the $2 \times 2$ minors involve three images only, we conclude the following:

- There is *no more* relationship among images among any four views. Hence, the so-called *quadrilinear constraints* [22, 27] do not impose any new or independent constraints on the four images other than the trilinear and bilinear constraints. This has indirectly proved that the quadrifocal tensors can be factorized into trifocal tensors or bifocal tensors.[4]

- Trilinear constraints (1.12) in general implies bilinear constraints (1.10), except when $\widehat{\mathbf{x}}_i T_i = \widehat{\mathbf{x}}_i R_i \mathbf{x}_1 = 0$ for some $i$. This corresponds to a degenerate case in which the pre-image $p$ lies on the line through optical centers $o_1, o_i$.

So far we have essentially given a much more simplified proof for the following facts regarding multilinear constraints among multiple images:

**Theorem 2 (Linear relationships among multiple views of a point).** *For any given $m$ images corresponding to a point $p \in \mathbb{E}^3$ relative to $m$ camera frames, the rank deficient matrix $M_p$ yields the following:*

1. *Any algebraic constraints among the $m$ images can be reduced to only those involving 2 and 3 images at a time. Formulae of these bilinear and trilinear constraints are given by (1.10) and (1.12) respectively.*

2. *For given $m$ images of a point, all the triple-wise trilinear constraints algebraically imply all pairwise bilinear constraints, except in the degenerate case in which the pre-image $p$ lies on the line through optical centers $o_1, o_i$ for some $i$.*

**Comment 2 (Rank condition v.s. multilinear constraints).** *Our discussion implies that multilinear constraints are certainly necessary for the rank of matrix $M_p$ (hence $N_p$) to be deficient. But, rigorously speaking, they are* **not** *sufficient. According to Lemma 1, for multilinear constraints to be equivalent to the rank deficiency condition, vectors in the $M_p$ matrix need to be non-zero. This is not always true for certain degenerate cases, as mentioned above. Hence, rank deficiency condition on $M_p$ is a much better mathematical way of capturing* all *constraints among multiple images and avoids artificial degeneracy that could be introduced by using algebraic equations. On the other hand, since such degeneracy is rare, in a loose term, we may say these two ways are "equivalent".*

Even though we have shown that in most situations trilinear constraints give a sufficient set of constraints, in practice they are relatively more difficult to use than the bilinear constraints. Any trilinear relationship requires matching feature points across three images instead two for the bilinear one. Given that feature matching is a rather difficult and time-consuming problem itself, in many practical applications, bilinear (epipolar) constraints are very much favored over trilinear constraints.

## 1.4 Geometric interpretation of the rank deficiency condition

In the previous section, we have classified all algebraic constraints that may arise among $m$ corresponding images of a point. We now know that the relationship among $m$ images essentially boils down to that between 2 and 3 views at a time, characterized by the bilinear constraint (1.10) and trilinear constraint (1.12) respectively. But we have not yet explained what these equations mean and whether there is a simpler intuitive geometric explanation for all these algebraic relationships. We now try to do that rigorously here without using any heavy machinery from algebraic geometry. Our final goal here is to see how all the analysis and results (including those for degenerate cases) can be captured by an extremely simple statement based on the rank of the $M_p$ matrix.

### 1.4.1 Uniqueness of pre-image

Given 3 vectors $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3 \in \mathbb{R}^3$, if they are indeed images of some 3-D point $p$ with respect to the three camera frames as shown in Figure 1.1, they should automatically satisfy both the bilinear and trilinear

[4] Although we only proved it for the special case with $\Pi_1 = [I, 0]$, the general case differs from this special one only by a choice of a reference frame.

constraints, e.g.

$$\begin{aligned} \text{bilinear:} \quad & \mathbf{x}_2^T \widehat{T_2} R_2 \mathbf{x}_1 = 0, \quad \mathbf{x}_3^T \widehat{T_3} R_3 \mathbf{x}_1 = 0, \\ \text{trilinear:} \quad & \widehat{\mathbf{x}_2}(T_2 \mathbf{x}_1^T R_3^T - R_2 \mathbf{x}_1 T_3^T)\widehat{\mathbf{x}_3} = 0. \end{aligned}$$

Now we ask ourselves the inverse problem: If the three vectors $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ satisfy either the bilinear constraints or trilinear constraints, are they necessarily images of some single point in 3-D, the so-called *pre-image*?

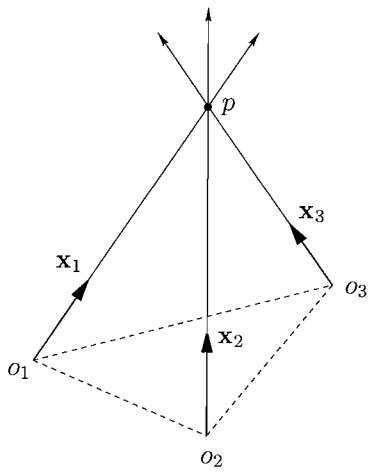


Figure 1.1: Three rays extended from the three images $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ intersect at one point $p$ in 3-D, the pre-image of $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$.

Let us first study whether bilinear constraints are sufficient to determine a unique pre-image in 3-D. For the given three vectors $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$, suppose that they satisfy three pairwise epipolar constraints

$$\mathbf{x}_2^T F_{21} \mathbf{x}_1 = 0, \quad \mathbf{x}_3^T F_{31} \mathbf{x}_1 = 0, \quad \mathbf{x}_3^T F_{32} \mathbf{x}_2 = 0, \tag{1.13}$$

with $F_{ij} = \widehat{T_{ij}} R_{ij}$ the fundamental matrix between the $i^{th}$ and $j^{th}$ images. Note that each image (as a point on the image plane) and the corresponding optical center uniquely determine a line in 3-D that passes through them. This gives us a total of three lines. Geometrically, the three epipolar constraints simply imply that each pair of the three lines are coplanar. So when do three pairwise coplanar lines intersect at exactly one point in 3-D? If these three lines are not coplanar, the intersection is uniquely determined, so is the pre-image. If all of them do lie on the same plane, such a unique intersection is not always guaranteed. As shown in Figure 1.2, this may occur when the lines determined by the images lie on the plane spanned by the three optical centers $o_1, o_2, o_3$, the so-called *trifocal plane*, or when the three optical centers lie on a straight line regardless of the images, the so-called *rectilinear motion*. The first case is of less practical effect since

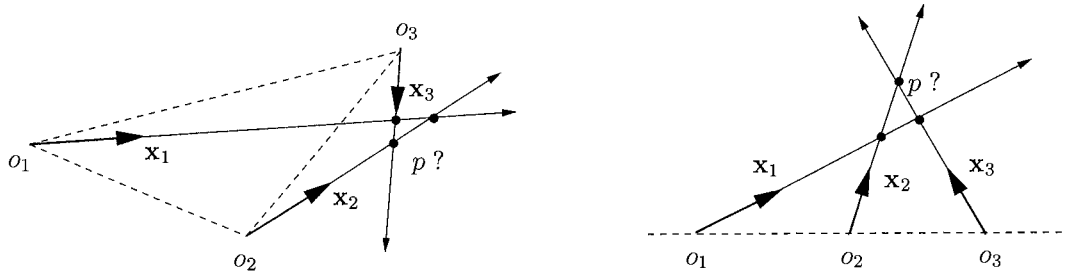


Figure 1.2: Two cases when the three lines determined by the three images $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ lie on the same plane, in which case they may not necessarily intersect at a unique point $p$.

3-D points generically do not lie on the trifocal plane. The second case is more important: Regardless of

what 3-D feature points one chooses, pairwise epipolar constraints alone do not provide sufficient constraints to determine a unique 3-D point from any given three image vectors. In such a case, extra constraints need to be imposed on the three images in order to obtain a unique pre-image.

Would trilinear constraints suffice to salvage the situation? The answer to this is yes and let us show why. Given any three vectors $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3 \in \mathbb{R}^3$, suppose they satisfy the trilinear constraint equation

$$\widehat{\mathbf{x}_2}(T_2\mathbf{x}_1^T R_3^T - R_2\mathbf{x}_1 T_3^T)\widehat{\mathbf{x}_3} = 0.$$

In order to determine $\mathbf{x}_3$ uniquely (up to a scale) from this equation, we need the matrix

$$\widehat{\mathbf{x}_2}(T_2\mathbf{x}_1^T R_3^T - R_2\mathbf{x}_1 T_3^T) \in \mathbb{R}^{3\times 3}$$

to be of exact rank 1. The only case that $\mathbf{x}_3$ is undetermined is when this matrix is rank 0, that is

$$\widehat{\mathbf{x}_2}(T_2\mathbf{x}_1^T R_3^T - R_2\mathbf{x}_1 T_3^T) = 0.$$

That is

$$\text{range}(T_2\mathbf{x}_1^T R_3^T - R_2\mathbf{x}_1 T_3^T) \subset \text{span}\{\mathbf{x}_2\}. \tag{1.14}$$

If $T_3$ and $R_3\mathbf{x}_1$ are linearly independent, then (1.14) holds if and only if the vectors $R_2\mathbf{x}_1, T_2, \mathbf{x}_2$ are linearly dependent. This condition simply means that the line associated to the first image $\mathbf{x}_1$ coincide with the line determined by the optical centers $o_1, o_2$.[5] If $T_3$ and $R_3\mathbf{x}_1$ are linearly dependent, $\mathbf{x}_3$ is determinable since $R_3\mathbf{x}_1$ lies on the line determined by the optical centers $o_1, o_3$. Hence we have shown, that $\mathbf{x}_3$ cannot be uniquely determined from $\mathbf{x}_1, \mathbf{x}_2$ by the trilinear constraint if and only if

$$\widehat{T_2}R_2\mathbf{x}_1 = 0, \quad \text{and} \quad \widehat{T_2}\mathbf{x}_2 = 0. \tag{1.15}$$

Due to the symmetry of the trilinear constraint equation, $\mathbf{x}_2$ is not uniquely determined from $\mathbf{x}_1, \mathbf{x}_3$ by the trilinear constraint if and only if

$$\widehat{T_3}R_3\mathbf{x}_1 = 0, \quad \text{and} \quad \widehat{T_3}\mathbf{x}_3 = 0. \tag{1.16}$$

We still need to show that these three images indeed determine a unique pre-image in 3-D if either one of the images can be determined from the other two by the trilinear constraint. This is obvious. Without loss of generality, suppose it is $\mathbf{x}_3$ that can be uniquely determined from $\mathbf{x}_1$ and $\mathbf{x}_2$. Simply take the intersection $p' \in \mathbb{E}^3$ of the two lines associated to the first two images and project it back to the third image plane – such intersection exists since the two images satisfy epipolar constraint.[6] Call this image $\mathbf{x}_3'$. Then $\mathbf{x}_3'$ automatically satisfies the trilinear constraint. Hence $\mathbf{x}_3' = \mathbf{x}_3$ due to its uniqueness. Therefore $p'$ is the 3-D point $p$ where all the three lines intersect in the first place. As we have argued before, the trilinear constraint (1.12) actually implies bilinear constraint (1.10). Therefore the 3-D pre-image $p$ is uniquely determined if either $\mathbf{x}_3$ can be determined from $\mathbf{x}_1, \mathbf{x}_2$ or $\mathbf{x}_2$ can be determined from $\mathbf{x}_1, \mathbf{x}_3$. So we have in fact proven the following known fact:

**Lemma 2 (Properties of bilinear and trilinear constraints).** *Given three vectors $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3 \in \mathbb{R}^3$ and three camera frames with distinct optical centers. If the three images satisfy pairwise epipolar constraints*

$$\mathbf{x}_i^T \widehat{T_{ij}} R_{ij}\mathbf{x}_j = 0, \quad i, j = 1, 2, 3,$$

*a unique pre-image $p$ is determined except when the three lines associated to the three images are coplanar. If these vectors satisfy all triple-wise trilinear constraints*[7]

$$\widehat{\mathbf{x}_j}(T_{ji}\mathbf{x}_i^T R_{ki}^T - R_{ji}\mathbf{x}_i T_{ki}^T)\widehat{\mathbf{x}_k} = 0, \quad i, j, k = 1, 2, 3,$$

*then they determine a unique pre-image $p \in \mathbb{E}^3$ except when the three lines associated to the three images are collinear.*

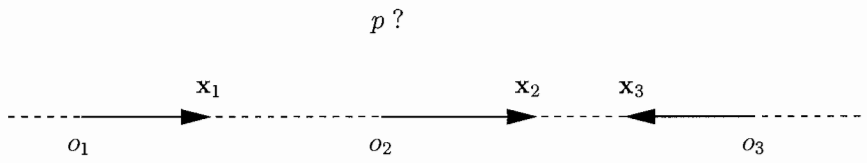


Figure 1.3: If the three images and the three optical centers lie on a straight line, any point on this line is a valid pre-image that satisfies all the constraints.

The two cases (which are essentially one case) in which the bilinear constraints may become degenerate are shown in Figure 1.2. Figure 1.3 shows the only case in which trilinear constraint may become degenerate. In simple terms, "bilinear fails for coplanar and trilinear fails for collinear". For more than 3 views, in order to check the uniqueness of the pre-image, one needs to apply the above lemma to every pairwise or triple-wise views. The possible number of combinations of degenerate cases make it very hard to draw any consistent conclusion. However, in terms of the rank condition on the multiple view matrix, Lemma 2 can be generalized to multiple views in a much more concise and unified way:

**Theorem 3 (Uniqueness of pre-image).** *Given $m$ vectors on the image planes with respect to $m$ camera frames, they correspond to the same point in the 3-D space if the maximum rank of the $M_p$ matrix relative to all the camera frames is of rank 1. If its rank is always 0, the point is determined up to the line where all the camera centers must lie on.*

Hence both the largest and smallest singular values of $M_p$ have meaningful geometric interpretation: the smallest being zero is necessary for a unique pre-image; the largest being zero is necessary for a non-unique pre-image.

### 1.4.2 Geometry of the multiple view matrix

Now we are ready to discover the interesting geometry that the matrix $M_p$ really captures. From the equation

$$\lambda_i \mathbf{x}_i = \lambda_1 R_i \mathbf{x}_1 + T_i, \quad i = 2, \ldots, m, \tag{1.17}$$

it is direct to see that

$$M_p \begin{bmatrix} 1 \\ \lambda_1 \end{bmatrix} = 0. \tag{1.18}$$

So the coefficient that relates the two columns of the $M_p$ matrix is simply the depth of the point $p$ in 3-D space with respect to the center of the first camera frame (the reference).[8] Hence, from the $M_p$ matrix alone, we may know the distance from the point $p$ to the camera center $o_1$. One can further prove that for any two points of the same distance from $o_1$, there always exist a set of camera frames for each point such that their images give exactly the same $M_p$ matrix.[9] Hence we can interpret $M_p$ as a map from a point in 3-D space to a scalar

$$M_p : p \in \mathbb{R}^3 \mapsto d \in \mathbb{R}_+,$$

where $d = \|p - o_1\|$. This map is certainly surjective but not injective. Points that may give rise to the same $M_p$ matrix hence lie on a sphere $\mathbb{S}^2$ centered around the camera center $o_1$. The above scalar $d$ is exactly the radius of the sphere. We may summarize our discussion into the following theorem:

[5] In other words, the pre-image point $p$ lies on the epipole between the first and second camera frames.

[6] If these two lines are parallel, we take the intersection in the plane at infinity.

[7] Although there seem to be total of nine possible $(i, j, k)$, there are in fact only three different trilinear constraints due to the symmetry in the trilinear constraint equation.

[8] Here we implicitly assume that the image surface is a sphere with radius 1. If it is a plane instead, statements below should be interpreted accordingly.

[9] We omit the detail of the proof here for simplicity.

**Theorem 4 (Geometry of the multiple view matrix $M_p$).** *The matrix $M_p$ associated to a point $p$ in 3-D maps this point to a unique scalar $d$. This map is surjective but not injective and two points may give rise to the same $M_p$ matrix if and only if they are of the same distance to the center $o_1$ of the reference camera frame. That distance is given by the coefficients which relate the two columns of $M_p$.*

Therefore, knowing $M_p$ (*i.e.*, the distance $d$), we know that $p$ must lie on a sphere of radius $d = \|p - o_1\|$. Given three camera frames, we can choose either camera center as the reference, then we essentially have three $M_p$ matrix for each point. Each $M_p$ matrix gives a sphere around each camera center in which the point $p$ should stay. The intersection of two such spheres in 3-D space is generically a circle, as shown in Figure 1.4. Then one would imagine that, in general, the intersection of all three spheres determines the 3-D location of the point $p$ up to two solutions, unless all the camera centers lie on the same line as $o_1, o_2$ (*i.e.*, except for the rectilinear motion). In the next chapter which studies rank deficiency condition for a line, we

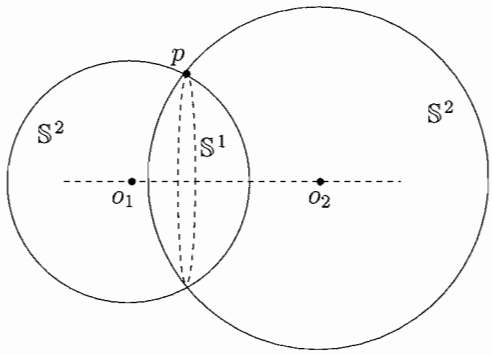


Figure 1.4: The rank deficient matrices $M_p$ of a point relative to two distinct camera centers determine the point $p$ up to a circle.

further show that there are some profound relationships between the $M_p$ matrix for a point and that for a line.

## 1.5 Applications of the rank deficiency condition

The rank deficiency condition of the multiple view matrix $M_p$ allows us to use all the constraints among multiple images simultaneously for purposes such as feature matching or motion recovery, without specifying a particular set of pairwise or triple-wise frames. Ideally, one would like to formulate the entire problem of reconstructing 3-D motion and structure as one optimizing some global objective function[10] subject to the rank deficiency condition. However, such an approach usually relies on very difficult nonlinear optimization methods. Instead, we here divide the overall problem into a few subproblems: matching features assuming known motion, and estimating motion assuming known matched features.

### 1.5.1 Multiple view matching test for point features

Notice that $M_p \in \mathbb{R}^{3(m-1)\times 2}$ being rank deficient is equivalent to the determinant of $M_p^T M_p \in \mathbb{R}^{2\times 2}$ being zero:

$$\det(M_p^T M_p) = 0. \tag{1.19}$$

In general $M_p^T M_p$ is a function of the projection matrix $\Pi$ and images $\mathbf{x}_1, \ldots, \mathbf{x}_m$. If $\Pi$ is known and we like to test if given $m$ vectors $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathbb{R}^3$ indeed satisfy all the constraints that $m$ images of a single 3-D pre-image should, we only need to test if the above determinant is zero. A more numerically robust algorithm would be:

[10] Such as the so-called *reprojection error* in image.

**Algorithm 1 (Multiple view matching test).** *Suppose the projection matrix $\Pi$ associated to $m$ camera frames are given. Then for given $m$ vectors $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathbb{R}^3$,*

1. *Compute the matrix $M_p \in \mathbb{R}^{3(m-1)\times 2}$ according to (1.8).*
2. *Compute second eigenvalue $\lambda_2$ of $M_p^T M_p$;*
3. *If $\lambda_2 \leq \epsilon$ for some pre-fixed threshold, the $m$ image vectors match.*

### 1.5.2 Multiple view motion estimation from point features

Now suppose that $m$ images $\mathbf{x}_1^i, \ldots, \mathbf{x}_m^i$ of $n$ points $p^i$, $i = 1, \ldots, n$ are given and we want to use them to estimate the unknown projection matrix $\Pi$. The rank deficiency condition of the $M_p$ matrix can be written as:

$$\alpha^i \begin{bmatrix} \widehat{\mathbf{x}_2^i} T_2 \\ \widehat{\mathbf{x}_3^i} T_3 \\ \vdots \\ \widehat{\mathbf{x}_m^i} T_m \end{bmatrix} + \begin{bmatrix} \widehat{\mathbf{x}_2^i} R_2 \mathbf{x}_1^i \\ \widehat{\mathbf{x}_3^i} R_3 \mathbf{x}_1^i \\ \vdots \\ \widehat{\mathbf{x}_m^i} R_m \mathbf{x}_1^i \end{bmatrix} = 0 \in \mathbb{R}^{3(m-1)\times 1}, \tag{1.20}$$

for proper $\alpha^i \in \mathbb{R}, i = 1, \ldots n$.

From (1.1) we have $\lambda_j^i \mathbf{x}_j^i = \lambda_1^i R_j \mathbf{x}_1^i + T_j$. Multiplying by $\widehat{\mathbf{x}_j^i}$ we obtain $\widehat{\mathbf{x}_j^i}(R_j \mathbf{x}_1^i + T_j/\lambda_1^i) = 0$. Therefore $\alpha^i = 1/\lambda_1^i$ can be interpreted as the inverse of the depth of point $p^i$ with respect to the first frame. The set of equations in (1.20) is equivalent to finding vectors $\vec{R}_j = [r_{11}, r_{12}, r_{13}, r_{21}, r_{22}, r_{23}, r_{31}, r_{32}, r_{33}]^T \in \mathbb{R}^9$ and $\vec{T}_j = T_j \in \mathbb{R}^3$, $j = 2, \ldots, m$, such that:

$$P_j \begin{bmatrix} \vec{T}_j \\ \vec{R}_j \end{bmatrix} = \begin{bmatrix} \alpha^1 \widehat{\mathbf{x}_j^1} & \widehat{\mathbf{x}_j^1} * \mathbf{x}_1^{1T} \\ \alpha^2 \widehat{\mathbf{x}_j^2} & \widehat{\mathbf{x}_j^2} * \mathbf{x}_1^{2T} \\ \vdots & \vdots \\ \alpha^n \widehat{\mathbf{x}_j^n} & \widehat{\mathbf{x}_j^n} * \mathbf{x}_1^{nT} \end{bmatrix} \begin{bmatrix} \vec{T}_j \\ \vec{R}_j \end{bmatrix} = 0 \in \mathbb{R}^{3n}, \tag{1.21}$$

where $A * B$ is the *Kronecker product* of $A$ and $B$. It can be shown that if $\alpha^i$'s are known, the matrix $P_j$ is of rank 11 if more than $n \geq 6$ points in general position are given. In that case, the kernel of $P_j$ is unique, and so is $(R_j, T_j)$.

#### Euclidean reconstruction

For simplicity, we here assume that the camera is perfectly calibrated. Therefore, is $A(t) = I$, $R_i$ is a rotation matrix in $SO(3)$ and $T_i$ is a translation vector in $\mathbb{R}^3$. Given the first two images of (at least) eight points in general configuration, $T_2 \in \mathbb{R}^3$ and $R_2 \in SO(3)$ can be estimated using the classic *eight point algorithm*. The equation given by the first row in (1.20) implies $\alpha^i \widehat{\mathbf{x}_2^i} T_2 = -\widehat{\mathbf{x}_2^i} R_2 \mathbf{x}_1^i$, whose least squares solution up to scale (recall that $T_2$ is recovered up to scale from the eight point algorithm) is given by:

$$\alpha^i = -\frac{(\widehat{\mathbf{x}_2^i} T_2)^T \widehat{\mathbf{x}_2^i} R_2 \mathbf{x}_1^i}{||\widehat{\mathbf{x}_2^i} T_2||^2}, \quad i = 1, \ldots, n. \tag{1.22}$$

These values of $\alpha^i$ can therefore be used to initialize the equation (1.20). Multiplying on the left the $j$-th block of (1.20) by $T_j^T$ yields the epipolar constraints $\mathbf{x}_j^{iT} \widehat{T_j} R_j \mathbf{x}_1^i = 0$. We know that in general the solution $(R_j, T_j)$ of these equations is unique, with $T_j$ recovered up to scale. Hence the solution to (1.20) is unique and we can then recover from (1.20) the scale of each $T_j$ up to a single scale for all the $T_j$'s (recall that the $\alpha^i$'s were computed up to scale). Since the $\alpha^i$'s are known, (1.21) becomes a set of linear equations on $\vec{T}_j$ and $\vec{R}_j$, whose solution can be described as follows.

Let $\tilde{T}_j \in \mathbb{R}^3$ and $\tilde{R}_j \in \mathbb{R}^{3\times 3}$ be the (unique) solution of (1.21). Such a solution is obtained as the eigenvector of $P_j$ associated to the smallest singular value. Let $\tilde{R}_j = U_j S_j V_j^T$ be the SVD of $\tilde{R}_j$. Then the solution of (1.21) in $\mathbb{R}^3 \times SO(3)$ is given by:

$$T_j = \frac{\text{sign}(\det(U_j V_j^T))}{\sqrt[3]{\det(S_j)}} \tilde{T}_j \in \mathbb{R}^3, \tag{1.23}$$

$$R_j = \text{sign}(\det(U_j V_j^T))\, U_j V_j^T \in SO(3). \tag{1.24}$$

In the presence of noise, solving for $\alpha^i$ using only the first two frames may not necessarily be the best thing to do. Nevertheless, this arbitrary choice of $\alpha^i$ allows us to compute all the motions $(R_j, T_j)$, $j = 2, \ldots, m$. Given all the motions, the least squares solution for $\alpha^i$ from (1.20) is given by:

$$\alpha^i = -\frac{\sum_{j=2}^m (\widehat{\mathbf{x}_j^i} T_j)^T \widehat{\mathbf{x}_j^i} R_j \mathbf{x}_1^i}{\sum_{j=2}^m \|\widehat{\mathbf{x}_j^i} T_j\|^2}, \; i = 1, \ldots, n. \tag{1.25}$$

Note that these $\alpha^i$'s are the same as those in (1.22) if $m = 2$. One can then recompute the motion given this new $\alpha^i$'s, until the error in $\alpha$ is small enough.

We then have the following linear algorithm for multiple view motion and structure estimation:

**Algorithm 2 (Multiple view six-eight point algorithm).** *Given $m$ images $\mathbf{x}_1^i, \ldots, \mathbf{x}_m^i$ of $n$ points $p^i$, $i = 1, \ldots, n$, estimate the motions $(R_j, T_j)$, $j = 2, \ldots, m$ as follows:*

1. *Initialization: $k = 0$*
   (a) *Compute $(R_2, T_2)$ using the eight point algorithm for the first two views.*
   (b) *Compute $\alpha_k^i = \alpha^i / \alpha^1$ from (1.22).*
2. *Compute $(\tilde{R}_j, \tilde{T}_j)$ as the eigenvector associated to the smallest singular value of $P_j$, $j = 2, \ldots, m$.*
3. *Compute $(R_j, T_j)$ from (1.23) and (1.24) for $j = 2, \ldots, m$.*
4. *Compute the new $\alpha^i = \alpha_{k+1}^i$ from (1.25). Normalize so that $\alpha_{k+1}^1 = 1$.*
5. *If $\|\alpha_k - \alpha_{k+1}\| > \epsilon$, for a pre-specified $\epsilon > 0$, then $k = k + 1$ and goto 2. Else stop.*

The camera motion is then $(R_j, T_j), j = 2, \ldots, m$ and the structure of the points (with respect to the first camera frame) is given by the converged depth scalar $\lambda_1^i = 1/\alpha^i, i = 1, \ldots, n$. There are a few notes for the proposed algorithm:

1. It makes use of all multilinear constraints simultaneously for motion and structure estimation. $(R_j, T_j)$'s seem to be estimated using pairwise views only but that is not exactly true. The computation of the matrix $P_j$ depends on all $\alpha^i$ each of which is in turn estimated from the $M_p^i$ matrix involving all views. The reason to set $\alpha_{k+1}^1 = 1$ is to fix the universal scale. It is equivalent to putting the first point at a distance of 1 to the first camera center.
2. It can be used either in a batch fashion or a recursive one: initializes with two views, recursively estimates camera motion and automatically updates scene structure when new data arrive.
3. It may effectively eliminate the effect of occluded feature points - if a point is occluded in a particular image, simply drop the corresponding group of three rows in the $M_p$ matrix without affecting the condition on its rank.

**Remark 3 (Euclidean, affine or projective reconstruction).** *We must point out that, although the algorithm seems to be proposed for the calibrated camera (Euclidean) case only, it works just the same if the camera is weakly calibrated (affine) or uncalibrated (projective). The only change needed is at the initialization step. In order to initialize $\alpha^i$'s, either an Euclidean, affine or projective choice for $(R_2, T_2)$ needs to be specified. This is where our knowledge of the camera calibration enters the picture. In the worst case, i.e., when we have no knowledge on the camera at all, any choice of a projective frame will do. In that case, the relative transformation among camera frames and the scene structure are only recovered up to a projective transformation. Once that is done, the rest of the algorithm runs exactly the same.*

## 1.6 Simulations on synthetic data

In this section, we show by simulations the performance of the proposed algorithm. We compare motion and structure estimates with those of the eight point algorithm for two views and then we compare the estimates as a function of the number of frames.

### 1.6.1 Setup

The simulation parameters are as follows: number of trials is 1000, number of feature points is 20, number of frames is 3 or 4 (unless we vary it on purpose), field of view is $90^o$, depth variation is from 100 to 400 units of focal length and image size is $500 \times 500$. Camera motions are specified by their translation and rotation axes. For example, between a pair of frames, the symbol $XY$ means that the translation is along the $X$-axis and rotation is along the $Y$-axis. If $n$ such symbols are connected by hyphens, it specifies a sequence of consecutive motions. The ratio of the magnitude of translation $||T||$ and rotation $\theta$, or simply the $T/R$ *ratio*, is compared at the center of the random cloud scattered in the truncated pyramid specified by the given field of view and depth variation (see Figure 1.5). For each simulation, the amount of rotation between frames is given by a proper angle and the amount of translation is then automatically given by the $T/R$ ratio. We always choose the amount of total motion such that all feature points will stay in the field of view for all frames. In all simulations, independent Gaussian noise with std given in pixels is added to each image point. Error measure for rotation is $\arccos\left(\frac{tr(R\tilde{R}^T)-1}{2}\right)$ in degrees where $\tilde{R}$ is an estimate of the true $R$. Error measure for translation is the angle between $T$ and $\tilde{T}$ in degrees where $\tilde{T}$ is an estimate of the true $T$.

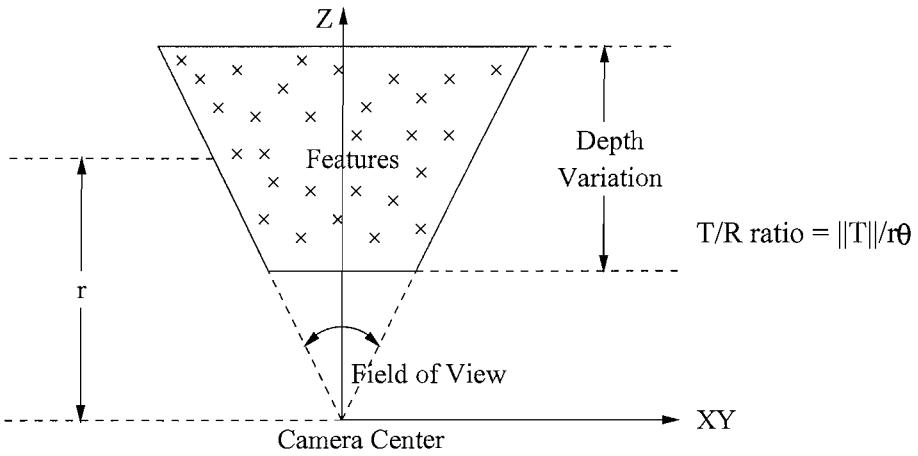


Figure 1.5: Simulation setup

### 1.6.2 Comparison with the 8 point algorithm

Figure 1.6 plots the errors of rotation estimates and translation estimates compared with results from the standard 8-point linear algorithm. Figure 1.7 shows a histogram of the relative translation scale for a noise level of 3 pixels as well as the error on the estimation of the structure. As we see, the multiple view linear algorithm not only generalizes but also outperforms the well-known eight point linear algorithm for two views. This is because the algorithm implicitly uses the estimated structure of the scene for the estimation of the motion, while the eight point algorithm does not.

### 1.6.3 Error as a function of the number of frames

In this simulation, we analyze the effect of the number of frames on motion and structure estimates. In principle, motion estimates should not necessarily improve, since additional frames introduce more data as

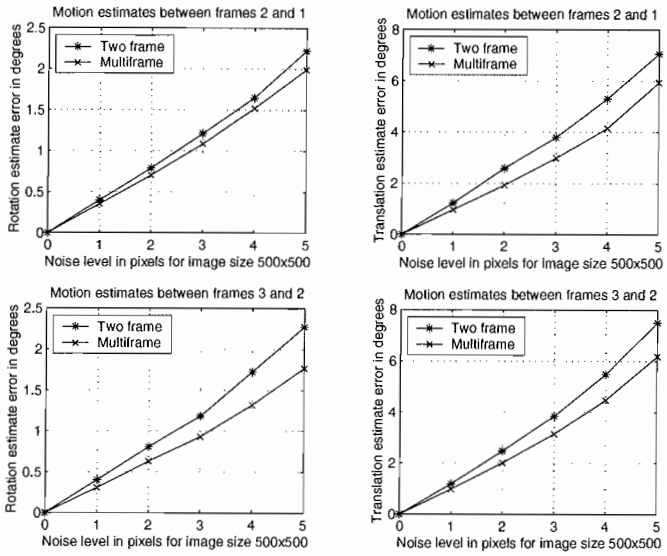


Figure 1.6: Motion estimate error comparison between 8 point algorithm and multiple view linear algorithm.

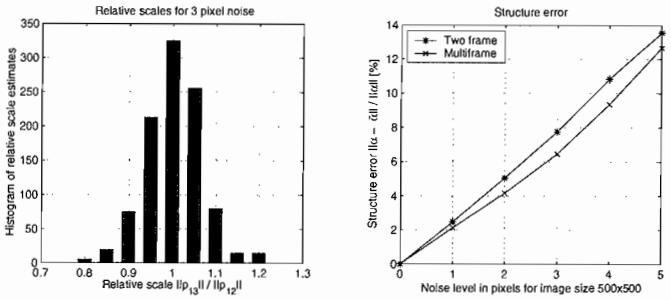


Figure 1.7: Relative scale and structure estimate error comparison. Motion is $XX$-$YY$ and relative scale is 1.

well as more unknowns. However, structure estimates should improve, because additional data is available to estimate the same structure. We consider 7 frames with motion $XX$-$YY$-$X(XY)$-$ZY$-$(XY)Z$-$(YZ)(YZ)$ and plot the estimation errors for the first pair of frames as a function of the number of frames. As expected, that rotation estimates and relative translation scales approximately independent on the number of frames. Translation estimates with three frames are better than those with two frames, but there is no significant improvement for more than three frames. Finally, structure estimates in general improve with the number of frames.

## 1.7 Experiments on real images

We now consider an indoor sequence with the camera undergoing rectilinear motion. We compare the performance of the proposed multiple view linear algorithm against the conventional eight point algorithm for two views and the multiple view nonlinear algorithm in [30]. In order to work with real images, we need to calibrate the camera, track a set of feature points and establish their correspondences across multiple frames. We calibrated the camera from a set of planar feature points using Zhang's technique [34]. For feature tracking and correspondence, we adapted the algorithm from [35].

The multiple view nonlinear algorithm is initialized with estimates from the eight-point linear algorithm. Since the translation estimates of the linear algorithm are given up to scale only, for the multiple view

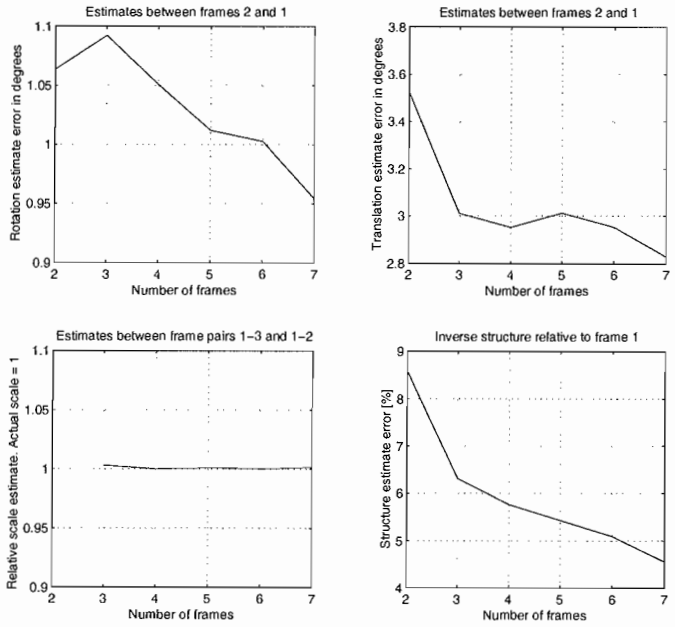


Figure 1.8: Estimation error as a function of the number of frames. Noise level is 3 pixels and relative scale is 1.

case an initialization of the relative scale between consecutive translations is required. This is done by triangulation since the directions of the translations are known. For example, the relative scale between $T_{21}$ and $T_{32}$ is $\sin(\beta)/\sin(\gamma)$ where $\beta$ is the angle between $T_{31}$ and $R_{21}T_{21}$ and $\gamma$ is the angle between $T_{23}$ and $R_{13}T_{13}$. Recall in the multiple view case the vector of translations $\mathcal{T}$ is recovered up to one single scale. The estimated motion is then compared with the ground truth data. Error measures are the same as in the previous section.

We use 4 images of an indoor scene, with the motion of the camera in a straight line (rectilinear motion) along the $Z$-axis (see Figure 1.9). The relative scales between consecutive translations are 2:1 and 1:2, respectively. Even though the motion is rectilinear, relative scales still can be initialized by triangulation, because image measurements are noisy.

Figure 1.10 shows the error between the motion and structure estimates and ground truth. It can be observed that the proposed linear algorithm is able to recover the correct motion and structure, giving better results in most of the cases.

## 1.8 Discussions and conclusions

As we have seen, the rank deficiency condition is indeed a criterion for a consistent 3-D reconstruction of a scene from multiple images. Linear algorithms developed from this condition work reasonably well even in the presence of noises. But such solutions are by no means optimal and, in the end, the rank deficiency condition would not be *exactly* satisfied by the solution either – due to noises in the measurements. Therefore nonlinear methods still need to be deployed in order to obtain a "consistent and optimal" solution for 3-D reconstruction. Our future work would involve how to relax such an algebraic condition by incorporating it into better conditioned optimization problems for 3-D reconstruction. For example, how to minimize the reprojection error subject to such a rank deficiency condition.

One advantage of the rank deficiency condition is that it unifies nicely the point and line cases, as one may soon see in the upcoming chapters. All theory and algorithms for point features and line features run in exact parallel. This allows possible combination of these algorithms and use them together for a consistent 3-D reconstruction. Another advantage of this condition is revealed through the linear algorithms derived from it. It provides a unified treatment to the Euclidean, affine and projective camera models and indicates

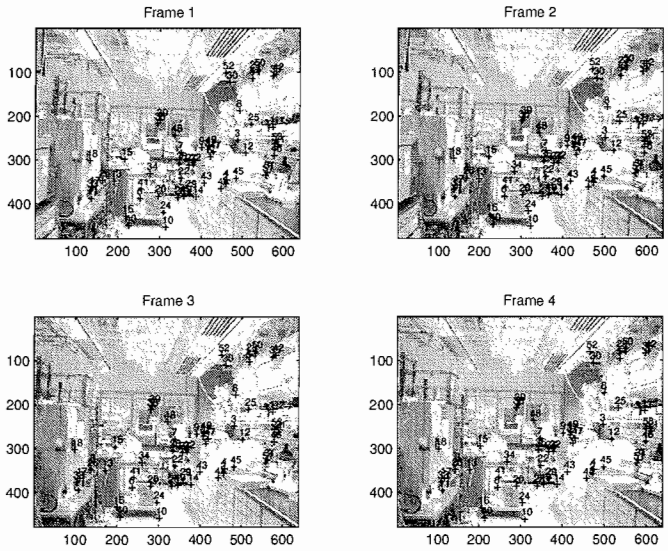


Figure 1.9: Indoor rectilinear motion image sequence

exactly where and when such knowledge on camera should be taken into account for recovery. Furthermore, our study integrates previous study on the two views and three views. In particular, since the non-trivial constraints for corresponding image points start with two views, for the motion recovery, the initialization of the Algorithm 2 uses bilinear constraints. In the case of image lines, the non-trivial constraints start with three views, the initialization of the corresponding algorithm then naturally uses trilinear constraints.

All in all, we believe that the rank deficiency condition provides us a fundamental tool to extend the work since Longuet-Higgins [16] on the two view case for point to a multiple view setting for point, line, curve and even surface. This condition no longer discriminates Euclidean, affine and projective camera models which used to be treated separately in the computer vision literature. Degeneracy or singularity of this condition also corresponds to degenerate configurations for the overall system which consists of the 3-D point or line of interest and all the camera frames. Such a unifying tool may ultimately shed light on an integration of all types of image measurements for the purpose of a consistent reconstruction of 3-D camera motion and scene structure.

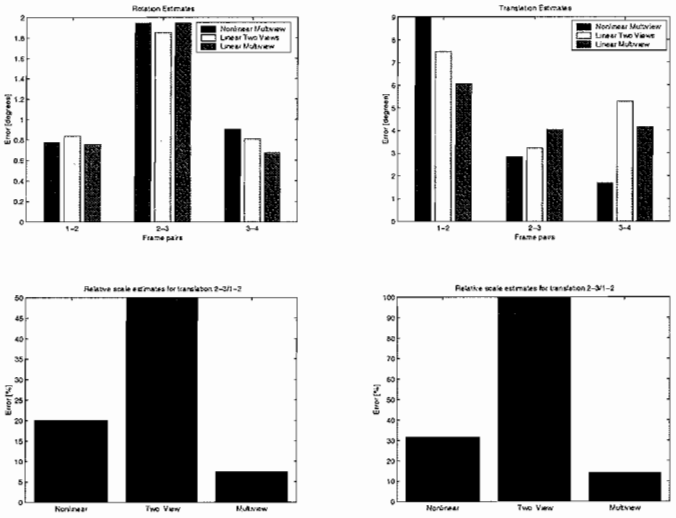


Figure 1.10: Motion and structure estimates for indoor rectilinear motion image sequence

# Chapter 2

# The Multiple View Matrix for Line Features

*Kun Huang, Yi Ma, Jana Košecká*
*Submitted to CVPR01, May 18th.*

## Abstract

*In this chapter, a new rank deficiency condition for multiple images of a line is presented. It is shown that a set of $m$ image lines correspond to a unique 3-D line if and only if an associated multiple view matrix $M_l \in \mathbb{R}^{(m-1)\times 4}$ is of rank 1. This condition is shown to be equivalent to all multilinear constraints among image lines, but it tremendously simplifies previously known derivations. Since rank deficiency is a purely linear algebraic condition, it gives rise to a set of natural linear algorithms for line matching and motion estimation from images of multiple lines. These linear algorithms use all available data simultaneously without specifying a particular choice of image triplets. Hence apart from the initialization, the algorithms allow us to bypass trifocal tensors used for similar purposes. The theory and algorithms for the line case are developed in exact parallel to that for the point case. Geometric interpretation of the $M_l$ matrix and the duality between point and line are also clearly revealed through this approach. We present simulation and experimental results that validate the proposed algorithms.*
**Key words:** Multiple view matrix, multilinear constraints, rank deficiency condition, feature matching, motion recovery.

## 2.1 Introduction

The constraints governing the multiple view geometry of points and lines are the stepping stones for development of algorithms for motion recovery, feature matching and image transfer. Great deal of work has been done in the formulation of multiple view constraints, studying of their dependency and their application for motion recovery, recognition as well as rendering. The multiple view constraints naturally arise as rank deficiency constraints of a particular $m$-view measurement/motion matrix. However the only known forms of these constraints which could be exploited algorithmically were the algebraic multilinear constraints capturing the relationship between projections of elementary features and parameters of the camera motion. Moving beyond two view case, the derivation and interpretation of these constraints utilized tensorial notation and led to complex constrained optimization problems. The clear geometric interpretation present in two view case was lost when considering multiple views.

The main focus of this chapter is the characterization of constraints governing the multiple view geometry of line features. This chapter is a companion to the preceding one on "The Multiple View Matrix for Point Features" and these two chapters together form a unified theoretical and algorithmic treatment of multiple view motion and structure recovery problem. Here we present a self-contained treatment of the line case,

which also serves as a link towards study of multiple view geometry of curves and surfaces. The main contributions of this chapter are:

- A new rank deficiency condition of so-called multiple view matrix $M_l$ for multiple images of a line, which is equivalent to known multilinear constraints but tremendously simplifies their derivation.
- This purely linear algebraic rank deficiency condition leads to a new set of algorithms for motion estimation, line matching and line transfer which consider all available data *simultaneously*, without specifying particular choice of triple-wise images.
- We offer a clear geometric interpretation of matrix $M_l$, which facilitates understanding of the uniqueness of a pre-image of multiple images of a line and critical configurations in multiple view geometry of lines.

The proposed algorithms are to a large extent developed in parallel to those for the point case in the preceding chapter. In particular since a non-trivial epipolar constraint for corresponding image points is present for two views, the initialization of the point based multiple view algorithm uses bilinear constraints for the motion recovery. In the case of image lines, the non-trivial constraint is associated with three views, hence initialization of the algorithm must use three views. Except for the initialization step, which essentially plays the role of choosing a reference frame, the algorithm is "factorizable", *i.e.*, it is based on SVD of all the data *simultaneously*. Hence the algorithm does not depend on a particular choice of a set of pairwise, triple-wise or quadruple-wise image frames.

The rank deficiency condition does not only unifies the theory and algorithms of multiple view geometry for points and lines, but also provides necessary and sufficient conditions for images of a much richer class of 3-D objects. For example, multiple images of a curve in 3-D space must satisfy the rank deficiency condition for corresponding points on and tangents to the image curves. Such constraints would allow us to uniquely determine the exact curve in 3-D space consistent with multiple images. Generally speaking, as a fundamental geometric constraint among multiple images, the rank deficiency condition certainly has a far-reaching impact on both the theoretic and algorithmic aspects of multiple view geometry and its power of generalization does not stop at the case of points or lines.

**Relation to previous work:** This work provides a new perspective on multiple view geometry of line features previously captured by the study of trilinear constraints only. There is an extensive literature with different formulations of multiple view constraints between points and lines as well as study of their dependency [6, 27, 10]. A comprehensive account of the state of the art can be found in recently published monographs [8] and [4]. Initially the constraints governing the relationships between projections of 3-D line and camera motion have been stated and exploited in [23] and later in [32]. The natural linear algorithms which followed from these formulations were versions of 8-point algorithms developed for point features in two frame setting. The linear algorithms first estimated 27 coefficients of the trilinear constraints, followed by their decomposition into respective motion parameters [8]. More general theoretical foundations of multiple view geometry were developed using tensorial notation and revealed that the only non-trivial constraints involve projections of line or point features up to three views [21, 1, 8]. When trilinear tensor was used for matching and image transfer, it was frequently estimated using point features and its coefficients were determined by two view motion estimation techniques based on epipolar geometry [19].

Movement beyond three views was to a large extent motivated by many successful applications of multiple view factorization techniques for structure and motion recovery developed for orthographic projection and point feature setting [26]. Later on, counterparts of factorization for lines and points assuming affine projection model [12, 18] were proposed. Factorization method for a projective reconstruction has also been studied given that relative scales among measurements are properly initialized [28]. Our work deals with a perspective camera but does not discriminate Euclidean, affine or projective camera model.

The presented work provides a new *global* characterization of multiple view geometry for line features in a way unified with point features. While some of the results presented in this chapter may have been known previously, we believe that it is the first time that one can obtain all the known and new results for both point and line features in a concise and unified framework. All the proofs and algorithms use only linear algebra, with no need of tensorial notation or algebraic geometry arguments.

**How this chapter is organized:** Section 2.2 formulates and briefly reviews the origin of constraints among multiple images of a 3-D line $L$. Section 2.3 introduces a new matrix $M_l$ associated with multiple views of

the line and demonstrates that the rank deficiency condition of matrix $M_l$ implies all trilinear constraints between arbitrary three views. In Section 2.4, we also provide a clear geometric interpretation of the rank deficiency condition, state in terms of the $M_l$ matrix the conditions on the uniqueness of pre-image from multiple views, and develop the duality between point and line features. As a natural application of the rank deficiency condition, Section 2.5 outlines two linear algorithms: one for matching line features and another for motion and structure estimation using corresponding line features. Section 2.6 presents simulation results for the proposed algorithms. Section 2.7 concludes the chapter.

## 2.2 Multiple views of a line

Consider a line $L \subset \mathbb{E}^3$, defined by an equation $L = \{\mathbf{X} \mid \mathbf{X} = \mathbf{X}_0 + \lambda v\}$, where $\mathbf{X}_0 = [X_0, Y_0, Z_0, 1]^T \in \mathbb{R}^4$ is a base point on this line, $v = [v_1, v_2, v_3, 0]^T \in \mathbb{R}^4$ is a non-zero vector indicating the direction of the line, and $\lambda \in \mathbb{R}$. An image $\mathbf{l}(t) = [a(t), b(t), c(t)]^T \in \mathbb{R}^3$ of $L$ taken by a moving camera satisfies the following equation

$$\mathbf{l}(t)^T \mathbf{x}(t) = \mathbf{l}(t)^T A(t) P g(t) \mathbf{X} = 0, \tag{2.1}$$

where $\mathbf{x}(t)$ is the image of the point $\mathbf{X} \in L$ at time $t$, $A(t) \in SL(3)$ is the camera calibration matrix (at time $t$), $P = [I, 0] \in \mathbb{R}^{3\times 4}$ is the constant projection matrix and $g(t) \in SE(3)$ is the coordinate transformation from the world frame to the camera frame at time $t$. Note that $\mathbf{l}(t)$ is the normal vector of the plane formed by the optical center $o$ and the line $L$ (in 3-D) at time $t$. Since the above equation holds for any point $\mathbf{X}$ on the line $L$, it yields

$$\mathbf{l}(t)^T A(t) P g(t) \mathbf{X}_0 = \mathbf{l}(t)^T A(t) P g(t) v = 0. \tag{2.2}$$

In the above equations, all $\mathbf{x}, \mathbf{X}$ and $g$ are in homogeneous representation. The reader may have noticed that here we allow the calibration matrix $A$ to change in time. This is partly because such a generalization does not make it any more difficult for the development of the results in this chapter. In addition, it encompasses a richer class of practical situations when some camera intrinsic parameters such as the focal length may indeed change from image to image.

In a realistic situation, we usually only obtain "sampled" images of $\mathbf{l}(t)$ at some time instances: $t_1, t_2, \ldots, t_m$. For simplicity we denote

$$\mathbf{l}_i = \mathbf{l}(t_i), \quad \Pi_i = A(t_i) P g(t_i). \tag{2.3}$$

The matrix $\Pi_i$ is then a $3 \times 4$ matrix which relates the $i^{th}$ image of the line $L$ to its world coordinates $(\mathbf{X}_0, v)$ by

$$\boxed{\mathbf{l}_i^T \Pi_i \mathbf{X}_0 = \mathbf{l}_i^T \Pi_i v = 0} \tag{2.4}$$

for $i = 1, \ldots, m$. In the above equations, except $\mathbf{l}_i$'s, everything else is unknown and subject to recovery. However, solving $\Pi_i$'s and $L$ (*i.e.*, $(\mathbf{X}_0, v)$) simultaneously from these equations is extremely difficult. A natural way to simplify the task is to exploit the rank deficiency condition from the following equations

$$N_l \mathbf{X}_0 = 0, \quad N_l v = 0, \tag{2.5}$$

where

$$N_l = \begin{bmatrix} \mathbf{l}_1^T \Pi_1 \\ \mathbf{l}_2^T \Pi_2 \\ \vdots \\ \mathbf{l}_m^T \Pi_m \end{bmatrix} \in \mathbb{R}^{m\times 4}. \tag{2.6}$$

Hence the rank of $N_l$ must be

$$\boxed{\text{rank}(N_l) \leq 2.} \tag{2.7}$$

## 2.3 The multiple view matrix and its rank

Without loss of generality, we may assume that the first camera frame is chosen to be the reference frame.[1] That gives the projection matrices $\Pi_i, i = 1, \ldots, m$ the general form

$$\Pi_1 = [I, 0], \quad \ldots, \quad \Pi_m = [R_m, T_m] \quad \in \mathbb{R}^{3\times 4}, \tag{2.8}$$

where $R_i \in \mathbb{R}^{3\times 3}, i = 2, \ldots, m$ is the first three columns of $\Pi_i$ and $T_i \in \mathbb{R}^3, i = 2, \ldots, m$ is the fourth column of $\Pi_i$. Although we have used the suggestive notation $(R_i, T_i)$ here, they are not necessarily the actual rotation and translation. $R_i$ could be an arbitrary $3 \times 3$ matrix. Only in the case when the camera is perfectly calibrated does $R_i$ correspond to actual camera rotation and $T_i$ to translation. Now the matrix $N_l$ becomes

$$N_l = \begin{bmatrix} \mathbf{l}_1^T & 0 \\ \mathbf{l}_2^T R_2 & \mathbf{l}_2^T T_2 \\ \vdots & \vdots \\ \mathbf{l}_m^T R_m & \mathbf{l}_m^T T_m \end{bmatrix} \in \mathbb{R}^{m\times 4}. \tag{2.9}$$

This matrix should have a rank of no more than 2. Multiplying $N_l$ on the right by the following matrix[2]

$$D_l = \begin{bmatrix} \widehat{\mathbf{l}_1} & \mathbf{l}_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{4\times 5} \tag{2.10}$$

yields

$$N_l' = \begin{bmatrix} 0 & \mathbf{l}_1^T \mathbf{l}_1 & 0 \\ \mathbf{l}_2^T R_2 \widehat{\mathbf{l}_1} & \mathbf{l}_2^T R_2 \mathbf{l}_1 & \mathbf{l}_2^T T_2 \\ \vdots & \vdots & \vdots \\ \mathbf{l}_m^T R_m \widehat{\mathbf{l}_1} & \mathbf{l}_m^T R_m \mathbf{l}_1 & \mathbf{l}_m^T T_m \end{bmatrix} \in \mathbb{R}^{m\times 5}. \tag{2.11}$$

Then since $D_l$ is of full rank 4, it yields

$$\text{rank}(N_l') = \text{rank}(N_l) \leq 2.$$

Obviously, this is true if and only if the following sub-matrix of $N_l'$

$$M_l = \begin{bmatrix} \mathbf{l}_2^T R_2 \widehat{\mathbf{l}_1} & \mathbf{l}_2^T T_2 \\ \mathbf{l}_3^T R_3 \widehat{\mathbf{l}_1} & \mathbf{l}_3^T T_3 \\ \vdots & \vdots \\ \mathbf{l}_m^T R_m \widehat{\mathbf{l}_1} & \mathbf{l}_m^T T_m \end{bmatrix} \in \mathbb{R}^{(m-1)\times 4} \tag{2.12}$$

has rank no more than one. We call the matrix $M_l$ the *multiple view matrix* associated to a line feature $L$. Hence we have proven the following:

**Theorem 5 (Rank deficiency equivalence condition).** *For the two matrices $N_l$ and $M_l$, we have*

$$\boxed{\textit{rank}(M_l) = \textit{rank}(N_l) - 1 \leq 1.} \tag{2.13}$$

*Therefore rank$(N_l)$ is either 2 or 1, depending on whether rank$(M_l)$ is 1 or 0, respectively.*

[1] Depending on the context, the reference frame could be either a Euclidean, affine or projective reference frame. In any case, the projection matrix for the first image becomes the standard projection matrix $[I, 0] \in \mathbb{R}^{3\times 4}$. The reader should note that we do not lose any generality for doing this.

[2] For a three dimensional vector $u \in \mathbb{R}^3$, we use $\widehat{u} \in \mathbb{R}^{3\times 3}$ to denote the skew symmetric matrix associated to $u$ such that for any vector $v \in \mathbb{R}^3$, we have $\widehat{u}v = u \times v$.

**Comment 3 (Constraint on rotation from $M_l$).** *One may notice that for the matrix $M_l$ to be of rank 1, it is necessary that the first three columns are of rank 1. This imposes constraints on the camera rotation $R_i$'s only. This type of constraints have been utilized in the literature for reconstruction using line segments, e.g. see [25]. $M_l$ however gives a more general presentation for all existing constraints.*

The rank deficiency condition certainly implies all trilinear constraints among the given $m$ images of the line. To see this more explicitly, notice that for $\text{rank}(M_l) \leq 1$, it is necessary for any pair of row vectors of $M_l$ to be linearly dependent. This gives us the well-known trilinear constraints

$$\boxed{\mathbf{l}_j^T T_j \mathbf{l}_i^T R_i \widehat{\mathbf{l}_1} - \mathbf{l}_i^T T_i \mathbf{l}_j^T R_j \widehat{\mathbf{l}_1} = 0} \tag{2.14}$$

among the first, $i^{th}$ and $j^{th}$ images. Hence the constraint $\text{rank}(M_l) \leq 1$ is a natural generalization of the trilinear constraint (for 3 views) to arbitrary $m$ views since when $m = 3$ it is equivalent to the trilinear constraint for lines, except for some rare degenerate cases.

It is easy to see from the rank of matrix $M_l$ that there will be no more independent relationship among either pairwise or quadruple-wise image lines. Trilinear constraints are the only non-trivial ones for all the images that correspond to a single line in 3-D.[3] So far we have essentially given a much more simplified proof for the following facts regarding multilinear constraints among multiple images of a line:

**Theorem 6 (Linear relationships among multiple views of a line).** *For any given $m$ images corresponding to a line $L$ in $\mathbb{E}^3$ relative to $m$ camera frames, the rank deficient matrix $M_l$ implies that any algebraic constraints among the $m$ images can be reduced to only those among 3 images at a time, characterized by the so-called trilinear constraints (2.14).*

Although trilinear constraints are necessary for the rank of matrix $M_l$ (hence $N_l$) to be 1, rigorously speaking they are **not** sufficient. For the equation (2.14) to be non-trivial, it is required that the entry $\mathbf{l}_i^T T_i$ in the involved rows of $M_l$ need to be non-zero. This is not always true for certain degenerate cases - such as the line is parallel to the translational direction. The rank deficiency condition on $M_l$ is a much better way of capturing *all* constraints among multiple images and avoids artificial degeneracy that could be introduced by using algebraic equations. On the other hand, since such degeneracy is rare, in loose terms, we may say these two ways are "equivalent". As we will see soon, the rank deficiency condition is much easier to use for geometric analysis as well as design of practical algorithms.

## 2.4 Geometric interpretation of the rank deficiency condition

In the previous section, we have classified the algebraic constraints that may arise among $m$ corresponding images of a line. We now know that the relationship among $m$ images essentially boils down to those among 3 views at a time, characterized by the trilinear constraints (2.14). But we have not yet explained what these equations mean and whether there is a simpler intuitive geometric interpretation for all these algebraic relationships. We now try to do that rigorously here without using any heavy machinery from algebraic geometry. Our final goal here is to see how all the analysis and results (including those for degenerate cases) can be captured by an extremely simple statement based on the rank of the multiple view matrix $M_l$.

### 2.4.1 Uniqueness of pre-image

Given 3 vectors $\mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3 \in \mathbb{R}^3$, if they are indeed images of some line $L$ in 3-D with respect to the three camera frames as shown in Figure 2.1, they should automatically satisfy the trilinear constraints, e.g.

$$\mathbf{l}_2^T T_2 \mathbf{l}_3^T R_3 \widehat{\mathbf{l}_1} - \mathbf{l}_3^T T_3 \mathbf{l}_2^T R_2 \widehat{\mathbf{l}_1} = 0.$$

Now we ask ourselves the inverse problem: If the three vectors $\mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3$ satisfy the trilinear constraints, are they necessarily images of some single line in 3-D, the so-called *pre-image*? As show in Figure 2.2, we denote

[3] Although we only proved it for the special case with $\Pi_1 = [I, 0]$, the general case differs from this special one only by a choice of a reference frame.

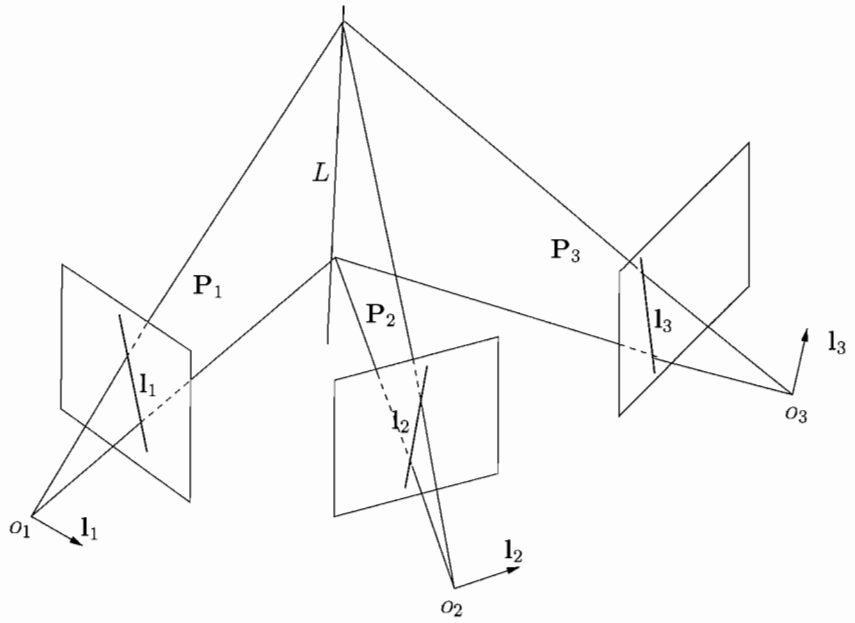


Figure 2.1: Three planes extended from the three images $\mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3$ intersect at one line $L$ in 3-D, the pre-image of $\mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3$.

the planes formed by the optical center $o_i$ of the $i^{th}$ frame and image line $\mathbf{l}_i$ to be plane $\mathbf{P}_i$, $i = 1, 2, 3$. Denote the intersection line between $\mathbf{P}_1$ and $\mathbf{P}_2$ as $L_2$ and the intersection line between $\mathbf{P}_1$ and $\mathbf{P}_3$ as $L_3$. As pointed out at the beginning, $\mathbf{l}_i \in \mathbb{R}^3$ is also the normal vector of $\mathbf{P}_i$. Then without loss of generality, we can assume that $\mathbf{l}_i$ is the unit normal vector of plane $\mathbf{P}_i$, $i = 1, 2, 3$ and the trilinear constraint still holds. Thus, $-\mathbf{l}_i^T T_i = d_i$ is the distance from $o_1$ to the plane $\mathbf{P}_i$ and $(\mathbf{l}_i^T R_i)^T = R_i^T \mathbf{l}_i$ is the unit normal vector of $\mathbf{P}_i$ expressed in the $1^{th}$ frame. Furthermore, $(\mathbf{l}_i^T R_i \widehat{\mathbf{l}_1})^T$ is a vector parallel to $L_i$ with length being $\sin(\theta_i)$, where $\theta_i \in [0, \pi]$ is the angle between the planes $\mathbf{P}_1$ and $\mathbf{P}_i$, $i = 2, 3$. Therefore, in the general case, the trilinear constraint implies two things: First, as $(\mathbf{l}_2^T R_2 \widehat{\mathbf{l}_1})^T$ is linear dependent of $(\mathbf{l}_3^T R_3 \widehat{\mathbf{l}_1})^T$, $L_2$ is parallel to $L_3$. Secondly, as $d_2 \sin(\theta_3) = d_3 \sin(\theta_2)$, the distance from $o_1$ to $L_2$ is the same with the distance from $o_1$ to $L_3$. They then imply that $L_2$ coincides with $L_3$, or in other words, the line $L$ in 3-D space is uniquely determined. However, we have degeneracy when $\mathbf{P}_1$ coincides with $\mathbf{P}_2$ and $\mathbf{P}_3$. In that case, $d_2 = d_3 = 0$ and $(\mathbf{l}_2^T R_2 \widehat{\mathbf{l}_1})^T = (\mathbf{l}_3^T R_3 \widehat{\mathbf{l}_1})^T = 0_{3\times 1}$. There are infinite number of lines in $\mathbf{P}_1 = \mathbf{P}_2 = \mathbf{P}_3$ that generate the same set of images $\mathbf{l}_1$, $\mathbf{l}_2$ and $\mathbf{l}_3$. The case when $\mathbf{P}_1$ coincides with only $\mathbf{P}_2$ or only $\mathbf{P}_3$ is more tricky. For example, if $\mathbf{P}_1$ coincides with $\mathbf{P}_2$ but not with $\mathbf{P}_3$, then $d_2 = 0$ and $(\mathbf{l}_2^T R_2 \widehat{\mathbf{l}_1})^T = 0_{3\times 1}$. However, if we re-index the images (frames), then we still can obtain a non-trivial trilinear constraint, from which $L$ can be deduced as the intersection line between $\mathbf{P}_1$ and $\mathbf{P}_3$. So we have in fact proven the following fact:

**Lemma 3 (Properties of trilinear constraints for lines).** *Given three camera frames with distinct optical centers and any three vectors $\mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3 \in \mathbb{R}^3$ that represents three lines on each image plane. If the three image lines satisfy trilinear constraints*

$$\mathbf{l}_j^T T_{ji} \mathbf{l}_k^T R_{ki} \widehat{\mathbf{l}_i} - \mathbf{l}_k^T T_{ki} \mathbf{l}_j^T R_{ji} \widehat{\mathbf{l}_i} = 0, \quad i, j, k = 1, 2, 3$$

*a unique pre-image $L$ is determined except when the three planes defined respectively by the centers $o_i$'s of the camera and the vectors $\mathbf{l}_i$'s as their normals all coincide with each other. For this only degenerate case, the matrix $M_l$ becomes zero.* [4]

For more than 3 views, in order to check the uniqueness of the pre-image, one needs to apply the above lemma to every triple-wise views. The possible combinations of degenerate cases make it very hard to draw

[4] Here we use subscripts $ji$ to indicate that the related transformation is from the $i^{th}$ frame to the $j^{th}$.

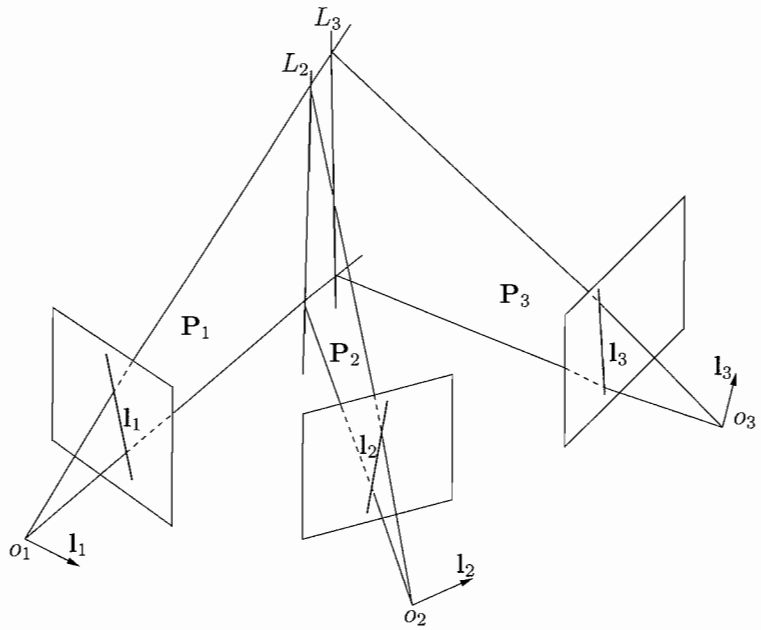


Figure 2.2: Three planes extended from the three images $\mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3$ intersect at lines $L_2$ and $L_3$, which actually coincides.

any consistent conclusion. However, in terms of the rank condition on the multiple view matrix, the lemma can be generalized to multiple views in a much more concise and unified form:

**Theorem 7 (Uniqueness of pre-image).** *Given $m$ vectors in $\mathbb{R}^3$ representing lines on the image planes with respect to $m$ camera frames, they correspond to the same line in the 3-D space if the maximum rank of the $M_l$ matrix relative to all the camera frames is of rank 1. If its rank is always 0 (i.e., the matrix $M_l$ is zero), then the line is determined up to a plane on which all the camera centers must lie.*

The proof follows directly from Theorem 5. So the case that the line in 3-D shares the same plane as all the centers of the camera frames is the only degenerate case when one will not able to determine the exact 3-D location of the line from its multiple images. As long as the camera frames have distinct centers, the set of lines that are coplanar with these centers is of only zero measure. Hence, roughly speaking, trilinear constraints among images of a line rarely fail in practice. Even the rectilinear motion will not pose a problem as long as enough number of lines in 3-D are observed, the same as in the point case. On the other hand, the theorem also suggests a criteria to tell from the matrix $M_l$ when a degenerate configuration is present: exactly when the largest singular value of all $M_l$ matrices (with respect to all camera frames) is close to zero.

### 2.4.2 Geometry of the multiple view matrix

Now we can have better understanding of the geometric meaning of the matrix $M_l$. If $\mathbf{l}_1, \ldots, \mathbf{l}_m$ are the $m$ images of one line $L$ in $m$ different views, and without loss of generality, $\mathbf{l}_i$'s are unit vectors, then $R_i^T \mathbf{l}_i$ is the normal vector of the plane $\mathbf{P}_i$ formed by $L$ and the optical center $o_i$ of the $i^{th}$ frame expressed in the $1^{th}$ frame. With $-\mathbf{l}_i^T T_i$ being the distance from $o_1$ to $\mathbf{P}_i$, $[\mathbf{l}_i{}^T R_i \quad \mathbf{l}_i^T T_i] X = 0$, $X = [x, y, z, 1] \in \mathbb{R}^4$ is the function of plane $\mathbf{P}_i$. Besides, since $\mathbf{l}_1, R_2^T \mathbf{l}_2, \ldots, R_m^T \mathbf{l}_m$ are all perpendicular to $L$, they are coplanar. Hence $(\mathbf{l}_i^T R_i \widehat{\mathbf{l}}_1)^T$ is parallel to the vector along the line $L$, *i.e.*, $[\mathbf{l}_i^T R_i \widehat{\mathbf{l}}_1, 0]^T \in \mathbb{R}^4$ is proportional to the vector $v$ which defines the line $L$ in 3-D. Since $M_l$ has rank 1, if we view each row of $M_l$ as homogeneous coordinates of some point in 3-D, then all the rows in fact defines a unique point in 3-D. This point is not necessarily on the image plane though. If we call this point $p$, then the vector defined by $p - o_1$ is obviously parallel to the original line $L$ in 3-D. We also call it $v$. Therefore the so-defined $M_l$ matrix in fact gives us a map from

lines in 3-D to vectors in 3-D:

$$M_l : L \subset \mathbb{R}^3 \mapsto v \in \mathbb{R}^3.$$

This map is certainly not injective but surjective. That is, from $M_l$ matrix alone, one will not be able to recover the exact 3-D location of the line $L$, but it will give us most of the information that we need to know about its images. Moreover, the vector gives the direction of the line, and the norm $\|v\|$ is exactly the ratio:

$$\|v\| = \sin(\theta_i)/d_i, \quad \forall i = 2, \ldots, m.$$

Roughly speaking, the farther away is the line $L$ from $o_1$, the smaller this ratio is. In fact, one can show that the family of parallel lines in 3-D that all map to the same vector $v$ form a cylinder centered around $o_1$. The above ratio is exactly the inverse of the radius of such a cylinder. We may summarize our discussion into the following theorem:

**Theorem 8 (Geometry of the matrix $M_l$).** *The matrix $M_l$ associated to a line $L$ in 3-D maps this line to a unique vector $v$ in 3-D. This map is surjective and two lines are mapped to the same vector if and only if they are: 1. parallel to each other and 2. of the same distance to the center $o$ of the reference camera frame. That distance is exactly $1/\|v\|$.*

Therefore, knowing $M_l$ (*i.e.*, the vector $v$), we know that $L$ must lie on a circle of radius $r = 1/\|v\|$, as shown in Figure 2.3. So if we can obtain the $M_l$ matrix for $L$ with respect to another camera center, we get

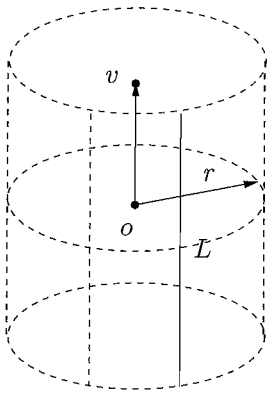


Figure 2.3: An equivalent family of parallel lines that give the same $M_l$ matrix.

two families of parallel lines lying on two cylinders. In general, these two cylinders intersect at two lines, unless the camera centers all lie on a line parallel to the line $L$. Hence, two $M_l$ matrices (for the same line in 3-D) with respect to two distinct camera centers determine the line $L$ up to two solutions. One would imagine that, in general, a third $M_l$ matrix respect to a third camera center will then uniquely determine the 3-D location of the line $L$.

### 2.4.3 Relationships between rank deficiency conditions for line and point

Recently, a similar rank deficiency condition for point has also been worked out. Let $\mathbf{x}_1, \ldots, \mathbf{x}_m$ be the $m$ images of a point $p$ in 3-D space, and $(R_i, T_i) \in \mathbb{R}^{3\times 4}$ be the corresponding transformation from the $i^{th}$ camera frame to the first, $i = 2, \ldots, m$. Denote

$$M_p = \begin{bmatrix} \widehat{\mathbf{x}_2} R_2 \mathbf{x}_1 & \widehat{\mathbf{x}_2} T_2 \\ \vdots & \vdots \\ \widehat{\mathbf{x}_m} R_m \mathbf{x}_1 & \widehat{\mathbf{x}_m} T_m \end{bmatrix} \in \mathbb{R}^{3(m-1)\times 2}.$$

Then according to the rank deficiency condition for point in Chapter 1, we have

$$\text{rank}(M_p) \leq 1.$$

The apparent similarities between both the rank deficiency conditions and the forms of $M_l$ and $M_p$ are expected due to the geometric duality between lines and point. In the 3-D space, a point can be uniquely determined by two lines, while a line can be uniquely determined by two points. So our question now is that if the two set of rank deficiency conditions can be derived from each other based on the geometric duality.

First we show that we can derive the rank deficiency condition for line from rank deficiency condition for point. Let $p_1$ and $p_2$ be two distinct points on a line $L$ in the 3-D space. Denote the $m$ images of $p_1$, $p_2$ under $m$ views to be $\mathbf{x}_1^1, \ldots, \mathbf{x}_m^1$ and $\mathbf{x}_1^2, \ldots, \mathbf{x}_m^2$, respectively. Hence, the $i^{th}$ $(i = 1, \ldots, m)$ view of $L$ can be expressed as $\mathbf{l}_i = \widehat{\mathbf{x}_i^2}\mathbf{x}_i^1$ or $\widehat{\mathbf{l}_i} = \mathbf{x}_i^2\mathbf{x}_i^{1T} - \mathbf{x}_i^1\mathbf{x}_i^{2T}$, $i = 1, \ldots, m$. From the rank deficiency of $M_p^1$ and $M_p^2$, we have $\widehat{\mathbf{x}_i^1}R_i\mathbf{x}_1^1 = \alpha\widehat{\mathbf{x}_i^1}T_i$ and $\widehat{\mathbf{x}_i^2}R_i\mathbf{x}_1^2 = \beta\widehat{\mathbf{x}_i^2}T_i$ for some $\alpha, \beta \in \mathbb{R}$ and $i = 1, \ldots, m$. This gives

$$\mathbf{l}_i^T R_i \widehat{\mathbf{l}_1} = -\mathbf{x}_i^{1T}\widehat{\mathbf{x}_i^2}T_i(\alpha\mathbf{x}_1^{2T} - \beta\mathbf{x}_1^{1T}), \quad \mathbf{l}_i^T T_i = -\mathbf{x}_i^{1T}\widehat{\mathbf{x}_i^2}T_i. \tag{2.15}$$

which means that each row of $M_l$ is spanned by the same vector $[(\alpha\mathbf{x}_1^{2T} - \beta\mathbf{x}_1^{1T}), 1]^T \in \mathbb{R}^4$. Therefore,

$$\text{rank}(M_p^1) \leq 1 \;\&\; \text{rank}(M_p^2) \leq 1 \quad \Rightarrow \quad \text{rank}(M_l) \leq 1.$$

The inverse direction of this duality is not so straightforward. The reason is obvious: The duality is not totally symmetric. In the 3-D space, any distinct two points can determine a line. However, not any two lines may intersect at one point unless they are coplanar. Hence, in order to prove the inverse, an additional coplanar condition for the two lines in space should be imposed. This will be investigated in the next chapter. But the matrices $M_p$ and $M_l$ already reveal some interesting *duality* between the camera center $o$ and the point $p$ in 3-D. For example, if the camera center moves on a straight line (the rectilinear motion), from the $M_p$ matrix associated to a point $p$, the 3-D location of the point $p$ can only be determined up to a circle. On the other hand, if the camera center is fixed but the point $p$ can move on a straight line $L$, from the $M_l$ matrix associated to the line $L$, the 3-D location of this line can only be determined up to a circle too. Mathematically speaking, matrices $M_p$ and $M_l$ define an equivalence relationship for points and lines in the 3-D space, respectively. They both group points and lines according to their distance to the center of the reference camera frame. Numerically, the sensitivity for $M_p$ and $M_l$ as rank deficient matrices depends very much on such distance. Roughly speaking, the farther away is the point or line, the more sensitive the matrix is to noise or disturbance. Hence, in practice, one may view $M_p$ and $M_l$ as a natural "metric" for the quality of the measurements associated to the multiple images of a 3-D point or line.

## 2.5 Applications of the rank deficiency condition

The rank deficiency condition on the matrix $M_l$ allows us to use all the constraints among multiple images of a line feature simultaneously without specifying a set of triple-wise frames. That is, it makes it possible to use all the data simultaneously for purposes such as feature matching or recovering camera configuration from multiple images of lines. Many natural algorithms are suggested by the simple form of the $M_l$ matrix. These algorithms run in exact parallel as those outlined for the case of point features.

Ideally, one likes to formulate the entire problem of reconstructing 3-D motion and structure as one optimizing some global objective function[5] subject to the rank deficiency condition. However, such an approach usually relies on very difficult nonlinear optimization methods. Instead, we here divide the overall problem into a few subproblems: matching features assuming known motion, and estimating motion assuming known matched features. Sections 4.1 and 4.2 treat these two subproblems respectively.

### 2.5.1 Multiple view matching test for line features

In general we like to test if given $m$ vectors $\mathbf{l}_1, \ldots, \mathbf{l}_m \in \mathbb{R}^3$ with known $\Pi$ indeed satisfy all the constraints that $m$ images of a single 3-D line (pre-image) should. There are two ways of performing this. One is based on the fact that $M_l$ matrix has rank no more than 1. Another is based on the geometric interpretation of $M_l$ matrix.

[5]Such as the *reprojection error* in image.

Since $\text{rank}(M_l) \leq 1$, the $4 \times 4$ matrix $M_l^T M_l$ also has rank no more than 1. Hence, ideally with given $\mathbf{l}_1, \ldots, \mathbf{l}_m$ and $\Pi$, the eigenvalues of $M_l^T M_l$ should satisfy $\lambda_1 \geq 0$, $\lambda_2 = \lambda_3 = \lambda_4 = 0$. A more numerically robust algorithm would be:

**Algorithm 3 (Multiple view matching test).** *Suppose the projection matrix $\Pi$ associated to $m$ camera frames are given. Then for given $m$ vectors $\mathbf{l}_1, \ldots, \mathbf{l}_m \in \mathbb{R}^3$,*

1. *Compute the matrix $M_l \in \mathbb{R}^{m \times 4}$ according to (2.12).*
2. *Compute second largest eigenvalue $\lambda_2$ of the $4 \times 4$ matrix $M_l^T M_l$;*
3. *If $\lambda_2 \leq \epsilon_1$ for some pre-fixed threshold $\epsilon_1$, then we say the $m$ image vectors match. Otherwise discard it as outliers.*

### 2.5.2 Multiple view motion estimation from line features

Now suppose that $m(\geq 3)$ images $\mathbf{l}_1^j, \ldots, \mathbf{l}_m^j$ of $n$ lines $L^j$, $j = 1, \ldots, n$ are given and we want to use them to estimate the unknown projection matrix $\Pi$. From the rank deficiency condition of the $M_l$ matrix, we know that the kernel of $M_l$ should have 3 dimensions. Denote $M_l^j$ to be the $M_l$ matrix for the $j^{th}$ line, $j = 1, \ldots, n$. Since ${\mathbf{l}_i^j}^T R_i \widehat{\mathbf{l}_1^j}$ is a vector parallel to $L^j$, then for any two linear independent vectors $u^j, w^j \in \mathbb{R}^3$ lying in the plane perpendicular to $L^j$, $\begin{bmatrix} u^j \\ 0 \end{bmatrix}$ and $\begin{bmatrix} w^j \\ 0 \end{bmatrix}$ are two base vectors in the kernel of $M_l^j$. Let $\begin{bmatrix} \alpha^j \\ 1 \end{bmatrix} \in \ker(M_l^j)$ be a vector orthogonal to both $\begin{bmatrix} u^j \\ 0 \end{bmatrix}$ and $\begin{bmatrix} w^j \\ 0 \end{bmatrix}$. Then

$$\alpha^j = [\alpha_1^j, \alpha_2^j, \alpha_3^j]^T = k^j \widehat{u^j} w^j, \tag{2.16}$$

for some $k^j \in \mathbb{R}$. It is direct to see that $\alpha^j$ is a vector parallel to the line $L^j$. It is in fact pointing to the opposite direction of the vector $v^j$ associated to the line $L^j$ and has a norm equal to the distance from the line $L^j$ to the center of the reference camera frame. Thus for the $i^{th}$ view, $i = 2, \ldots, m$, we define matrices:

$$P_i := \begin{bmatrix} {\mathbf{l}_i^1}^T & {\alpha^1}^T(-\widehat{\mathbf{l}_1^1} * {\mathbf{l}_i^1}^T) \\ \vdots & \vdots \\ {\mathbf{l}_i^n}^T & {\alpha^n}^T(-\widehat{\mathbf{l}_1^n} * {\mathbf{l}_i^n}^T) \\ 0 & {u^1}^T(-\widehat{\mathbf{l}_1^1} * {\mathbf{l}_i^1}^T) \\ \vdots & \vdots \\ 0 & {u^n}^T(-\widehat{\mathbf{l}_1^n} * {\mathbf{l}_i^n}^T) \\ 0 & {w^1}^T(-\widehat{\mathbf{l}_1^1} * {\mathbf{l}_i^1}^T) \\ \vdots & \vdots \\ 0 & {w^n}^T(-\widehat{\mathbf{l}_1^n} * {\mathbf{l}_i^n}^T) \end{bmatrix} \in \mathbb{R}^{3n \times 12}, \tag{2.17}$$

where $*$ is the *Kronecker* product. Then we have

$$P_i \begin{bmatrix} \vec{T}_i \\ \vec{R}_i \end{bmatrix} = 0, \tag{2.18}$$

where $\vec{T}_i = T_i$ and $\vec{R}_i = [r_{11}, r_{21}, r_{31}, r_{12}, r_{22}, r_{32}, r_{13}, r_{23}, r_{33}]^T$, with $r_{kl}$ being the $(kl)^{th}$ entry of $R_i$, $k, l = 1, 2, 3$. It can be shown that if $\alpha^j, u^j, w^j$'s are known, the matrix $P_i$ is of rank 11 if more than $n \geq 12$ lines in general position are given. In that case, the kernel of $P_i$ is unique, and so is $(R_i, T_i)$. Hence if we know $\alpha^j$ for each $j = 1, \ldots, n$, then we can estimate $(R_i, T_i)$ by performing singular value decomposition (SVD) on $P_i$. In practice, the initial estimations of $\alpha^j, u^j, w^j$'s may be noisy and only depend on local data, so we can use iteration method. First, we get estimates for $(R_i, T_i)$'s, then we use the estimated motions to re-calculate $\alpha^j$'s, and iterate in this way till the algorithm converges. However, currently there are several ways to update $\alpha^j, u^j, w^j$'s in each iteration. Initialization of $\alpha^j, u^j, w^j$ and the overall algorithm are described in the next section.

**Euclidean reconstruction**

Here we illustrate the overall algorithm for the case when the camera is assumed to be calibrated. That is $A(t) = I$ hence the projection matrix $\Pi_i = [R_i, T_i]$ represents actual Euclidean transformation between camera frames. We will discuss later on what happens if this assumption is violated.

Given the first three views of (at least) 12 lines in the 3-D space. Let $M_{l3}^j$ be the matrix formed by the first 3 columns of $M_l^j$ associated to the first three views. From $M_{l3}^j$, we can already estimate $u^j, w^j \in \ker(M_{l3}^j)$, where $u^j, w^j$ are defined above. So after we obtain an initial estimation of $(\tilde{R}_i, \tilde{T}_i)$ for $i = 2, 3$, we can calculate $\tilde{M}_{l3}^j$ and compute the SVD of it such that $\tilde{M}_{l3}^j = U_2 S_2 V_2^T$, then pick $\tilde{u}^j, \tilde{w}^j$ being the $2^{nd}$ and $3^{rd}$ columns of $V_2$ respectively. Then the estimation for $\alpha^j$ is $\tilde{\alpha}^j = k^j(\tilde{u}^j \times \tilde{w}^j)$ for some $k^j \in \mathbb{R}$ to be determined. $k$ can be estimated by least squares method such that

$$k^j = -\frac{\sum_{i=2}^m (\mathbf{l}_i^{j\,T} \tilde{T}_i)(\mathbf{l}_i^{j\,T} \tilde{R}_i \widehat{\mathbf{l}_1^j} \widehat{\tilde{u}^j} \tilde{w}^j)}{\sum_{i=2}^m (\mathbf{l}_i^{j\,T} \tilde{R}_i \widehat{\mathbf{l}_1^j} \widehat{\tilde{u}^j} \tilde{w}^j)^2}. \tag{2.19}$$

Using the estimated $\alpha^j, u^j, w^j$, we can now compute the matrix $P_i$ from (2.17). By performing SVD on $P_i$ such that $P_i = USV^T$, we then pick the last column of $V$ to obtain estimates respectively for $T_i$ and $R_i$ as $\bar{T}_i \in \mathbb{R}^3$ and $\bar{R}_i \in \mathbb{R}^{3\times 3}$ for $i = 2, \ldots, m$. Since $\bar{R}_i$ is not guaranteed to be a rotation matrix, in order to project $\bar{R}_i$ onto $SO(3)$, we can then do SVD on $\bar{R}_i$ such that $\bar{R}_i = U_i S_i V_i^T$, so the estimates for $R_i \in SO(3)$ and $T_i \in \mathbb{R}^3$ are

$$\tilde{T}_i = \frac{\text{sign}(\det(U_i V_i^T))}{\sqrt[3]{\det(S_i)}} \bar{T}_i \in \mathbb{R}^3, \tag{2.20}$$

$$\tilde{R}_i = \text{sign}(\det(U_i V_i^T)) U_i V_i^T \in SO(3). \tag{2.21}$$

The $\alpha^j, u^j, w^j$ can then be re-estimated from the full matrix $M_l^j$ computed from the motion estimates $(\tilde{R}_i, \tilde{T}_i)$.

Based on above arguments, we can summarize our algorithm as the following:

**Algorithm 4 (SFM from multiple views of lines).** *Given a set of $m$ images $\mathbf{l}_1^j, \ldots, \mathbf{l}_m^j$ of $n$ lines $L^j$, $j = 1, \ldots, n$, we can estimate the motions $(R_i, T_i)$, $i = 2, \ldots, m$ as the following:*

1. *Initialization*
   (a) *Set step counter $s = 0$.*
   (b) *Compute initial estimates for $(R_i, T_i)$, $i = 2, 3$ for the first three views.*
   (c) *Compute initial estimates of $\alpha_s^j, u_s^j, w_s^j$ for $j = 1, \ldots, n$ from the first three views.*
2. *Compute $P_i$ from (2.17) for $i = 2, \ldots, m$, and obtain $(\bar{R}_i, \bar{T}_i)$ from the eigenvector associated to its smallest singular value.*
3. *Compute $(\tilde{R}_i, \tilde{T}_i)$ from (2.20) and (2.21), $i = 2, \ldots, m$.*
4. *Compute $\alpha_{s+1}^j, u_{s+1}^j, w_{s+1}^j$ based on the $M^j$ matrix calculated by using $(\tilde{R}_i, \tilde{T}_i)$. Normalize $\alpha_{s+1}^j$ so that $\|\alpha_{s+1}^1\| = 1$.*
5. *If $\|\alpha_{s+1} - \alpha_s\| < \epsilon$ for some threshold $\epsilon$, then stop, else set $s = s + 1$, and goto 2.*

There are several notes for the above algorithm:

1. From (2.17) we can see that in order to get a unique $(R_i, T_i)$ from SVD on $P_i$, we want the rank of $P_i$ to be 11, this requires that we have at least 12 pairs of line correspondences.
2. There are several ways for the initial estimation of $\alpha^j$'s. There is an nonlinear algorithm for estimating trifocal tensor given by Hartley *et. al.* [8]. Although linear algorithms for three views of line features also exist [33], they usually require at least 13 lines matched across three views. In practice, one may instead use the two view 8 point algorithm to initialize the first three views.

3. The way that $\alpha^j$ is re-estimated from $M_l^j$ is not unique. It can also be recovered from the rows of $M_l^j$, from the relationship between $\alpha^j$ and $v$ mentioned above. The reason to set $\|\alpha_{s+1}^1\|$ to be 1 in Step 4 is to fix the universal scale. It is equivalent to putting the first line at a distance of 1 to the first camera center.

4. There is an interesting characteristic of the above algorithm: $(R_i, T_i)$ seem to be estimated using pairwise views only. But it is not exactly true. The computation of the matrix $P_i$ depends on all $\alpha^j, u^j, w^j$ each of which is estimated from the $M_l^j$ matrix for all views.

**Remark 4 (Euclidean, affine or projective reconstruction).** *We must point out that, although it seems to be proposed for the calibrated camera (Euclidean) case only, the algorithm works just the same if the camera is weakly calibrated (affine) or uncalibrated (projective). The only change needed is at the initialization step. In order to initialize $\alpha^j$'s, either a Euclidean, affine or projective choice for $(R_2, T_2)$ and $(R_3, T_3)$ needs to be specified. This is where our knowledge of the camera calibration enters the picture. In the worst case, i.e., when we have no knowledge on the camera at all, any choice of a projective frame will do. In that case, the relative transformation among camera frames and the scene structure are only recovered up to a projective transformation of choice. Once that is done, the rest of the algorithm runs exactly the same.*

## 2.6 Simulations on synthetic data

In this section, we show in simulation the performance of the above algorithm. We test it on two different scenarios: sensitivity of motion and structure estimates with respect to the level of noise on the line measurements; and the effect of number of frames on motion and structure estimates.

### 2.6.1 Setup

The simulation parameters are as follows: number of trials is 500, number of feature lines is typically 20, number of frames is 4 (unless we vary it on purpose), field of view is $90^o$, depth variation is from 100 to 400 units of focal length and image size is $500 \times 500$. Camera motions are specified by their translation and rotation axes. For example, between a pair of frames, the symbol $XY$ means that the translation is along the $X$-axis and rotation is along the $Y$-axis. If $n$ such symbols are connected by hyphens, it specifies a sequence of consecutive motions. The ratio of the magnitude of translation $\|T\|$ and rotation $\theta$, or simply the $T/R$ *ratio*, is compared at the center of the random cloud scattered in the truncated pyramid specified by the given field of view and depth variation. For each simulation, the amount of rotation between frames is given by a proper angle and the amount of translation is then automatically given by the $T/R$ ratio. We always choose the amount of total motion such that all feature (lines) will stay in the field of view for all frames. In all simulations, each image line is perturbed by independent noise with a standard deviation given in degrees. Error measure for rotation is $\arccos\left(\frac{tr(R\tilde{R}^T)-1}{2}\right)$ in degrees where $\tilde{R}$ is an estimate of the true $R$. Error measure for translation is the angle between $T$ and $\tilde{T}$ in degrees where $\tilde{T}$ is an estimate of the true $T$. Error measure for structure is approximately measured by the angle between $\alpha$ and $\tilde{\alpha}$ where $\tilde{\alpha}$ is an estimate of the true $\alpha$.

Algorithm 2 requires that we initialize with the first three views. In simulations below, we initialize the motion among the first three views using motion estimates from the linear algorithm for point features (see the previous chapter). The error in initial motion estimates correspond to an increasing level of noise on image points from 0 to 5 pixels. While the line measurements are perturbed by an increasing level of random angle from 0 to 2.5 degrees, the motion for the first three views are initialized with a corresponding level of noisy estimates. Of course, one may run existing algorithms based on trilinear constraint [33] for line features and initialize the first three views. But motion estimation from line measurements alone is much more sensitive to degenerate configurations - any line coplanar with the translation gives essential no information on the camera motion nor its own 3-D structure. On the other hand, point features give more stable initial estimates.

### 2.6.2 Motion and structure from four frames

Figures 2.4 and 2.5 plot respectively the motion and structure estimate errors versus the level of noises added on the line measurements. The motion sequence is $XX$-$YY$-$X(XY)$, where $(XY)$ means a direction half between the $X$-axis and $Y$-axis. Unlike point features, the quality of line feature measurements depends heavily on the camera motion. Among the 20 randomly generated 3-D lines, those which are coplanar with the translation will give little information on the camera motion or its 3-D location. These "bad" measurements typically contribute to a larger error in estimates. Their effect is quite noticeable in simulations and sometimes even causes numerical instability. By examining the multiple view matrix $M_l$ associated to each line, we believe in the future we will be able to find good criteria to eliminate the "bad" lines hence improve the motion and structure estimates. Even so, the current algorithm is still able to converge to reasonably good estimates for the uninitialized motion between the fourth frame and the first.

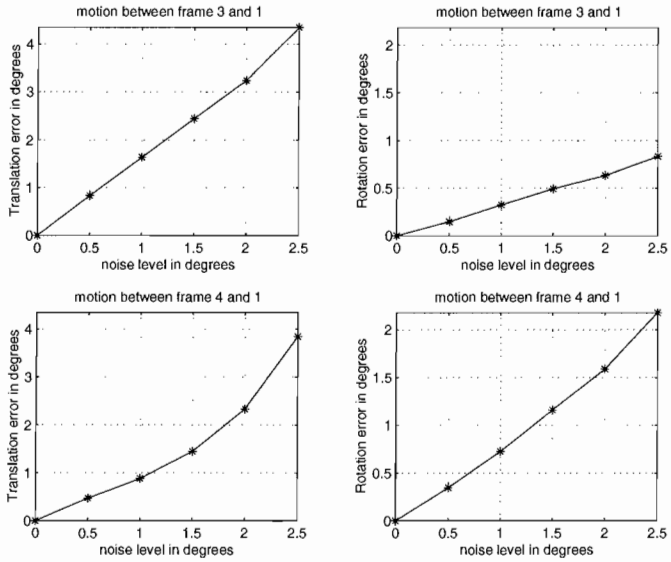


Figure 2.4: Motion estimate error from four views. The number of trials is 500. $T/R$ ratio is 1. The amount of rotation between frames is $20^o$.

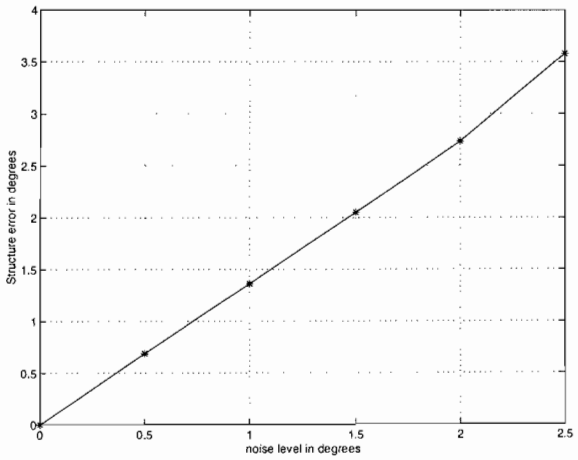


Figure 2.5: Structure estimate error from four views. The number of trials is 500. $T/R$ ratio is 1. The amount of rotation between frames is $20^o$.

### 2.6.3 Error as a function of number of frames

Figures 2.6 and 2.7 plot respectively the motion and structure estimate errors versus the number of frames. The noise level on the image line measurements is $1^o$. The motion sequence is an orbital motion around the set of 20 random lines generated at a distance between 200 and 400 units of focal length away from the camera centers. The amount of rotation is incrementally $15^o$ between frames and a corresponding amount of translation is used to generate the orbital motion. From Figure 2.7, we see that the structure estimates improve while the number of frames increases, which is expected. In fact, the error converges to the given level of noise on the line measurements. However, according to Figure 2.6, motion estimates do not necessarily improve with an increase number of frames since the number of motion parameters increase linearly with the number of frames. In fact, we see that after a few frames, additional frames will have little effect on the previous motion estimates.

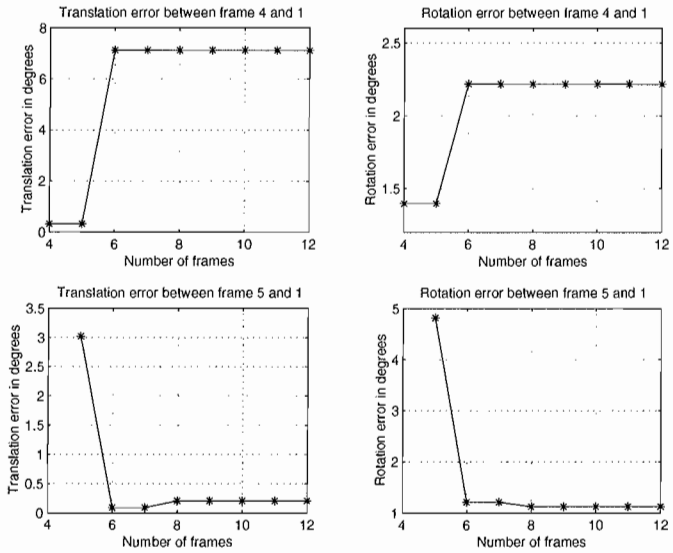


Figure 2.6: Motion (between frames 4-1 and 5-1) estimate error versus number of frames for an orbital motion. The number of trials is 500.

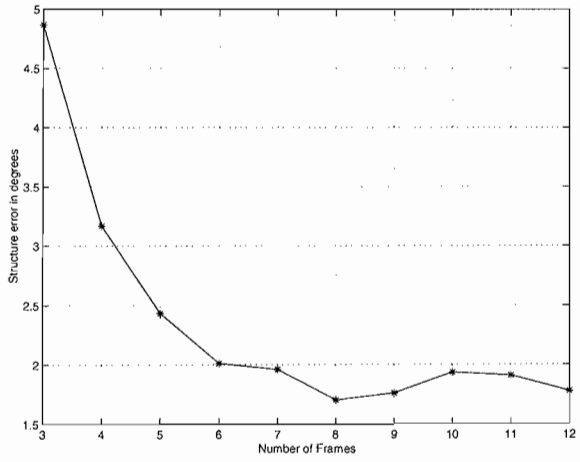


Figure 2.7: Structure estimate error versus number of frames for an orbital motion. The number of trials is 500.

**Comment 4.** *From our experience with the simulations, we want to mention a few things about the proposed linear algorithm for line features:*

- *Degenerate configuration is a more severe problem for line features than point features. Eliminating bad lines based on the multiple view matrix $M_l$ will improve the estimate significantly.*
- *The rank 1 condition for the four column $M_l$ is numerically less stable than that for the two column $M_p$ for point features. Better numerical techniques for imposing the rank condition on $M_l$ are worth investigating.*
- *The algorithm requires at least 12 lines which is twice as many as that required by the algorithm using point features. Increase the number of line features, motion estimates will improve but not the structure.*
- *In the future, we need to further investigate how these algorithms perform if a mixture of point and line features are used, or additional constraints such as coplanar are imposed on the features.*

## 2.7 Discussions and conclusions

The main contribution of this chapter is the derivation and characterization of a new matrix $M_l$ which captures the information about multiple views of line features. It is shown that the rank deficiency condition of $M_l$ implies all the trilinear constraints among $m$-views on a line feature. This condition can be stated in terms of simple linear dependence relationship between any two rows of matrix $M_l$ and provides the trilinear constraints in a compact form, without introducing the tensorial notation. The structure of $M_l$ reveals the duality between line and point features and also gives rise to a new set of linear algorithms for line matching across multiple views, or motion and structure estimation from line features. The development and all the results were established using linear algebraic techniques and all the involved entities have a clear geometric characterization.

Although in this chapter and the previous one we have only established the rank deficiency condition for multiple images of a point or a line in the 3-D space, the idea can be easily extended to more general 3-D objects such as polygons, curves or even surfaces. For example, in the case of a polygon, the duality between the rank deficiency conditions for its vertices and its edges can be easily established since now all features are automatically coplanar. The rank deficiency condition then automatically implies the homography among all images (see the next chapter). In the case of a curve, the rank deficiency conditions for corresponding points on the curve and corresponding tangent vectors to the curve determine each other. As a result, we believe that the rank deficiency condition indeed provides a universal tool for studying multiple view geometry.

In summary, we believe that utilizing geometric constraints among multiple images is the key to solve the 2-D to 3-D reconstruction problem. The rank deficiency condition is simply a very concise, insightful and unifying way to capture all the constraints. Due to its simplicity, it is also easier to be analyzed and used for designing linear, optimal and even robust algorithms for 3-D reconstruction. As the reader may have noticed that from the motion estimation algorithms for point and line features, the rank deficiency condition has united previous results on two and three views into the multiple view setting, through the initialization step of the proposed algorithms. It is also the initialization step where a choice of a Euclidean, affine or projective basis determines the nature of the solution. We believe future study will be able to reveal more detailed relationship of the proposed approach with the existing stratification scheme.

# Chapter 3

# The Multiple View Matrix for Planar Features

*Yi Ma, Kun Huang, René Vidal*
*Submitted to CVPR01, May 18th.*

## Abstract

*In this chapter, a new rank deficiency condition is presented for multiple images of a point or line feature on a 3-D plane. It is shown that a set of image points or lines correspond to a unique point or line on the plane if and only if the associated multiple view matrix $M_p$ or $M_l$ is of rank 1. This condition integrates all multilinear constraints as well as all homographic constraints for a planar scene. The rank deficient matrices $M_p$ and $M_l$ exhibit clear geometric meaning and the duality between point and line is naturally revealed. The rank condition gives rise to a linear algorithm for structure and motion estimation from images of multiple coplanar points (or lines). This algorithm is a specialization of similar algorithms known for generic 3-D points and lines. In particular, it utilizes multilinear constraints and homographic constraints simultaneously, as opposed to previous methods which mostly are based on homography only. It also systematically extends existing linear algorithms for planar scenes from two views to multiple views. Simulations are presented to evaluate the proposed algorithm.*

**Key words:** Multiple view matrix, rank deficiency condition, multilinear constraints, homography, planar scene, point-line duality, structure from motion.

## 3.1 Introduction

It is well-known that geometric constraints governing multiple images of a generic 3-D point or a line are given by the so-called *multilinear constraints*. In computer vision, these constraints have been the basis of numerous algorithms for motion and structure recovery. However, algorithms developed for a generic scene [16, 11] based on these constraints are known to be ill-conditioned when the set of 3-D feature points or lines happen to form certain degenerate configurations. A case of the most practical importance is when all the 3-D points or lines belong to a single 3-D plane. In this case a special algorithm based on the so-called *homography* has to be used instead [5]. The homography based approach works simply because it explicitly takes into account the knowledge that the points and lines are coplanar. However, such an approach usually ignores the multilinear constraints among images. In the light of a more unified characterization of constraints among multiple images through the so-called *rank deficiency condition* derived in the previous chapters, this chapter studies whether it can be generalized so as to give a unified characterization of all the constraints that are present - multilinear or homographic - for coplanar point and line features. In particular, the result should provide a clear theoretical explanation why algorithms developed for generic scenes do not apply to planar ones.

The study of planar features (point or line) is of both practical and theoretical importance. In many real-life applications, features observed by camera are mostly from the same 3-D plane. For example, in the case of landing of a helicopter or aircraft, an on-board camera system is usually designed to track the landing pad and estimate its relative orientation from certain pattern of features (points or lines) on the pad. Similar situations can be found in autonomous assembly lines, where objects mainly lie on a plane and vision systems are used to locate their coordinates. Planar features are also proven to be useful in reconstruction of architectural objects for many man-made structures are multi-faceted. For such objects, a full reconstruction very often can be decomposed to reconstruction of one facet at a time using the edge or corner features. In all these applications, a clear understanding of multiple view geometry of planar features is extremely important in order to properly modify the theory for a generic scene to a planar scene, as well as extend algorithms from two views to multiple views.

Theoretical significance of the planar case mainly lies within the *duality* between point and line features. As we know, two points in 3-D uniquely determine a line. Hence, the rank deficiency condition for the two points implies that for the line determined by them (see the preceding chapter). The opposite direction is not necessarily true simply because two lines in 3-D typically do not intersect at a point. This however can be resolved when a coplanar condition is imposed on all the features: two coplanar lines do intersect at a unique point. As we will show, the rank deficiency conditions for point and line indeed imply each other in the planar case. This duality essentially allows us to treat point and line as the same in the planar case. Given multiple images of a set of coplanar points, we can either use them directly or use the set of lines determined by every pair of points for reconstruction. The results would be exactly the same.

**Relation to previous work:** The 2-D homography between two images of a planar scene has been extensively studied in the computer vision literature. The affine transformation induced from the homography between the images has been used for many purposes, such as structure and motion recovery [24, 31, 13, 17, 5, 33], dense matching (of planar patches), and even camera self-calibration [29]. A comprehensive account of the computational issues and applications associated to homography can be found in [8]. While existing results are mostly developed in a two-view setting, in this chapter, we try to unify these results and extend them to multiple views. The fundamental tool is a rank deficiency condition on the so-called multiple view matrix proposed in the first two chapters and we here show how to generalize such a condition to the planar case. In particular, the theory will be developed in exact parallel for both point and line features and in the end their equivalence is summarized in a duality between the two rank deficient conditions for both features.

**How this chapter is organized:** Section 3.2 derives rank deficiency condition of the multiple view matrix for a point on a plane and reveals its relationship with multilinear constraints and homography; Section 3.3 gives a similar rank deficiency condition of the multiple view matrix for a line on a plane. Section 3.4 establishes a complete duality between point and line features using the multiple view matrix. Section 3.5 shows how to use the rank condition on the multiple view matrix for motion and structure reconstruction from a planar scene. Proposed algorithms use image data from all views and constraints (multilinear or homography) simultaneously. Simulation results for the proposed algorithms are given in Section 3.6.

## 3.2 The multiple view matrix for a point on a plane

An image $\mathbf{x}(t) = [x(t), y(t), z(t)]^T \in \mathbb{R}^3$ (in homogeneous coordinates) of a point $p \in \mathbb{E}^3$, with homogeneous coordinates $\mathbf{X} = [X, Y, Z, 1]^T \in \mathbb{R}^4$ relative to a fixed world coordinate frame, taken by a moving camera satisfies the following relationship

$$\lambda(t)\mathbf{x}(t) = A(t)Pg(t)\mathbf{X} \tag{3.1}$$

where $\lambda(t) \in \mathbb{R}_+$ is the (unknown) depth of the point $p$ relative to the camera frame, $A(t) \in SL(3)$ is the camera calibration matrix (at time $t$), $P = [I, 0] \in \mathbb{R}^{3\times 4}$ is the constant projection matrix and $g(t) \in SE(3)$ is the coordinate transformation from the world frame to the camera frame at time $t$. In the above equation, all $\mathbf{x}, \mathbf{X}$ and $g$ are in homogeneous representation. The reader may have noticed that here we allow the calibration matrix $A$ to change in time. This is partly because such a generalization does not make it any more difficult for the development of the results in this chapter. On the other hand, it encompasses a richer

class of practical situations when some camera intrinsic parameters such as the focal length may indeed change from image to image.

In a realistic situation, we usually only obtain "sampled" images of $\mathbf{x}(t)$ at some time instances, say $t_1, t_2, \ldots, t_m \in \mathbb{R}$. For simplicity we denote

$$\lambda_i = \lambda(t_i), \quad \mathbf{x}_i = \mathbf{x}(t_i), \quad \Pi_i = A(t_i)Pg(t_i). \tag{3.2}$$

The matrix $\Pi_i$ is then a $3 \times 4$ matrix which relates the $i^{th}$ image of the point $p$ to its world coordinates $\mathbf{X}$ by

$$\boxed{\mathbf{x}_i\lambda_i = \Pi_i\mathbf{X}} \tag{3.3}$$

for $i = 1, \ldots, m$. In the above equations, it is easy to see that, except $\mathbf{x}_i$'s, everything else are unknown and subject to recovery. Now we further assume that the feature point $p$ (in case we observe more than one of it) must lie on a plane $\mathbf{P}$ in 3-D. This plane can be described by a vector $\pi = [a, b, c, d] \in \mathbb{R}^4$ such that the coordinates $\mathbf{X}$ of any point on this plane further satisfies

$$\boxed{\pi\mathbf{X} = 0} \tag{3.4}$$

Although we assume such an constraint on the 3-D coordinates $\mathbf{X}$ of $p$ exists, but we do not assume that we know $\pi$ in advance.

In general, solving $\lambda_i$'s, $\Pi_i$'s, $v$ and $\mathbf{X}$ altogether from such equations is extremely difficult. A natural way to simplify the task is to decouple the recovery of the matrices $\Pi_i$'s from recovery of $\lambda_i$'s and $\mathbf{X}$. For that purpose, let us rewrite the system of equations (3.3) and equation (3.4) in a single matrix form

$$\begin{array}{rcl} \mathcal{I}\vec{\lambda} & = & \Pi\mathbf{X} \\ \begin{bmatrix} \mathbf{x}_1 & 0 & \cdots & 0 \\ 0 & \mathbf{x}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{x}_m \\ 0 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_m \end{bmatrix} & = & \begin{bmatrix} \Pi_1 \\ \Pi_2 \\ \vdots \\ \Pi_m \\ \pi \end{bmatrix} \mathbf{X} \end{array} \tag{3.5}$$

For obvious reasons, we will call $\vec{\lambda} \in \mathbb{R}^{3m}$ the *depth scale vector*, and $\Pi \in \mathbb{R}^{(3m+1)\times 4}$ the *projection matrix* associated to the *image matrix* $\mathcal{I} \in \mathbb{R}^{(3m+1)\times m}$.

From the above equation, one should notice that, in order to eliminate the unknowns $\vec{\lambda}$ and $\mathbf{X}$, the *only* relationship still holds between $\mathcal{I}$ and $\Pi$ is that the $m+4$ column vectors of the following $(3m+1) \times (m+4)$ matrix

$$N_p := [\Pi, \mathcal{I}] = \begin{bmatrix} \Pi_1 & \mathbf{x}_1 & 0 & \cdots & 0 \\ \Pi_2 & 0 & \mathbf{x}_2 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ \Pi_m & 0 & \cdots & 0 & \mathbf{x}_m \\ \pi & 0 & \cdots & \cdots & 0 \end{bmatrix} \tag{3.6}$$

are *linearly dependent* or

$$\boxed{\operatorname{rank}(N_p) \leq m + 3} \tag{3.7}$$

since it is clear from the equation (3.5) that the vector $v := [\mathbf{X}^T, -\vec{\lambda}^T,]^T \in \mathbb{R}^{m+4}$ is in the null space of the matrix $N_p$.

Without loss of generality, we may assume that the first camera frame is chosen to be the reference frame.[1] That gives the projection matrices $\Pi_i, i = 1, \ldots, m$

$$\Pi_1 = [I, 0], \quad \ldots, \quad \Pi_m = [R_m, T_m] \quad \in \mathbb{R}^{3\times 4}, \tag{3.8}$$

[1] Depending on the context, the reference frame could be either an Euclidean, Affine or Projective reference frame. In any case, the projection matrix for the first image becomes the standard projection matrix $[I, 0] \in \mathbb{R}^{3\times 4}$. The reader should notice that we do not lose any generality for doing this.

where $R_i \in \mathbb{R}^{3\times 3}, i = 2, \ldots, m$ are the first three columns of $\Pi_i$ and $T_i \in \mathbb{R}^3, i = 2, \ldots, m$ are the fourth column of $\Pi_i$. For $\pi = [a, b, c, d]$, define $\pi^1 = [a, b, c] \in \mathbb{R}^3$ and $\pi^2 = d \in \mathbb{R}$.

With the above notation, after a row manipulation which eliminates $\mathbf{x}_1$ from the first row of $N_p$, it is easy to see that $N_p$ has the same rank as the matrix in $\mathbb{R}^{(3m+1)\times(m+4)}$

$$\begin{bmatrix} I & 0 & 0 & 0 & \cdots & 0 \\ R_2 & T_2 & R_2\mathbf{x}_1 & \mathbf{x}_2 & \ddots & \vdots \\ \vdots & \vdots & \vdots & 0 & \ddots & 0 \\ R_m & T_m & R_m\mathbf{x}_1 & 0 & 0 & \mathbf{x}_m \\ \pi^1 & \pi^2 & \pi^1\mathbf{x}_1 & 0 & \cdots & 0 \end{bmatrix} = \left[\begin{array}{c|c} I & 0 \\ \hline R_2 & \\ \vdots & N_p' \\ R_m & \\ \pi^1 & \end{array}\right].$$

Multiplying the sub-matrix $N_p' \in \mathbb{R}^{(3m-2)\times(m+1)}$ on the left by the following matrix[2]

$$D_p' = \begin{bmatrix} \mathbf{x}_2^T & 0 & \cdots & 0 & 0 \\ \widehat{\mathbf{x}_2} & 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & \mathbf{x}_m^T & 0 \\ 0 & \cdots & 0 & \widehat{\mathbf{x}_m} & 0 \\ 0 & \cdots & 0 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{(4m-3)\times(3m-2)}$$

yields the matrix $D_p'N_p'$ in $\mathbb{R}^{(4m-3)\times(m+1)}$

$$\begin{bmatrix} \mathbf{x}_2^T T_2 & \mathbf{x}_2^T R_2\mathbf{x}_1 & \mathbf{x}_2^T\mathbf{x}_2 & 0 & 0 & 0 \\ \widehat{\mathbf{x}_2}T_2 & \widehat{\mathbf{x}_2}R_2\mathbf{x}_1 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & 0 & \ddots & 0 & 0 \\ \vdots & \vdots & 0 & 0 & \ddots & 0 \\ \mathbf{x}_m^T T_m & \mathbf{x}_m^T R_m\mathbf{x}_1 & 0 & 0 & 0 & \mathbf{x}_m^T\mathbf{x}_m \\ \widehat{\mathbf{x}_m}T_m & \widehat{\mathbf{x}_m}R_m\mathbf{x}_1 & 0 & 0 & 0 & 0 \\ \pi^2 & \pi^1\mathbf{x}_1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Since $D_p'$ is of full rank $3m - 2$, it yields $\text{rank}(N_p') = \text{rank}(D_p'N_p')$. Hence the original matrix $N_p$ is rank deficient if and only if the following sub-matrix of $D_p'N_p'$

$$M_p = \begin{bmatrix} \widehat{\mathbf{x}_2}T_2 & \widehat{\mathbf{x}_2}R_2\mathbf{x}_1 \\ \widehat{\mathbf{x}_3}T_3 & \widehat{\mathbf{x}_3}R_3\mathbf{x}_1 \\ \vdots & \vdots \\ \widehat{\mathbf{x}_m}T_m & \widehat{\mathbf{x}_m}R_m\mathbf{x}_1 \\ \pi^2 & \pi^1\mathbf{x}_1 \end{bmatrix} \in \mathbb{R}^{(3m-2)\times 2} \tag{3.9}$$

is rank deficient. The matrix $M_p$ is called the *multiple view matrix* for a planar point feature $p \in \mathbf{P}$. Notice that $M_p$ differs from that for a generic point feature only by adding the extra last row. From the above derivation, we have the following:

**Theorem 9 (Rank deficiency for planar point features).** *For the two matrices $N_p$ and $M_p$ defined above we always have*

$$\boxed{\textit{rank}(M_p) = \textit{rank}(N_p) - (m + 2) \leq 1.} \tag{3.10}$$

*That is $\textit{rank}(N_p) = m + 3$ if and only if $\textit{rank}(M_p) = 1$, and $\textit{rank}(N_p) = m + 2$ if and only if $\textit{rank}(M_p) = 0$.*

[2]For a three dimensional vector $u \in \mathbb{R}^3$, we use $\widehat{u} \in \mathbb{R}^{3\times 3}$ to denote the skew symmetric matrix associated to $u$ such that for any vector $v \in \mathbb{R}^3$, we have $\widehat{u}v = u \times v$.

Since the $M_p$ matrix is of rank less than 1, it is necessary that any of its sub-matrix is of rank less than 1. This immediately gives the bilinear constraints

$$\mathbf{x}_i^T \widehat{T}_i R_i \mathbf{x}_1 = 0 \tag{3.11}$$

and trilinear constraints

$$\widehat{\mathbf{x}}_i (T_i \mathbf{x}_1^T R_j^T - R_i \mathbf{x}_1 T_j^T) \widehat{\mathbf{x}_j} = 0 \tag{3.12}$$

for $i, j = 2, \ldots, m$. In addition to those, we also obtain extra constraints due to the planar condition (by considering the sub-matrix consisting of the $i^{th}$ group of three rows of $M_p$ and its last row)

$$\boxed{\widehat{\mathbf{x}}_i T_i \pi^1 \mathbf{x}_1 - \widehat{\mathbf{x}}_i R_i \mathbf{x}_1 \pi^2 = 0.} \tag{3.13}$$

When the plane $\mathbf{P}$ does not cross the camera center $o_1$, *i.e.*, $\pi^2 \neq 0$, these equations give exactly the well-known homography constraints for planar image feature points

$$\widehat{\mathbf{x}}_i \left( R_i - \frac{1}{\pi^2} T_i \pi^1 \right) \mathbf{x}_1 = 0 \tag{3.14}$$

between the $1^{st}$ and the $i^{th}$ views. The matrix $H_i = \left( R_i - \frac{1}{\pi^2} T_i \pi^1 \right)$ in the equation is the well-known *homography matrix* between the two views. In any case, there is *not* any non-trivial constraints among any four views. From the above discussion, we see that the $M_p$ matrix being of rank less than 1 is a much more unified way to describe all the constraints among multiple images from a planar scene.

## 3.3 The multiple view matrix for a line on a plane

Now consider a line $L \subset \mathbb{E}^3$, defined by an equation $L = \{\mathbf{X} \mid \mathbf{X} = \mathbf{X}_0 + \lambda v\}$, where $\mathbf{X}_0 = [X_0, Y_0, Z_0, 1]^T \in \mathbb{R}^4$ is a base point on this line, $v = [v_1, v_2, v_3, 0]^T \in \mathbb{R}^4$ is a non-zero vector indicating the direction of the line, and $\lambda \in \mathbb{R}$. An image $\mathbf{l}(t) = [x(t), y(t), z(t)]^T \in \mathbb{R}^3$ of $L$ taken by a moving camera satisfies the following equation

$$\mathbf{l}(t)^T \mathbf{x}(t) = \mathbf{l}(t)^T A(t) P g(t) \mathbf{X} = 0 \tag{3.15}$$

where $\mathbf{x}(t)$ is the image of the point $\mathbf{X} \in L$ at time $t$. Since the above equation holds for any point $\mathbf{X}$ on the line $L$, it yields

$$\mathbf{l}(t)^T A(t) P g(t) \mathbf{X}_0 = \mathbf{l}(t)^T A(t) P g(t) v = 0. \tag{3.16}$$

In the above equations, all $\mathbf{x}, \mathbf{X}$ and $g$ are in homogeneous representation. It is worth knowing that geometrically $\mathbf{l}(t)$ is the normal vector of the plane formed by the optical center $o$ and the line $L$ (in 3-D) at time $t$.

In a realistic situation, we usually only obtain "sampled" images of $\mathbf{l}(t)$ at some time instances: $t_1, t_2, \ldots, t_m$. For simplicity we denote $\mathbf{l}_i = \mathbf{l}(t_i), \Pi_i = A(t_i) P g(t_i)$. The matrix $\Pi_i$ is then a $3 \times 4$ matrix which relates the $i^{th}$ image of the line $L$ to its world coordinates $(\mathbf{X}_0, v)$ by

$$\boxed{\mathbf{l}_i^T \Pi_i \mathbf{X}_0 = \mathbf{l}_i^T \Pi_i v = 0} \tag{3.17}$$

for $i = 1, \ldots, m$. In the above equations, it is easy to see that, except $\mathbf{l}_i$'s, everything else are unknown and subject to recovery. Now assume that the line $L$ belongs to the plane $\mathbf{P}$, we also have

$$\boxed{\pi \mathbf{X}_0 = \pi v = 0} \tag{3.18}$$

However, solving $\Pi_i$'s, $\pi$ and $L$ (*i.e.*, $(\mathbf{X}_0, v)$) directly from such equations is extremely difficult. A natural way to simplify the task is to exploit the rank deficiency condition from the following two equations $N_l \mathbf{X}_0 = 0$ and $N_l v = 0$ where

$$N_l = \begin{bmatrix} \mathbf{l}_1^T \Pi_1 \\ \mathbf{l}_2^T \Pi_2 \\ \vdots \\ \mathbf{l}_m^T \Pi_m \\ \pi \end{bmatrix} \in \mathbb{R}^{(m+1)\times 4}. \tag{3.19}$$

Since $\mathbf{X}_0$ and $v$ are in general linearly independent, we have

$$\boxed{\text{rank}(N_l) \leq 2} \tag{3.20}$$

Without loss of generality, we may assume that the first camera frame is chosen to be the reference frame. Then the projection matrices $\Pi_i, i = 1, \ldots, m$ are of the form given in (3.8). The matrix $N_l$ now is

$$N_l = \begin{bmatrix} \mathbf{l}_1^T & 0 \\ \mathbf{l}_2^T R_2 & \mathbf{l}_2^T T_2 \\ \vdots & \vdots \\ \mathbf{l}_m^T R_m & \mathbf{l}_m^T T_m \\ \pi^1 & \pi^2 \end{bmatrix} \in \mathbb{R}^{(m+1)\times 4}. \tag{3.21}$$

This matrix should have a rank of no more than 2. Multiplying $N_l$ on the right by the following matrix

$$D_l = \begin{bmatrix} \widehat{\mathbf{l}_1} & \mathbf{l}_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{4\times 5} \tag{3.22}$$

yields a matrix in $\mathbb{R}^{(m+1)\times 5}$:

$$N_l' = \begin{bmatrix} 0 & \mathbf{l}_1^T \mathbf{l}_1 & 0 \\ \mathbf{l}_2^T R_2 \widehat{\mathbf{l}_1} & \mathbf{l}_2^T R_2 \mathbf{l}_1 & \mathbf{l}_2^T T_2 \\ \vdots & \vdots & \vdots \\ \mathbf{l}_m^T R_m \widehat{\mathbf{l}_1} & \mathbf{l}_m^T R_m \mathbf{l}_1 & \mathbf{l}_m^T T_m \\ \pi^1 \widehat{\mathbf{l}_1} & \pi^1 \mathbf{l}_1 & \pi^2 \end{bmatrix}. \tag{3.23}$$

Since $D_l$ is of full rank 4, we have $\text{rank}(N_l') = \text{rank}(N_l) \leq 2$. Obviously, this is true if and only if the following sub-matrix of $N_l'$

$$M_l = \begin{bmatrix} \mathbf{l}_2^T R_2 \widehat{\mathbf{l}_1} & \mathbf{l}_2^T T_2 \\ \mathbf{l}_3^T R_3 \widehat{\mathbf{l}_1} & \mathbf{l}_3^T T_3 \\ \vdots & \vdots \\ \mathbf{l}_m^T R_m \widehat{\mathbf{l}_1} & \mathbf{l}_m^T T_m \\ \pi^1 \widehat{\mathbf{l}_1} & \pi^2 \end{bmatrix} \in \mathbb{R}^{m\times 4} \tag{3.24}$$

has rank no more than one. The matrix $M_l$ is called the *multiple view matrix* for a planar line feature $L \subset \mathbf{P}$. Notice that $M_l$ differs from that for a generic line feature only by adding the extra last row. From the above derivation, we have the following:

**Theorem 10 (Rank deficiency for planar line features).** *For the two matrices $N_l$ and $M_l$ defined above, we have*

$$\boxed{\textit{rank}(M_l) = \textit{rank}(N_l) - 1 \leq 1.} \tag{3.25}$$

*That is $\textit{rank}(N_l) = 2$ if and only if $\textit{rank}(M_l) = 1$, and $\textit{rank}(N_l) = 1$ if and only if $\textit{rank}(M_l) = 0$.*

It is then obvious that, according to the rank condition, any two rows of $M_l$ must be linearly dependent. Hence, in addition to the trilinear constraints

$$\mathbf{l}_j^T T_j \mathbf{l}_i^T R_i \widehat{\mathbf{l}_1} - \mathbf{l}_i^T T_i \mathbf{l}_j^T R_j \widehat{\mathbf{l}_1} = 0 \tag{3.26}$$

for $i, j = 2, \ldots, m$, we also obtain the homography constraint in terms of line feature (by considering the $i^{th}$ row of $M_l$ and its last row)

$$\boxed{\mathbf{l}_i^T R_i \widehat{\mathbf{l}_1} \pi^2 - \mathbf{l}_i^T T_i \pi^1 \widehat{\mathbf{l}_1} = 0.} \tag{3.27}$$

When $\pi^2 \neq 0$, we can further write the above equation in a more familiar form

$$\mathbf{l}_i^T \left( R_i - \frac{1}{\pi^2} T_i \pi^1 \right) \widehat{\mathbf{l}_1} = 0, \quad \forall i = 2, \ldots, m \tag{3.28}$$

as homography between the $1^{st}$ and the $i^{th}$ views. As before, the matrix $H_i = \left( R_i - \frac{1}{\pi^2} T_i \pi^1 \right)$ in the equation is the homography matrix between the two views. In any case, there is *not* any non-trivial constraints among any four views. As in the point case, we see that the $M_l$ matrix being of rank less than 1 is a much more unified way to describe all the constraints among image lines from a planar scene.

## 3.4 Dual geometric relationships between coplanar point and line

The duality between the constraints for planar point features and line features is already clear from the two homography equations (3.13) and (3.27) for points and lines respectively. More specifically, we see that the role of a pair of image points $(\mathbf{x}_1, \mathbf{x}_i)$ is exactly equivalent to that of a pair of image lines $(\mathbf{l}_i, \mathbf{l}_1)$.

We are now ready to establish a complete duality between a pair of lines and a pair of points on the same plane $\mathbf{P}$. In the first case, two points determine a line. The two $M_p$ matrices associated to the two points then determine the $M_l$ matrix associated to the line. This has been proven for the general case in the preceding chapter. As we pointed out in there, the opposite direction is not always true for two generic lines in 3-D since they do not necessarily intersect at one point unless they are coplanar.

Now, let $p$ be a point in space and $L_1, L_2$ be two distinct lines intersect at $p$. Assume the equation of the plane formed by $L_1$ and $L_2$ is $\pi \mathbf{X} = 0$ with $\mathbf{X}$ the homogeneous coordinates for the point $p$. As before, let $\pi = [\pi^1, \pi^2] \in \mathbb{R}^4$, $\pi^1 = [a, b, c] \in \mathbb{R}^3$ and $\pi^2 = d \in \mathbb{R}$. Let $\mathbf{l}_i^j$ $(i = 1, \ldots, m,\ j = 1, 2)$ be the $i^{th}$ images of the $j^{th}$ line. Then the $i^{th}$ image of $p$ is $\mathbf{x}_i = \lambda_i \widehat{\mathbf{l}_i^1} \mathbf{l}_i^2$ and $\widehat{\mathbf{x}_i} = \lambda_i (\mathbf{l}_i^2 \mathbf{l}_i^{1\,T} - \mathbf{l}_i^1 \mathbf{l}_i^{2\,T})$. Associated to the two lines are two $M_l^1$ and $M_l^2$ matrices and associated to the point is an $M_p$ matrix. We now show the rank condition for $M_l^1, M_l^2$ implies that for $M_p$. From the rank deficiency condition for planar line features , we have $\mathbf{l}_i^{1\,T} R_i \widehat{\mathbf{l}_1^1} \pi^2 - \mathbf{l}_i^{1\,T} T_i \pi^1 \mathbf{l}_1^1 = 0$ and $\mathbf{l}_i^{2\,T} R_i \widehat{\mathbf{l}_1^2} \pi^2 - \mathbf{l}_i^{2\,T} T_i \pi^1 \mathbf{l}_1^2 = 0$. This gives

$$\widehat{\mathbf{x}_i} R_i \mathbf{x}_1 \pi^2 - \widehat{\mathbf{x}_i} T_i \pi^1 \mathbf{x}_1 = 0, \ \forall i = 2, \ldots, m. \tag{3.29}$$

This means the two columns of $M_p$ are linear dependent, with a ratio $-\frac{\pi^1 \mathbf{x}_1}{\pi^2}$. This ratio exactly corresponds to the depth of the feature point $p$. Therefore, we have proven for two coplanar lines

$$\text{rank}(M_l^1) \leq 1 \ \& \ \text{rank}(M_l^2) \leq 1 \quad \Rightarrow \quad \text{rank}(M_p) \leq 1. \tag{3.30}$$

So we can concisely describe the duality between coplanar points and lines in terms of the multiple view matrix:

**Theorem 11 (Duality between coplanar points and lines).** *The $M_p$ matrices associated two distinct points on a plane being of rank less than 1 implies the $M_l$ matrix associated to the line determined by them being of rank less than 1. On the other hand, the $M_l$ matrices associated two distinct lines on a plane being of rank less than 1 determines the $M_p$ matrix associated to the intersection of the two lines being of rank less than 1.*

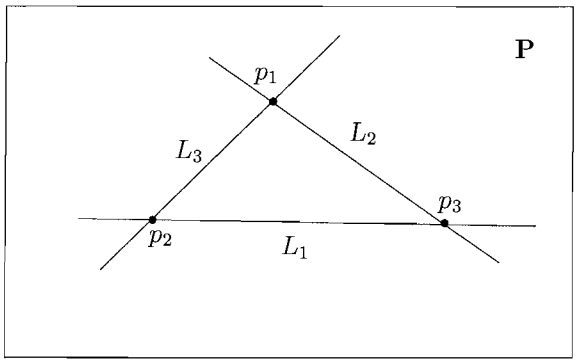


Figure 3.1: Duality between a set of three points and three lines in a plane $\mathbf{P}$: the rank deficiency condition for $p_1, p_2, p_3$ is exactly equivalent that for $L_1, L_2, L_3$.

An immediate implication of this theorem is that given a set of feature points sharing the same 3-D plane, it really does not matter too much whether one uses the $M_p$ matrices for the points, or one uses the $M_l$ matrices for the lines determined by pairwise points (in all the views). They essentially give exactly the same set of constraints. This is illustrated in Figure 3.1.

Although geometric meanings for the multiple view matrices $M_p$ and $M_l$ have been discussed in the previous two chapters respectively, we now give a more specific geometric interpretation for planar $M_p$ and $M_l$ matrices. The key here is to notice that whatever an $M_p$ or $M_l$ represents (a sphere or a circle) for a generic point or line, it now must take intersection with the 3-D plane where the feature belongs. In the case of a point on the plane, $M_p$ indicates a circle as the intersection of the plane and a sphere (had the point been considered as generic). Hence two $M_p$ matrices of the same point relative to two distinct camera frames determine the point $p$ up to two solutions (as the intersection of two circles on the plane). Naturally, three such matrices in general pin-point the exact location of the point $p$. In the case of a line, $M_l$ indicates two points as the intersection of the plane and a circle (had the line been considered as generic). Hence two $M_l$ matrices of the same line relative to two distinct camera frames determine the exact location of the line on the plane. We illustrate this in Figure 3.2.

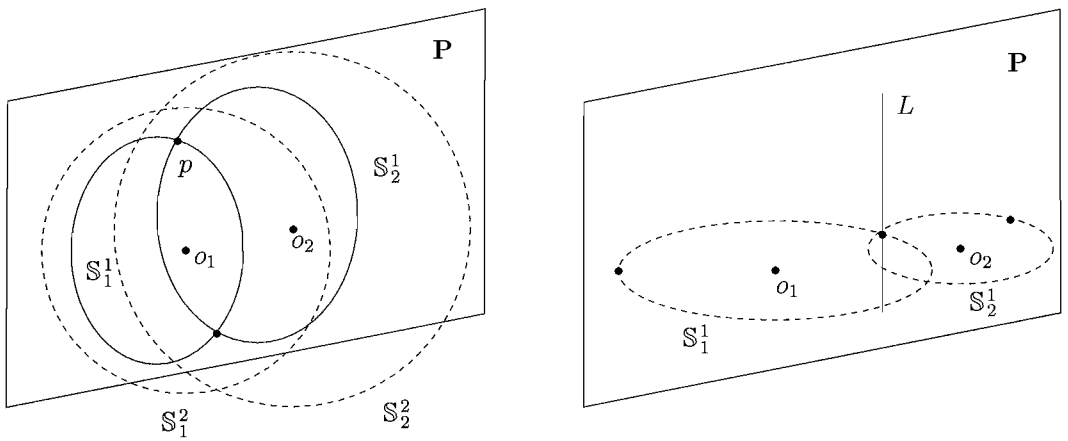


Figure 3.2: Geometry of planar multiple view matrices for a point or line relative to two distinct camera frames. The generic $M_p$ matrices for a point $p$ relative to $o_1, o_2$ correspond to two spheres $\mathbb{S}^2$. They intersect with the plane $\mathbf{P}$ at two circles $\mathbb{S}^1$ which further intersect at two candidate positions for $p$ on the plane. The generic $M_l$ matrices for a line $L$ relative to $o_1, o_2$ correspond to two circles $\mathbb{S}^1$. Each of them intersects with the plane $\mathbf{P}$ at two points. The common point gives the only location of the line $L$ on the plane.

Here, we observe an interesting fact that although the rank deficiency condition for the $M_p$ or $M_l$ implies

each other in the planar case, an $M_l$ matrix does preserve more information than an $M_p$ matrix: $M_l$ keeps not only the distance of a line but also its direction while $M_p$ only keeps the distance of a point. This explains why a less number of $M_l$ matrices are needed to locate a line on the plane than that needed for a point.

If the rank of the multiple view matrix $M$ is indeed of less than 1 for some multiple images of a point or a line on a plane, what does that mean geometrically? As a corollary to the results given for generic situations in the previous two chapters, we immediately have the following results regarding the uniqueness of the pre-image in terms of the rank of the $M$ matrix:

**Theorem 12 (Pre-image for planar point features).** *Given $m$ vectors $\mathbf{x}_1, \dots \mathbf{x}_m \in \mathbb{R}^3$ with respect to $m$ camera frames and given a 3-D plane $\mathbf{P}$, these vectors correspond to the same point in the plane if and only if the rank of the $M_p$ matrix is 1. If the rank is 0, the point is determined up to a line in the plane $\mathbf{P}$ on which all the camera centers must also lie.*

**Theorem 13 (Pre-image for planar line features).** *Given $m$ vectors $\mathbf{l}_1, \dots \mathbf{l}_m \in \mathbb{R}^3$ with respect to $m$ camera frames and given a 3-D plane $\mathbf{P}$, these vectors correspond to the same line in the plane if and only if the rank of the $M_l$ matrix is 1. If the rank is 0, the line is determined up to the same plane $\mathbf{P}$ on which all the camera centers must also lie.*

## 3.5 Euclidean structure from motion using coplanar features

The rank deficiency condition on $M_p$ and $M_l$ for coplanar points and lines allow us to utilize simultaneously multilinear constraints and homography among multiple images for purposes such as image matching, mapping images to a new view and recovering 3-D motion and structure. Existing algorithms for planar features typically exploit only the homography which is only part of all the constraints. Although homography is algebraically equivalent to the rank condition on the multiple view matrix, simulations show that algorithms based on the rank condition is numerically more stable, since they exploits both motion and structure information. Furthermore, homography is typically between pairwise views and it is well-known that for two views of a planar scene there are typically two physically possible solutions for the structure from motion problem [3]. As we will soon see, one of the advantages of using multiple views for a planar scene is that such ambiguous solution can be easily eliminated.

The duality between the rank deficiency conditions for point and line given by Theorem 11 implies that it does not matter whether we choose to use point or line features for a planar scene. The resulting algorithms should be algebraically equivalent. Therefore, although algorithms for generic 3-D points and lines were developed separately in the previous chapters, they become exactly the same when specialized to coplanar points and lines. Hence, we here demonstrate only for point features how to extend the generic linear algorithm given in Chapter 1 for motion and structure estimation to the planar case. For simplicity, we assume that the camera is perfectly calibrated, hence $(R, T) \in SE(3)$ and the projection matrix $\Pi$ corresponds to the actual Euclidean transformation between camera frames.

Now suppose that $m$ images $\mathbf{x}_1^i, \dots, \mathbf{x}_m^i$ of $n$ points $p^i$, $i = 1, \dots, n$ lying on a plane $\pi$ are given and we want to use them to estimate the unknown projection matrix $\Pi$ and the parameters of the plane. The rank deficiency condition of the $M$ matrix can be written as:

$$\alpha^i \begin{bmatrix} \widehat{\mathbf{x}_2^i} T_2 \\ \widehat{\mathbf{x}_3^i} T_3 \\ \vdots \\ \widehat{\mathbf{x}_m^i} T_m \\ \pi^2 \end{bmatrix} + \begin{bmatrix} \widehat{\mathbf{x}_2^i} R_2 \mathbf{x}_1^i \\ \widehat{\mathbf{x}_3^i} R_3 \mathbf{x}_1^i \\ \vdots \\ \widehat{\mathbf{x}_m^i} R_m \mathbf{x}_1^i \\ \pi^1 \mathbf{x}_1^i \end{bmatrix} = 0 \tag{3.31}$$

for proper $\alpha^i \in \mathbb{R}, i = 1, \dots n$.

From (3.1) we have $\lambda_j^i \mathbf{x}_j^i = \lambda_1^i R_j \mathbf{x}_1^i + T_j$. Multiplying by $\widehat{\mathbf{x}_j^i}$ we obtain $\widehat{\mathbf{x}_j^i}(R_j \mathbf{x}_1^i + T_j/\lambda_1^i) = 0$. Therefore $\alpha^i = 1/\lambda_1^i$ can be interpreted as the inverse of the depth of point $p^i$ with respect to the first frame. The set of

equations in (3.31) is equivalent to finding vectors $\pi \in \mathbb{R}^4$, $\vec{R}_j = [r_{11}, r_{12}, r_{13}, r_{21}, r_{22}, r_{23}, r_{31}, r_{32}, r_{33}]^T \in \mathbb{R}^9$ and $\vec{T}_j = T_j \in \mathbb{R}^3$, $j = 2, \ldots, m$, such that:

$$Q\pi^T := \begin{bmatrix} \mathbf{x}_1^{1T} & \alpha^1 \\ \mathbf{x}_1^{2T} & \alpha^2 \\ \vdots & \vdots \\ \mathbf{x}_1^{nT} & \alpha^n \end{bmatrix} \pi^T = 0, \tag{3.32}$$

$$P_j \begin{bmatrix} \vec{T}_j \\ \vec{R}_j \end{bmatrix} := \begin{bmatrix} \alpha^1 \widehat{\mathbf{x}_j^1} & \widehat{\mathbf{x}_j^1} * \mathbf{x}_1^{1T} \\ \alpha^2 \widehat{\mathbf{x}_j^2} & \widehat{\mathbf{x}_j^2} * \mathbf{x}_1^{2T} \\ \vdots & \vdots \\ \alpha^n \widehat{\mathbf{x}_j^n} & \widehat{\mathbf{x}_j^n} * \mathbf{x}_1^{nT} \end{bmatrix} \begin{bmatrix} \vec{T}_j \\ \vec{R}_j \end{bmatrix} = 0 \tag{3.33}$$

where $A * B$ denotes the *Kronecker product* of the two matrices $A$ and $B$, and $Q \in \mathbb{R}^{n \times 4}$ and $P_j \in \mathbb{R}^{3n \times 12}$ are defined from the equations properly.

Now we illustrate how to use the above formulation to develop two algorithms for motion (and structure) recovery in the case that the camera is calibrated. The extension to affine or projective camera model would be straightforward.

Given the first two images of (at least) four points in general configuration, $\pi \in \mathbb{R}^4$, $T_2 \in \mathbb{R}^3$ and $R_2 \in SO(3)$ can be estimated using the standard *four point planar algorithm* [3]. In general, there are two physically possible solutions for $(\pi, R_2, T_2)$ from the four point algorithm, with $\pi^1, \pi^2$ recovered with the same scale - notice that in the homography equation, $\pi$ shows up as $\pi^1/\pi^2$. This scale can be easily fixed by choosing $\|\pi^1\| = 1$ and $\pi^2 = 1$. [3] Given these two solutions for $(\pi, R_2, T_2)$, we can solve for $\alpha$ from the equations in the first and last rows of (3.31). These equations are $\alpha^i \widehat{\mathbf{x}_2^i} T_2 = -\widehat{\mathbf{x}_2^i} R_2 \mathbf{x}_1^i$ and $\alpha^i \pi^2 = -\pi^1 \mathbf{x}_1^i$, whose least squares solution up to scale (the inverse of the common scale of $\pi^2$ and $T_2$) is given by:

$$\alpha^i = -\frac{(\widehat{\mathbf{x}_2^i} T_2)^T \widehat{\mathbf{x}_2^i} R_2 \mathbf{x}_1^i + \pi^2 \pi^1 \mathbf{x}_1^i}{\|\widehat{\mathbf{x}_2^i} T_2\|^2 + (\pi^2)^2} \tag{3.34}$$

for $i = 1, \ldots, n$.

Given these initial values of $\alpha^i$, equations (3.32) and (3.33) become linear, thus one can solve for the rest of $(R_j, T_j)$ and re-estimate $\pi$. Since there are two possible values for $(\pi, R_2, T_2)$ from the four point planar algorithm, there are two possible values for $\alpha$. Therefore, in principle there are two possible solutions[4] for $(\pi, R_j, T_j)$ provided that $\text{rank}(P_j) = 11$ and $\text{rank}(Q) = 3$. One can show that if the feature points are in a general position in 3-D, the rank of $P_j \in \mathbb{R}^{3n \times 12}$ is 11 provided that $n \geq 6$. However, since here all points lie on the same plane, the maximum rank of $P_j$ becomes 8 for arbitrary $n \geq 4$ points in a general configuration on the plane, while the rank of $Q$ is always 3. It is straightforward to verify that the solution $[\vec{T}_j^T, \vec{R}_j^T]^T \in \mathbb{R}^{12}$ is in the four dimensional kernel of $P_j$ which is spanned by the columns of the following matrix in $\mathbb{R}^{12 \times 4}$:

$$K_j := \begin{bmatrix} \pi^2 & 0 & 0 & 0 \\ 0 & \pi^2 & 0 & 0 \\ 0 & 0 & \pi^2 & 0 \\ \pi^{1T} & 0_{3\times 1} & 0_{3\times 1} & r_1 - \frac{T_{j1}}{\pi^2} \pi^{1T} \\ 0_{3\times 1} & \pi^{1T} & 0_{3\times 1} & r_2 - \frac{T_{j2}}{\pi^2} \pi^{1T} \\ 0_{3\times 1} & 0_{3\times 1} & \pi^{1T} & r_3 - \frac{T_{j3}}{\pi^2} \pi^{1T} \end{bmatrix},$$

where $[r_1^T, r_2^T, r_3^T]^T := \vec{R}_j$ and $[T_{j1}, T_{j2}, T_{j3}]^T := \vec{T}_j$. The last column yields exactly the homography matrix $H_j := (R_j - \frac{1}{\pi^2} T_j \pi^1) \in \mathbb{R}^{3\times 3}$ between the $j^{th}$ and the $1^{st}$ views. Therefore, we have proved the following:

[3] This choice puts the plane $\pi$ at a unit distance from the origin of the first camera frame.

[4] We will shortly show that the solution is actually unique if $m \geq 3$.

**Lemma 4.** *The rank deficiency condition for planar point features rank$(M_p) = 1$ (see (3.9)) is equivalent to the set of homography constraints (3.13).*

Generic v.s. planar scene] Notice that the dimension of the kernel of $P_j$ is 4 instead of 1 is in fact the reason why the generic linear algorithm given in Chapter 1 becomes ill-conditioned for features from planar scene. However, motion and structure recovery from planar scene is still possible if the structure of the kernel of $P_j$ as given by the matrix $K_j$ is properly exploited. Unlike the generic case, the problem is typically nonlinear.

At this stage, the problem boils down to recovering $(R_j, T_j)$ from the homography matrix $H_j$. We can either assume that $\pi$ is unknown and use the four point planar algorithm, or assume that $\pi$ is known from solving (3.32).

In the first case, we obtain two solutions $(\pi_j, R_j, T_j)$ from image pairs $(\mathbf{x}_1^i, \mathbf{x}_j^i), i = 1, \ldots n$ for each $j = 2, \ldots, m$. Since $\pi_j^1 = \pi_2^1$, there are only two solutions for the plane $\pi$ and all relative motions $(R_j, T_j), j = 2, \ldots, m$, rather than the $2^{m-1}$ possible combinations. Furthermore, if $m \geq 3$, one can show that only one of these two solutions satisfies $\pi_j^1 = \pi_2^1, j = 3, \ldots, m$. Finally, all $T_j$ are recovered relative to the same choice of $\pi$. Hence it is straightforward to see that the relative translation scale between $T_j$ and $T_2$ is indeed $\|T_j\|/\|T_2\|$. Therefore, we have the following linear algorithm for motion and structure estimation from planar feature points:

**Algorithm 5 (Multiple view four point algorithm).** *Given $m$ images $\mathbf{x}_1^i, \ldots, \mathbf{x}_m^i$ of $n$ points $p^i$ in the 3-D space, $i = 1, \ldots, n$, we can estimate the motion $(R_j, T_j)$, $j = 2, \ldots, m$, the plane $\pi$ and the inverse structure $\alpha$ as follows:*

1. *Find the two solutions $(\pi_j, R_j, T_j), j = 2, \ldots, m$ from the four point planar algorithm subject to $\|\pi^1\| = \pi^2 = 1$.*
2. *If $m \geq 3$, find the unique solution satisfying $\pi_j^1 = \pi_2^1, j = 3, \ldots, m$.*
3. *Let $\pi = [\pi^1, 1]$.*
4. *Solve for $\alpha$ from (3.31) using linear least squares:*

$$\alpha^i = -\frac{\sum_{j=2}^m (\widehat{\mathbf{x}_j^i T_j})^T \widehat{\mathbf{x}_j^i} R_j \mathbf{x}_1^i + \pi^2 \pi^1 \mathbf{x}_1^i}{\sum_{j=2}^m \|\widehat{\mathbf{x}_j^i} T_j\|^2 + (\pi^2)^2} \tag{3.35}$$

*for $i = 1, \ldots, n$.*

Now we consider the second case in which $(\pi, R_2, T_2)$ hence $\alpha^i$ are given and we estimate $(R_j, T_j)$ only. First, we find $T_j \in \mathbb{R}^3$ such that $H_j + T_j \pi^1/\pi^2 \in O(3)$,[5] that is,

$$(H_j + T_j \pi^1/\pi^2)^T (H_j + T_j \pi^1/\pi^2) = I. \tag{3.36}$$

Then let $R_j = H_j + T_j \pi^1/\pi^2$. If $\det(R_j) = -1$, then flip the sign of both $R_j$ and $T_j$. Now once we get all the motions $(R_j, T_j)$, $\alpha^i$ can be re-estimated as in (3.35). Note that these $\alpha^i$'s are the same as that in (3.34) if $m = 2$. One can then recompute the motion given this new $\alpha^i$'s, until the error in updating $\alpha$ is small enough. We then have the following linear algorithm for multiple view motion and structure estimation:

**Algorithm 6 (Iterative SFM from planar features).** *Given $m$ images $\mathbf{x}_1^i, \ldots, \mathbf{x}_m^i$ of $n$ points $p^i$ in the 3-D space, $i = 1, \ldots, n$, we can estimate the motions $(R_j, T_j)$, $j = 2, \ldots, m$, the plane $\pi$ and the inverse structure $\alpha$ as follows:*

1. *Initialization: Set k=0; compute $\pi$ and $\alpha_k^i$ from Algorithm 1. Normalize so that $\alpha_k^1 = 1$.*
2. *Compute $\pi$ as the eigenvector associated to the smallest singular values of $Q$ and the homography matrix $H_j$ as the vector in the kernel of $P_j$ in the form of the last column of the matrix $K_j$, $j = 2, \ldots, m$, respectively.*

[5] It is known that if the matrix $R_j$ in $H_j$ is in $O(3)$, then the middle eigenvalue of $H_j$ should be 1. If not, one should scale $H_j$ properly before carrying on.

3. *Compute* $(R_j, T_j)$ *from* $H_j$ *using (3.36) for* $j = 2, \ldots, m$.

4. *Compute the new* $\alpha^i = \alpha^i_{k+1}$ *from (3.35). Normalize so that* $\alpha^1_{k+1} = 1$.

5. *If* $||\alpha_k - \alpha_{k+1}|| > \epsilon$, *for a pre-specified* $\epsilon > 0$, *then* $k = k + 1$ *and goto 2. Else stop.*

The camera motion is then $(R_j, T_j), j = 2, \ldots, m$, the plane is $\pi$ and the structure of the points (with respect to the first camera frame) is given by the converged depth scalar $\lambda^i_1 = 1/\alpha^i, i = 1, \ldots, n$.

Uniqueness of solution] It is known that the four point algorithm for two views of a planar scene gives two physically possible solutions [3]. One is the true solution and the second one corresponds to an exchange of the plane normal and the direction of translation. One should expect that the second solution can be further eliminated from a translation of the camera in a different direction. The consistency checking in the Step 2 of Algorithm 1 is designed for this purpose. Therefore, in general, there is usually a unique solution to Algorithm 1. As for Algorithm 2, the consistency is reflected through the kernel of the matrix $P_j$. For the $\pi$ from the second solution, there will be no valid solution for $R_j, T_j$ which satisfy the equation (3.36). Hence at Step 3 in the Algorithm 2, a simple comparison between the two solutions will eliminate the erroneous one.

## 3.6 Simulations on synthetic data

In this section, we show by simulations the performance of the Algorithm 5. Table 3.1 shows the simulation parameters used. In the table, u.f.l. stands for *units of focal length.* The number of frames is typically 3 (unless we vary it on purpose). The ratio of the magnitude of translation $||T||$ and rotation $\theta$, or simply the $T/R$ *ratio*, is compared at the center of the random cloud scattered in the truncated pyramid specified by the given field of view and depth variation. For all simulations, independent Gaussian noise with std given in pixels is added to each image point. In general, the amount of rotation between consecutive frames is given by a proper angle and the amount of translation is then automatically given by the $T/R$ ratio. We always choose the amount of total motion such that all feature points will stay in the field of view for all frames. In the following, camera motions will be specified by their translation and rotation axes. For example, between a pair of frames, the symbol $XY$ means that the translation is along the $X$-axis and rotation is along the $Y$-axis. If $n$ such symbols are connected by hyphens, it specifies a sequence of consecutive motions. Error measure for rotation is $\arccos\left(\frac{tr(R\tilde{R}^T)-1}{2}\right)$ in degrees where $\tilde{R}$ is an estimate of the true $R$. Error measure for translation is the angle between $T$ and $\tilde{T}$ in degrees where $\tilde{T}$ is an estimate of the true $T$.

Table 3.1: Simulation parameters

| Parameter | Unit | Value |
|---|---|---|
| Number of trials | | 1000 |
| Number of points | | 20 |
| Field of view | degrees | 90 |
| Depth | u.f.l. | 100 |
| Image size | pixels | $500 \times 500$ |

Figure 3.3 shows the motion estimation error as a function of the noise. As we can see, the proposed algorithm gives a pretty good estimation of the motion of the camera, with an error below 0.53 degrees even for large amounts of noise.

Figure 3.4 shows the relative translation scale error for a noise level of 5 pixels, as well as the inverse structure error. Again, the algorithm gives a very good estimation of both quantities.

Figure 3.5 compares the multiple view four point algorithm with the iterative algorithm for a typical motion sequence of 13 frames. From the results, the iterative scheme indeed improves all the structural estimates: the depth of the feature points and the location of the plane.

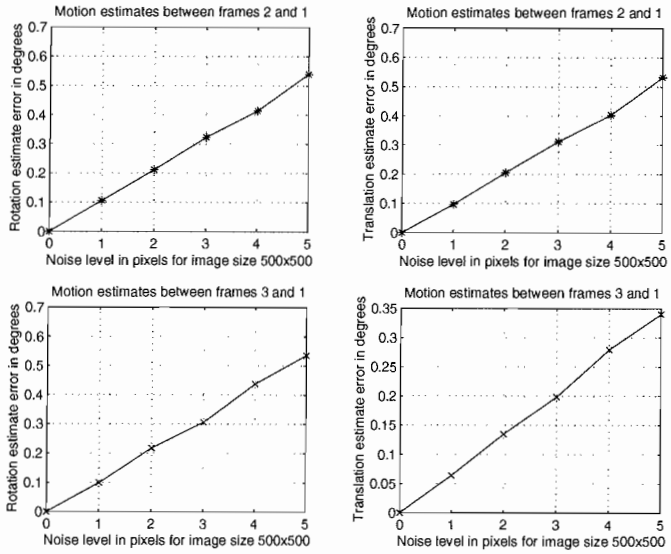


Figure 3.3: Motion estimate error. The number of trials is 1000, the plane is $\pi = [0\ 0\ 1\ -\ 100]$, camera motions are $XX$-$YY$, relative scale is 1.5 and $T/R$ ratio is 1.5

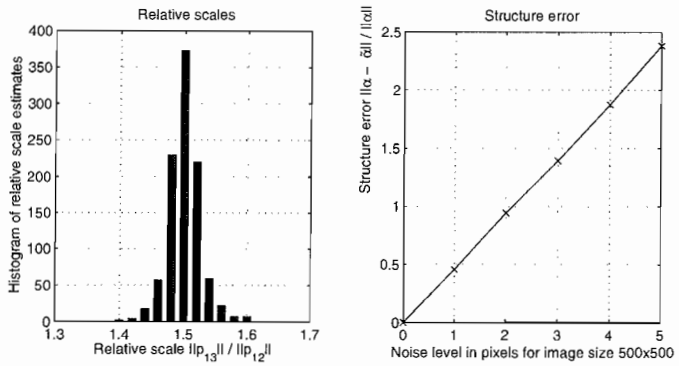


Figure 3.4: Relative scale and structure estimate error. The number of trials is 1000, the plane is $\pi = [0\ 0\ 1\ -\ 100]$, camera motions are $XX$-$YY$, relative scale is 1.5 and $T/R$ ratio is 1.5

Figure 3.6 compares the two algorithms from 3 to 16 frames for a typical motion sequence. As we expected, the structural estimates improve with the number of frames while the motion estimates remain very much the same. The apparent chattering in subplots 1 and 4 is due to the particular motion sequence we choose: a repetitive sequence of $XX$-$YY$-$ZZ$ motion. The higher error corresponds to the $ZZ$-motion is expected, since with the same amount of motion and noise, the signal-to-noise ratio (SNR) is typically much higher in this case.

## 3.7 Discussions and conclusions

In this chapter, we have studied in parallel multiple view geometry for point and line features on a 3-D plane. We see that for these coplanar features, the algebraic and geometric constraints governing their multiple views can be concisely described by certain rank deficient matrices: $M_p$ for point and $M_l$ for line. These matrices are natural extensions for those known for a generic point or line features. Their rank deficiency then captures both the multilinear constraints and the homographic constraints which are special to planar scene.

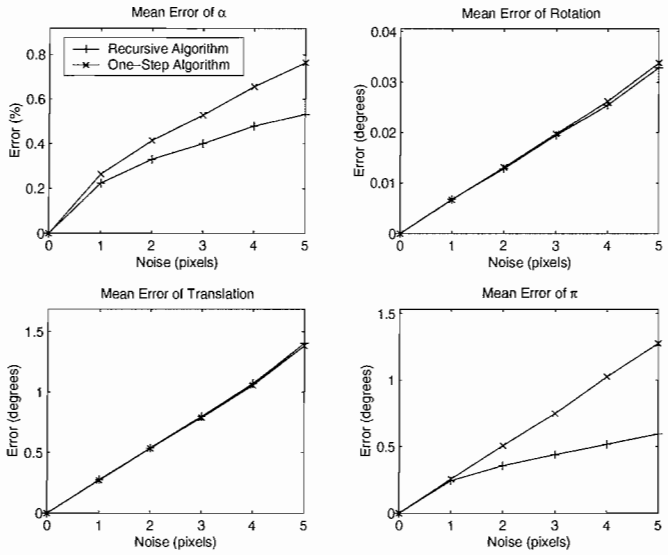


Figure 3.5: Motion and structure estimates error comparison. The number of trials is 1000, the plane is $\pi = [0\ 0\ 1\ -100]$, camera motions are $XX$-$YY$. Motion error is compared at the first motion.

Using the rank deficiency condition, a complete duality between coplanar point and line features is clearly revealed. In particular, we have shown that, in the planar case, the rank deficiency condition for point implies that for line, and vice versa. As a consequence of this duality, it is then equivalent to use either point or line to study the geometry or develop algorithms for multiple views of a planar scene.

For motion and structure recovery, we have shown how to use such rank deficiency condition to extend the well-known four point algorithm from two views to multiple views. Through our approach, we give a clear explanation of the difficulty in applying generic algorithms to planar scenes - due to a further drop of rank for the matrix $P_j$. Nonetheless, by examining the full structure of the kernel of this matrix, we provide ways to salvage the situation and develop special algorithms for motion and structure from planar (point) features.

According to the analysis of proposed algorithms, for a planar scene, one of the advantages of multiple views over two views is that certain ambiguity can be eliminated. The two physically possible solutions for two views can be further narrowed down to only one by comparing with those from other views. The true solution is in general the only one which will be consistent for all views. Simulation results confirm that the proposed algorithms give good motion and structure estimates despite high noises. In particular, they confirm that the quality of estimates for the 3-D structure indeed improves with more views. We are currently running more simulations as well as experiments on a helicopter landing image sequence in order to fully evaluate the proposed algorithm for practical applications.

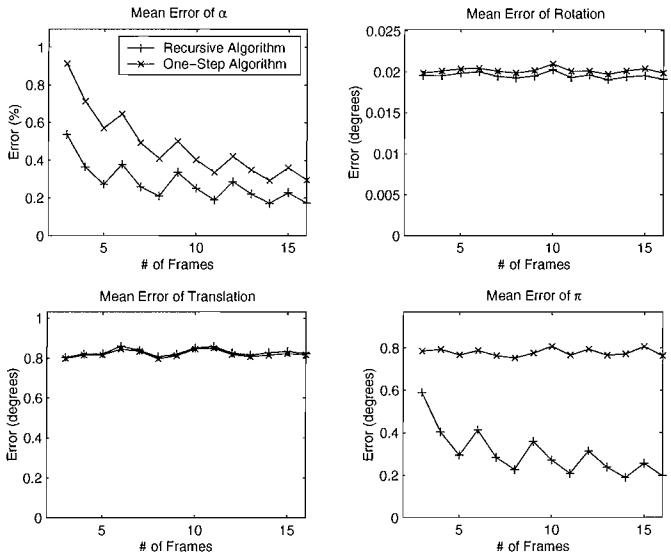


Figure 3.6: Error in motion and structural estimates versus number of image frames for a sequence of repetitive motion $XX$-$YY$-$ZZ$. The noise level is 3 pixels. Motion error is compared at the first motion.

# Chapter 4

# The Universal Multiple View Matrix

*Yi Ma, Jana Košecká, Kun Huang*
*Submitted to ACCV02, June 18th.*

## Abstract

*Geometric relationships governing projections of points and lines in multiple views and the associated algorithms have been studied to a large extent separately in multiple view geometry. In this chapter we present a universal rank deficiency condition of the so-called multiple view matrix $M$ comprised of arbitrary combinations of point and line features across multiple views. This condition can be utilized towards matching points and lines across multiple views and motion and structure recovery from multiple views. The work presented here is an extension of recently developed rank deficiency conditions for point features and line features, which have been shown equivalent to all multi-linear constraints and generalize the previously known relationships between points and lines in three views, to multiple views. The proposed formulation allows us to carry out meaningful geometric analysis for multiple views of corresponding point and line features and systematically characterize all degenerate cases. One motivation behind this formulation is to utilize the constraints governing both point and line features towards a consistent recovery of motion and structure from multiple views. Simulation results are presented to validate the multiple view matrix based approach.*

**Key words:** The multiple view matrix, rank deficiency condition, mixed point and line features, structure from motion, degeneracy.

## 4.1 Introduction

Characterization of the existing geometric constraints has a long history both in computer vision and photogrammetry and has important implications for a variety of applications. The geometric relationships governing observable feature primitives in multiple views provide a starting point from which one can determine the choice of primitives to represent a 3-D scene and consequently formulate and solve the problem of motion and structure recovery from multiple views.

The basic formulation of the geometric constraints governing projections of point features in two views originated in photogrammetry which could be traced back to the beginning of last century [14] and then was revived later in the computer vision community in early eighties [16]. Natural extensions (of theoretical importance and with profound practical implications) had been those considering multiple views and different feature primitives. In the computer vision literature, fundamental and structure independent relationships between image features and camera displacements were first described by the so-called multi-linear matching constraints [6, 27, 10]. Most of the previous work focused on the algebraic aspects of these multi-linear constraints, along with the algorithms which followed from the same formulation. This line of work culminated recently in publication of two monographs on this topic [8, 4].

The constraints among multiple views and associated algorithms were mostly developed separately for point and line features and for different number of views. Such a development relied on the use of (often difficult to follow) tensorial notation, which was preceded by algebraic elimination of some of the unknowns to render otherwise intrinsically nonlinear relationships as linear ones. A distinguished role in this development was the use of the so-called trilinear constraints and their associated trilinear tensors. Trilinear constraints revealed certain geometric relationships between point and line features among three views [23, 20, 7] and were used extensively for feature matching, point-line transfer to a new view, and motion and structure recovery among three views. In order to apply the trilinear constraints to more than three views, however, one typically resorted to certain cascading scheme [1]. Given that the choice of cascading is by no means unique and many degenerate configurations may occur among the chosen triple-wise views, it was difficult to draw any consistent conclusions on the global geometry for the multiple views altogether.

An associated line of work considered formulations of the multiple view geometry for the affine projection case, which rendered the basic relationships among 3-D feature primitives as linear ones [12, 18]. Before that, several multiple view motion and structure recovery techniques exploited the rank deficiency properties of certain multiple view measurement matrix for orthographic projection and resulted in factorization based algorithms for structure and motion recovery [26]. The study of the underlying rank deficiency constraints was also extended to an uncalibrated setting and perspective projection case for point features [28].

The main contribution of our work here is the derivation of a new general rank deficiency condition on a formal multiple view matrix $M$, which combines measurements from multiple views of point and line features. This condition generalizes all previous rank deficiency conditions developed separately for point and line features and it certainly completes previous efforts to use both line and point features for structure from motion [15, 7, 23]. The rank deficiency condition of the newly introduced multiple view matrix $M$ clearly reveals the relationship among all previously known or unknown multi-linear constraints. Furthermore, the matrix $M$ generalizes previously studied trilinear constraints involving mixed point and line features to a multiple view setting hence allows a geometrically meaningful global analysis of arbitrarily many views with arbitrarily mixed features, with no need to cascade pair-wise, triple-wise or quadruple-wise views. Its linear structure directly facilitates feature matching, feature transfer across multiple views and motion and structure recovery. The presented formulation also enables a clear geometric analysis and characterization of all degenerate cases in the multiple view setting, which was intractable using trilinear constraint based methods. An additional appeal of this approach is the sole use of linear algebraic techniques, with no need to introduce tensorial notation.

**How this chapter is organized:** The starting point of the development is the extension of multi-linear constraints between points and lines to mixed point and line features and multiple views. We will first demonstrate in Section 4.3 two different derivations of the constraints on mixed point and line features. In Section 4.4, we present a universal rank condition of a unified multiple view matrix $M$ from which all the known cases can be instantiated. The geometric interpretation of the rank condition of the matrix $M$ given in Section 4.5 clearly reveals the degeneracy of certain configurations of features and motions. In Section 4.6, we outline ideas how to use the multiple view matrix of mixed features to develop algorithms for the recovery of structure and motion from multiple views. Simulation results in Section 4.7 will demonstrate the benefits of the proposed approach. Section 4.8 concludes the chapter.

## 4.2 Multiple views of a point on a line

An image $\mathbf{x}(t) = [x(t), y(t), 1]^T \in \mathbb{R}^3$ of a point $p \in \mathbb{E}^3$, with coordinates $\mathbf{X} = [X, Y, Z, 1]^T \in \mathbb{R}^4$ relative to a fixed world coordinate frame, taken by a moving camera satisfies the following relationship:

$$\lambda(t)\mathbf{x}(t) = A(t)Pg(t)\mathbf{X} \tag{4.1}$$

where $\lambda(t) \in \mathbb{R}_+$ is the (unknown) depth of the point $p$ relative to the camera frame, $A(t) \in SL(3)$ is the camera calibration matrix (at time $t$), $P = [I, 0] \in \mathbb{R}^{3\times 4}$ is the constant projection matrix and $g(t) \in SE(3)$ is the coordinate transformation from the world frame to the camera frame at time $t$. In the above equation, all $\mathbf{x}, \mathbf{X}$ and $g$ are in *homogeneous representation*. Now suppose that $p$ is lying on a straight line $L \subset \mathbb{E}^3$, defined by $L = \{\mathbf{X} \mid \mathbf{X} = \mathbf{X}_0 + \lambda v\}$, where $\mathbf{X}_0 = [X_0, Y_0, Z_0, 1]^T \in \mathbb{R}^4$ is a base point on this line,

$v = [v_1, v_2, v_3, 0]^T \in \mathbb{R}^4$ is a non-zero vector indicating the direction of the line, and $\lambda \in \mathbb{R}$. An image $\mathbf{l}(t) = [a(t), b(t), c(t)]^T \in \mathbb{R}^3$ of $L$ taken by the moving camera then satisfies the following equation:

$$\mathbf{l}(t)^T \mathbf{x}(t) = \mathbf{l}(t)^T A(t) P g(t) \mathbf{X} = 0 \tag{4.2}$$

for the image $\mathbf{x}(t)$ of any point on the line $L$. In a realistic situation, we usually only obtain "sampled" images of $\mathbf{x}(t)$ or $\mathbf{l}(t)$ at some time instances: $t_1, t_2, \ldots, t_m$. For simplicity we denote

$$\lambda_i = \lambda(t_i), \quad \mathbf{x}_i = \mathbf{x}(t_i), \quad \mathbf{l}_i = \mathbf{l}(t_i), \quad \Pi_i = A(t_i) P g(t_i). \tag{4.3}$$

The matrix $\Pi_i$ is then a $3 \times 4$ matrix which relates the $i^{th}$ image of the point $p$ to its world coordinates $\mathbf{X}$ by:

$$\mathbf{x}_i \lambda_i = \Pi_i \mathbf{X} \tag{4.4}$$

as well as relates the $i^{th}$ image of the line $L$ to its world coordinates $(\mathbf{X}_0, v)$ by:

$$\mathbf{l}_i^T \Pi_i \mathbf{X}_0 = \mathbf{l}_i^T \Pi_i v = 0 \tag{4.5}$$

for $i = 1, \ldots, m$. Since the image points are on the image lines, we further have relationship:

$$\mathbf{l}_i^T \mathbf{x}_i = 0 \tag{4.6}$$

for $i = 1, \ldots, m$.

It has been observed previously that the basic relationship which holds between the image measurements and motions of the camera after eliminating the unknowns $\lambda_i$'s and $\mathbf{X}$ is that the $m + 4$ column vectors of the following matrix:

$$N_p := [\Pi,\ \mathcal{I}] = \begin{bmatrix} \Pi_1 & \mathbf{x}_1 & 0 & \cdots & 0 \\ \Pi_2 & 0 & \mathbf{x}_2 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ \Pi_m & 0 & \cdots & 0 & \mathbf{x}_m \end{bmatrix} \in \mathbb{R}^{3m \times (m+4)} \tag{4.7}$$

are *linearly dependent* or:

$$\boxed{\text{rank}(N_p) \leq m + 3}$$

since it is clear that the vector $u := [\mathbf{X}^T, -\lambda_1, \ldots, -\lambda_m,]^T \in \mathbb{R}^{m+4}$ is in the null space of the matrix $N_p$ due to (1.2). For the line feature, correspondingly, the following matrix:

$$N_l = \begin{bmatrix} \mathbf{l}_1^T \Pi_1 \\ \mathbf{l}_2^T \Pi_2 \\ \vdots \\ \mathbf{l}_m^T \Pi_m \end{bmatrix} \in \mathbb{R}^{m \times 4} \tag{4.8}$$

satisfies:

$$\boxed{\text{rank}(N_l) \leq 2} \tag{4.9}$$

since it is clear that the vectors $\mathbf{X}_0$ and $v$ are in the null space of the matrix $N_l$ due to (2.4).

## 4.3 Special multiple view matrices for mixed features

The above observations of the rank deficiency of the matrices associated with points and lines in multiple views are cannot be exploited algorithmically in the above form and further simplifications are due. In this section we will study two special cases and show how to simplify the rank deficiency condition on matrices $N_p$ and $N_l$ to obtain constraints on mixed point and line features.

### 4.3.1 The multiple view matrix for the point-line-line case

**Point-line-line constraints from $N_p$**

Without loss of generality, we may assume that the first camera frame is chosen to be the reference frame.[1] That gives the projection matrices $\Pi_i, i = 1, \ldots, m$ the general form:

$$\Pi_1 = [I, 0], \quad \ldots, \quad \Pi_m = [R_m, T_m] \quad \in \mathbb{R}^{3\times 4}, \tag{4.10}$$

where $R_i \in \mathbb{R}^{3\times 3}, i = 2, \ldots, m$ is the first three columns of $\Pi_i$ and $T_i \in \mathbb{R}^3, i = 2, \ldots, m$ is the fourth column of $\Pi_i$. Although we have used the suggestive notation $(R_i, T_i)$ here, they are not necessarily the actual rotation and translation. $R_i$ could be an arbitrary $3 \times 3$ matrix. Only in the case when the camera is perfectly calibrated does $R_i$ correspond to actual camera rotation and $T_i$ to translation. The rank deficiency of the matrix $N_p$ is after some simplification equivalent to the rank deficiency of the matrix:

$$N_p' = \begin{bmatrix} T_2 & R_2\mathbf{x}_1 & \mathbf{x}_2 & 0 & \cdots & 0 \\ T_3 & R_3\mathbf{x}_1 & 0 & \mathbf{x}_3 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & 0 \\ T_m & R_m\mathbf{x}_1 & 0 & \cdots & 0 & \mathbf{x}_m \end{bmatrix} \in \mathbb{R}^{3(m-1)\times(m+1)} \tag{4.11}$$

or more precisely $\text{rank}(N_p) = 3 + \text{rank}(N_p')$. Multiplying the following matrix:

$$D_l = \begin{bmatrix} \widehat{\mathbf{l}_2} & 0 & 0 & \cdots & 0 \\ \mathbf{l}_2^T & 0 & 0 & \cdots & 0 \\ 0 & \widehat{\mathbf{l}_3} & 0 & \cdots & 0 \\ 0 & \mathbf{l}_3^T & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 0 & \widehat{\mathbf{l}_m} \\ 0 & \cdots & 0 & 0 & \mathbf{l}_m^T \end{bmatrix} \in \mathbb{R}^{4(m-1)\times 3(m-1)} \tag{4.12}$$

on the left of $N_p'$ yields:

$$D_l N_p' = \begin{bmatrix} \widehat{\mathbf{l}_2} T_2 & \widehat{\mathbf{l}_2} R_2\mathbf{x}_1 & \widehat{\mathbf{l}_2}\mathbf{x}_2 & 0 & 0 & 0 \\ \mathbf{l}_2{}^T T_2 & \mathbf{l}_2{}^T R_2\mathbf{x}_1 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & 0 & \ddots & 0 & 0 \\ \vdots & \vdots & 0 & 0 & \ddots & 0 \\ \widehat{\mathbf{l}_m} T_m & \widehat{\mathbf{l}_m} R_m\mathbf{x}_1 & 0 & 0 & 0 & \widehat{\mathbf{l}_m}\mathbf{x}_m \\ \mathbf{l}_m^T T_m & \mathbf{l}_m^T R_m\mathbf{x}_1 & 0 & 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{4(m-1)\times(m+1)}. \tag{4.13}$$

Since $D_l$ is of full rank $3(m-1)$ we have:

$$\text{rank}(N_p') = \text{rank}(D_l N_p').$$

Hence the original matrix $N_p$ is rank deficient only if the following sub-matrix of $D_l N_p'$:

$$M_{pl} := \begin{bmatrix} \mathbf{l}_2^T T_2 & \mathbf{l}_2^T R_2\mathbf{x}_1 \\ \mathbf{l}_3^T T_3 & \mathbf{l}_3^T R_3\mathbf{x}_1 \\ \vdots & \vdots \\ \mathbf{l}_m^T T_m & \mathbf{l}_m^T R_m\mathbf{x}_1 \end{bmatrix} \in \mathbb{R}^{(m-1)\times 2} \tag{4.14}$$

[1] Depending on the context, the reference frame could be either a Euclidean, affine or projective reference frame. In any case, the projection matrix for the first image becomes the standard projection matrix $[I, 0] \in \mathbb{R}^{3\times 4}$. The reader should note that we do not lose any generality for doing this.

is rank deficient. We call the matrix $M_{pl}$ the *multiple view matrix* for the *point-line-line* case. More precisely, we have:

$$\boxed{\text{rank}(M_{pl}) \leq \text{rank}(N_p) - (m+2) \leq 1.} \tag{4.15}$$

**Point-line-line constraints from $N_l$**

Similarly we can start with the matrix $N_l$ which captures all the constraints of on the images of a 3-D line in multiple views. With the choice of the first frame as the reference frame, we have:

$$N_l = \begin{bmatrix} \mathbf{l}_1^T & 0 \\ \mathbf{l}_2^T R_2 & \mathbf{l}_2^T T_2 \\ \vdots & \vdots \\ \mathbf{l}_m^T R_m & \mathbf{l}_m^T T_m \end{bmatrix} \in \mathbb{R}^{m \times 4}. \tag{4.16}$$

This matrix should have a rank of no more than 2. Multiplying the following matrix[2]

$$D_p = \begin{bmatrix} \mathbf{x}_1 & \widehat{\mathbf{x}_1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{4 \times 5} \tag{4.17}$$

to the right of $N_l$ yields:

$$N_l D_p = \begin{bmatrix} 0 & \mathbf{l}_1^T \widehat{\mathbf{x}_1} & 0 \\ \mathbf{l}_2^T R_2 \mathbf{x}_1 & \mathbf{l}_2^T R_2 \widehat{\mathbf{x}_1} & \mathbf{l}_2^T T_2 \\ \vdots & \vdots & \vdots \\ \mathbf{l}_m^T R_m \mathbf{x}_1 & \mathbf{l}_m^T R_m \widehat{\mathbf{x}_1} & \mathbf{l}_m^T T_m \end{bmatrix} \in \mathbb{R}^{m \times 5}. \tag{4.18}$$

Since $D_p$ is of full rank 4, hence

$$\text{rank}(N_l D_p) = \text{rank}(N_l) \leq 2$$

And this is true if and only if the following sub-matrix of $N_l D_p$:

$$M_{pl} = \begin{bmatrix} \mathbf{l}_2^T R_2 \mathbf{x}_1 & \mathbf{l}_2^T T_2 \\ \vdots & \vdots \\ \mathbf{l}_m^T R_m \mathbf{x}_1 & \mathbf{l}_m^T T_m \end{bmatrix} \in \mathbb{R}^{(m-1) \times 2} \tag{4.19}$$

satisfies $\text{rank}(M_{pl}) \leq 1$.[3] More precisely, we have:

$$\boxed{\text{rank}(M_{pl}) = \text{rank}(N_l) - 1 \leq 1.} \tag{4.20}$$

Matrix $M_{pl}$ captures all the constraints between a feature point $\mathbf{x}_1$ in the reference few and corresponding lines $\mathbf{l}_i$, which the point is incident to in all other views. Equations (4.15) and (4.20) relate the rank of *point-line-line* mixed feature matrix $M_{pl}$ to the rank of $N_l$ and $N_p$ respectively. It is worth noting that for the rank deficiency condition to be true it is necessary for all $2 \times 2$ minors of $M_{pl}$ to be zero, *i.e.*, the following constraints hold among arbitrary triplets of local images:

$$[\mathbf{l}_i^T R_i \mathbf{x}_1][\mathbf{l}_j^T T_j] - [\mathbf{l}_i^T T_i][\mathbf{l}_j^T R_j \mathbf{x}_1] = 0 \quad \in \mathbb{R}, \quad i, j = 2, \ldots, m. \tag{4.21}$$

These are exactly the well-known *point-line-line* relationships among three views. Similar trilinear constraints can be also obtained for points only (point-point-point) or lines only (line-line-line) and their alternative derivation, which proceeds in a similar manner, can be found in previous chapters. Before proceeding to the general formulation, we will treat one more special case separately.

[2] For a three dimensional vector $u \in \mathbb{R}^3$, we use $\widehat{u} \in \mathbb{R}^{3 \times 3}$ to denote the skew symmetric matrix associated to $u$ such that for any vector $v \in \mathbb{R}^3$, we have: $\widehat{u}v = u \times v$. Notice that $\widehat{u}$ is skew-symmetric, *i.e.*, $\widehat{u}^T = -\widehat{u}$.

[3] Notice that except for a change of the order of columns, the matrix $M_{pl}$ is the same as the one defined in (4.14).

### 4.3.2 The multiple view matrix for the line-point-point case

Consider the situation when you observe a line in the reference view and a feature point belonging to the line in remaining views. To derive the constraints which this feature set has to satisfy we start again with the matrix:

$$N_p = \begin{bmatrix} I & 0 & \mathbf{x}_1 & 0 & \cdots & 0 \\ R_2 & T_2 & 0 & \mathbf{x}_2 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & 0 \\ R_m & T_m & 0 & \cdots & 0 & \mathbf{x}_m \end{bmatrix} \in \mathbb{R}^{3m\times(m+4)}. \tag{4.22}$$

Multiplying on its left the following matrix:

$$D_l' = \begin{bmatrix} \mathbf{l}_1^T & 0 \\ \widehat{\mathbf{l}_1} & 0 \\ 0 & I_{3(m-1)\times 3(m-1)} \end{bmatrix} \in \mathbb{R}^{(3m+1)\times 3m} \tag{4.23}$$

yields:

$$D_l' N_p = \begin{bmatrix} \mathbf{l}_1^T & 0 & 0 & 0 & \cdots & 0 \\ \widehat{\mathbf{l}_1} & 0 & \widehat{\mathbf{l}_1}\mathbf{x}_1 & 0 & \cdots & 0 \\ R_2 & T_2 & 0 & \mathbf{x}_2 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & 0 \\ R_m & T_m & 0 & \cdots & 0 & \mathbf{x}_m \end{bmatrix} \in \mathbb{R}^{(3m+1)\times(m+4)}. \tag{4.24}$$

Since $\text{rank}(D_l') = 3m$, we have $\text{rank}(N_p) = \text{rank}(D_l' N_p) \leq m + 3$. Now multiply on the left of $D_l' N_p$ by the the following matrix:

$$D_p' = \begin{bmatrix} I_{4\times 4} & 0 & 0 & \cdots & 0 \\ 0 & \widehat{\mathbf{x}_2} & 0 & \cdots & 0 \\ 0 & \mathbf{x}_2^T & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 0 & \widehat{\mathbf{x}_m} \\ 0 & \cdots & 0 & 0 & \mathbf{x}_m^T \end{bmatrix} \in \mathbb{R}^{(4m+1)\times(3m+1)}. \tag{4.25}$$

It is direct to verify that the rank of the resulting matrix $D_p' D_l' N_p$ is related to its sub-matrix:

$$N_p'' = \begin{bmatrix} \mathbf{l}_1^T & 0 \\ \widehat{\mathbf{x}_2} R_2 & \widehat{\mathbf{x}_2} T_2 \\ \vdots & \vdots \\ \widehat{\mathbf{x}_m} R_m & \widehat{\mathbf{x}_m} T_m \end{bmatrix} \in \mathbb{R}^{(3m-2)\times 4} \tag{4.26}$$

as $\text{rank}(N_p'') + m \leq \text{rank}(D_p' D_l' N_p) = \text{rank}(N_p)$. Now multiplying:

$$\begin{bmatrix} \mathbf{l}_1 & \widehat{\mathbf{l}_1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{4\times 5} \tag{4.27}$$

on the right of $N_p''$ yields:

$$\begin{bmatrix} \mathbf{l}_1^T \mathbf{l}_1 & 0 & 0 \\ \widehat{\mathbf{x}_2} R_2 \mathbf{l}_1 & \widehat{\mathbf{x}_2} R_2 \widehat{\mathbf{l}_1} & \widehat{\mathbf{x}_2} T_2 \\ \vdots & \vdots & \vdots \\ \widehat{\mathbf{x}_m} R_m \mathbf{l}_1 & \widehat{\mathbf{x}_m} R_m \widehat{\mathbf{l}_1} & \widehat{\mathbf{x}_m} T_m \end{bmatrix} \in \mathbb{R}^{(3m-2)\times 5}. \tag{4.28}$$

We call its sub-matrix:

$$M_{lp} := \begin{bmatrix} \widehat{\mathbf{x}_2} R_2 \widehat{\mathbf{l}_1} & \widehat{\mathbf{x}_2} T_2 \\ \vdots & \vdots \\ \widehat{\mathbf{x}_m} R_m \widehat{\mathbf{l}_1} & \widehat{\mathbf{x}_m} T_m \end{bmatrix} \in \mathbb{R}^{3(m-1)\times 4} \tag{4.29}$$

the *multiple view matrix* for the *line-point-point* case. Its rank is related to that of $N_p$ as:

$$\boxed{\text{rank}(M_{lp}) \leq \text{rank}(N_p) - (m+1) \leq 2.} \tag{4.30}$$

Now, one could easily foresee that in a similar manner one could derive *point-point-line* constraints. A natural question to ask is what other combination of features will yield multiple view constraints of a similar form. Instead of giving independent derivations on a case by case basis, we next put forward a universal rank condition on a formal multiple view matrix. This formulation will capture all types of constraints among point features, line features as well as their mixture.

## 4.4 Rank condition on the universal multiple view matrix

Without loss of generality, we still choose the first camera frame to be the reference frame. For the $m$ images $\mathbf{x}_1, \ldots, \mathbf{x}_m$ of a point $p$ on a line $L$ with its $m$ images $\mathbf{l}_1, \ldots, \mathbf{l}_m$, we define the following set of formal matrices:

$$\begin{aligned} D_i &:= \widehat{\mathbf{x}_i} \in \mathbb{R}^{3\times 3} \quad \text{or} \quad \mathbf{l}_i^T \in \mathbb{R}^3, \\ D_i^\perp &:= \mathbf{x}_i \in \mathbb{R}^3 \quad \text{or} \quad \widehat{\mathbf{l}_i}^T \in \mathbb{R}^{3\times 3}, \end{aligned}$$

where the transpose on $\widehat{\mathbf{l}_i}^T$ is purely stylistic. Then, depending on whether the available measurement from the $i^{th}$ image is a point feature or a line feature, the $D_i$ matrix chooses a corresponding value. That choice is completely independent of the other $D_i$'s. The "dual" matrix $D_i^\perp$ can be viewed as the *orthogonal supplement* to $D_i$.[4] Using the above definition of $D_i$ and $D_i^\perp$, we now "formally" define a *universal multiple view matrix*:

$$M := \begin{bmatrix} D_2 R_2 D_1^\perp & D_2 T_2 \\ D_3 R_3 D_1^\perp & D_3 T_3 \\ \vdots & \vdots \\ D_m R_m D_1^\perp & D_m T_m \end{bmatrix}. \tag{4.31}$$

Depending on the particular choice for each $D_i$ or $D_1^\perp$, the dimension of the matrix $M$ may vary - the reason why we call the definition "formal". But no matter what the choice for each individual $D_i$ or $D_1^\perp$ is, $M$ will always be a valid matrix of certain dimension. As an extension of existing results on multiple view matrix, one can directly prove the following statement:

**Theorem 14 (Formal rank condition on the multiple view matrix).** *Consider a point $p$ lying on a line $L$ and their images $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathbb{E}^3$ and $\mathbf{l}_1, \ldots, \mathbf{l}_m \in \mathbb{E}^3$ relative to $m$ camera frames whose relative configuration is given by $(R_i, T_i)$ for $i = 2, \ldots, m$. Then for different choices for $D_i$ and $D_1^\perp$ in the definition of the multiple view matrix $M$, the rank of $M$ belongs to the following two cases:*

1. *If $D_1^\perp = \widehat{\mathbf{l}_1}^T$ and $D_i = \widehat{\mathbf{x}_i}$ for some $i \geq 2$, then*

$$\boxed{\textit{rank}(M) \leq 2.} \tag{4.32}$$

[4] In fact, there are many equivalent matrix representations for $D_i$ and $D_i^\perp$. For example, assuming $\|\mathbf{x}_i\| = \|\mathbf{l}_i\| = 1$, the rest of the theory remains exactly the same if we replace in the definition $\widehat{\mathbf{x}_i}$ by $I - \mathbf{x}_i\mathbf{x}_i^T \in \mathbb{R}^{3\times 3}$ and $\widehat{\mathbf{l}_i}^T$ by $I - \mathbf{l}_i\mathbf{l}_i^T \in \mathbb{R}^{3\times 3}$, matrices representing projection to the subspaces orthogonal to $\mathbf{x}_i$ and $\mathbf{l}_i$ respectively. It is easy to see this because $\widehat{u}^T\widehat{u} = I - uu^T$ for any unit vector $u \in \mathbb{R}^3$. We choose $\widehat{\mathbf{x}_i}$ and $\widehat{\mathbf{l}_i}^T$ here because they are the simplest forms representing the orthogonal subspaces of $\mathbf{x}_i$ and $\mathbf{l}_i$ and also linear in $\mathbf{x}_i$ and $\mathbf{l}_i$ respectively. So the "duality" between $D_i$ and $D_i^\perp$ is indeed the duality between any subspace $S \subset \mathbb{R}^3$ and its orthogonal supplement subspace $S^\perp \subset \mathbb{R}^3$.

*2. Otherwise,*

$$\boxed{rank(M) \leq 1.} \tag{4.33}$$

A complete proof of all cases implied by this theorem simply relies on different basic matrix manipulations of the matrix $N_p$, similar to what we have shown for $M_{pl}$ and $M_{lp}$ in the above, or for $M_p$ and $M_l$ in two previous chapters. Notice that the above theorem gives a universal description of the incidence condition on any combination of line or point features on the $m$ images: The rank of $M$ should always be no more than 1 except for one special case. To see why the overall $M$ can be rank 2 in the special case 1, notice that in general each sub-matrix of the type:

$$\begin{bmatrix} \widehat{\mathbf{x}}_i R_i \widehat{\mathbf{l}}_1^T & \widehat{\mathbf{x}}_i T_i \end{bmatrix} \in \mathbb{R}^{3\times 4} \tag{4.34}$$

would have rank 2 already. But its geometric reason is yet to be revealed in the next section.

**Remark 5 (Coplanar features).** *In the case that all features are from a plane given by $\pi = [\pi^1, \pi^2] \in \mathbb{R}^4$ with $\pi^1 \in \mathbb{R}^3, \pi^2 \in \mathbb{R}$, simply append*

$$\begin{bmatrix} \pi^1 D_1^\perp & \pi^2 \end{bmatrix} \tag{4.35}$$

*to the matrix $M$ in its formal definition (4.31). The rank condition on $M$ remains the same. The rank condition on the new $M$ matrix then implies* all *constraints among multiple images of planar features, including the homographic constraints studied in [5] (see Chapter 3 for more details).*

**Remark 6 (Features at infinity).** *In Theorem 1, if the point $p$ and line $L$ are in the plane at infinity $\mathbb{P}^3 \setminus \mathbb{E}^3$, the rank condition on the multiple view matrix $M$ is just the same. Hence the rank condition extends to multiple view geometry of the entire projective space $\mathbb{P}^3$, and it does not discriminate Euclidean, affine or projective assumption on the underlying space.*

**Remark 7 (Occlusion).** *If any feature is occluded in a particular image, the corresponding row (or a group of rows) is simply omitted from $M$; or if only the point is occluded but not the entire line(s) on which the point lies, then simply replace the missing image of the point by the corresponding image(s) of the line(s). In either case, the overall rank condition on $M$ remains unaffected. In fact, the rank condition on $M$ gives a very effective criterion to tell whether or not a set of (mixed) features indeed correspond to the same 3-D point on the same 3-D line. If the features are miss-matched, either due to occlusion or errors in establishing correspondence, the rank condition will be violated.*

As a result of Theorem 1, any previously known or unknown relationships among multiple images of point or line features are simply certain instantiations of the Theorem 1. Whenever an instance of the multiple view matrix $M$ is chosen, we can then spell out a specific set of constraints that the rank of $M$ imposes locally on pair-wise or triple-wise images. In general, there are four basic types of constraints: *point-point-point, line-line-line, point-line-line*, and *line-point-point.* It is worth noting that the rank description is far more general and universal than these special constraints, since restricting the constraints to triple-wise views may introduce certain artificial degeneracies.[5] In any case, there will be no further relationship among any four views, even in the mixed feature scenario.[6] For clarity, we demonstrate using the following examples how to obtain these different types of constraints by instantiating $M$:

**Example 1 (Point-point-point constraints).** *Let us choose $D_1^\perp = \mathbf{x}_1, D_2 = \widehat{\mathbf{x}_2}, D_3 = \widehat{\mathbf{x}_3}$. Then we get a multiple view matrix:*

$$M = \begin{bmatrix} \widehat{\mathbf{x}_2} R_2 \mathbf{x}_1 & \widehat{\mathbf{x}_2} T_2 \\ \widehat{\mathbf{x}_3} R_3 \mathbf{x}_1 & \widehat{\mathbf{x}_3} T_3 \end{bmatrix} \in \mathbb{R}^{6\times 2}. \tag{4.36}$$

[5] For example, some three views may form a degenerate configuration among themselves but no longer after putting them together with many other views.

[6] In fact, this is quite expected: While the rank condition geometrically corresponds to the incidence condition that lines intersect at a point and that planes intersect at a line, incidence condition that three-dimensional subspaces intersect at a plane is a void condition in $\mathbb{E}^3$.

*Then rank$(M) \leq 1$ gives:*

$$[\widehat{\mathbf{x}_2} R_2 \mathbf{x}_1][\widehat{\mathbf{x}_3} T_3]^T - [\widehat{\mathbf{x}_3} R_3 \mathbf{x}_1][\widehat{\mathbf{x}_2} T_2]^T = 0 \quad \in \mathbb{R}^{3\times 3}. \tag{4.37}$$

*The above equation give the point-point-point type of constraints on three images.*

**Example 2 (Line-line-line constraints).** *Let us choose $D_1^\perp = \widehat{\mathbf{l}_1}^T, D_2 = \mathbf{l}_2^T, D_3 = \mathbf{l}_3^T$. Then we get a multiple view matrix:*

$$M = \begin{bmatrix} \mathbf{l}_2^T R_2 \widehat{\mathbf{l}_1}^T & \mathbf{l}_2^T T_2 \\ \mathbf{l}_3^T R_3 \widehat{\mathbf{l}_1}^T & \mathbf{l}_3^T T_3 \end{bmatrix} \in \mathbb{R}^{2\times 4}. \tag{4.38}$$

*Then rank$(M) \leq 1$ gives:*

$$\left[\mathbf{l}_2^T R_2 \widehat{\mathbf{l}_1}^T\right][\mathbf{l}_3^T T_3] - \left[\mathbf{l}_3^T R_3 \widehat{\mathbf{l}_1}^T\right][\mathbf{l}_2^T T_2] = 0 \quad \in \mathbb{R}^3. \tag{4.39}$$

*The above equation gives the line-line-line type of constraints on three images.*

**Example 3 (Point-line-line constraints).** *Let us choose $D_1^\perp = \mathbf{x}_1, D_2 = \mathbf{l}_2^T, D_3 = \mathbf{l}_3^T$. Then we get a multiple view matrix:*

$$M = \begin{bmatrix} \mathbf{l}_2^T R_2 \mathbf{x}_1 & \mathbf{l}_2^T T_2 \\ \mathbf{l}_3^T R_3 \mathbf{x}_1 & \mathbf{l}_3^T T_3 \end{bmatrix} \in \mathbb{R}^{2\times 2}. \tag{4.40}$$

*Then rank$(M) \leq 1$ gives*

$$[\mathbf{l}_2^T R_2 \mathbf{x}_1][\mathbf{l}_3^T T_3] - [\mathbf{l}_3^T R_3 \mathbf{x}_1][\mathbf{l}_2^T T_2] = 0 \quad \in \mathbb{R}. \tag{4.41}$$

*The above equation gives the point-line-line type of constraints on three images.*

**Example 4 (Line-point-point constraints).** *Let us choose $D_1^\perp = \widehat{\mathbf{l}_1}^T, D_2 = \widehat{\mathbf{x}_2}, D_3 = \widehat{\mathbf{x}_3}$. Then we get a multiple view matrix:*

$$M = \begin{bmatrix} \widehat{\mathbf{x}_2} R_2 \widehat{\mathbf{l}_1}^T & \widehat{\mathbf{x}_2} T_2 \\ \widehat{\mathbf{x}_3} R_3 \widehat{\mathbf{l}_1}^T & \widehat{\mathbf{x}_3} T_3 \end{bmatrix} \in \mathbb{R}^{6\times 4}. \tag{4.42}$$

*Then rank$(M) \leq 2$ implies that all $3 \times 3$ sub-matrices of $M$ have determinant zero. These equations give the line-point-point type of constraints on three images.*

Similarly, other choices in $D_i$ and $D_1^\perp$ will give all possible types of constraints among any views of point and line features arbitrarily mixed.

## 4.5 Geometric interpretation of the multiple view matrix

Since there are practically infinitely many possible combinations of point and line features for arbitrarily many views, it is impossible to provide a geometric description to each of them. Instead, we are going to discuss a few essential cases which will give the reader a clear idea how the rank condition works geometrically. Understanding these cases would be sufficient for the reader to carry out a similar analysis to any other case.

Lets first consider the more general case, *i.e.*, the case 2 in Theorem 14 when rank$(M) \leq 1$ and we will discuss the case 1 afterwards. For the case 2, there are only two interesting sub-cases depending on the value of the rank of $M$:

$$1.\ \text{rank}(M) = 1, \quad \text{and} \quad 2.\ \text{rank}(M) = 0. \tag{4.43}$$

When the rank of $M$ is 1, it corresponds to the generic cases that, regardless of the particular choice of features in $M$, all these features satisfy the incidence condition, *i.e.*, all the point features (if more than 2

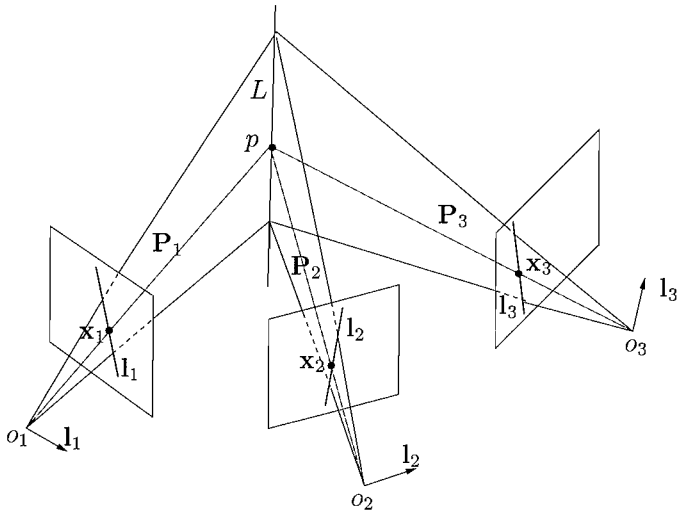


Figure 4.1: Generic configuration for the case rank$(M) = 1$. Planes extended from the images $\mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3$ intersect at one line $L$ in 3-D. Lines extended from the images $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ intersect at one point $p$. $p$ must lie on $L$.

present in $M$) are from a unique 3-D point $p$, lines features (if more than 3 present in $M$) are from a unique 3-D line $L$, and if both point and line features are present, the point $p$ then must lie on the line $L$ in 3-D. This is illustrated in Figure 4.1. But what happens if there are not enough point or line features present in $M$? For example, in $M_{pl}$, there is only one point feature $\mathbf{x}_1$ present. Still rank$(M_{pl}) = 1$ means that a line $L$ is uniquely determined by $\mathbf{l}_2, \ldots, \mathbf{l}_m$ and the point $p$ is consequently determined by the $L$ and its first image $\mathbf{x}_1$. On the other hand, if there is only one line features present in some $M$ but more than two point features in $M$, $L$ can then be a family of lines (on a plane in fact) passing through the point $p$ determined by the rest of point features in $M$.

When the rank of $M$ is 0, it means all the entries of $M$ are zeros. It is easy to verify that this corresponds to a set of degenerate cases when the 3-D location of the point or the line cannot be uniquely determined from their multiple images (no matter how many), and the incidence condition between the point $p$ and the line $L$ no longer holds. In these cases, the best we can do is: 1. When there are more than two point features present in $M$, the 3-D location of the point $p$ can be determined up to a line which connects all camera centers (related to these point features); 2. When there are more than three line features are present in $M$, the 3-D location of the line $L$ can be determined up to the plane on which all related camera centers must lie; 3. When both point and line features are present in $M$, we can usually determine the point $p$ up to a line (connecting all camera centers related to the point features) which is lying on the same plane on which the rest of the camera centers (related to the line features) and the line $L$ must lie. Let us demonstrate this on a concrete example. Suppose the number of views is $m = 6$ and we choose the matrix $M$ to be:

$$M = \begin{bmatrix} \mathbf{l}_2^T R_2 \mathbf{x}_1 & \mathbf{l}_2^T T_2 \\ \mathbf{l}_3^T R_3 \mathbf{x}_1 & \mathbf{l}_3^T T_3 \\ \mathbf{l}_4^T R_4 \mathbf{x}_1 & \mathbf{l}_4^T T_4 \\ \widehat{\mathbf{x}_5} R_5 \mathbf{x}_1 & \widehat{\mathbf{x}_5} T_5 \\ \widehat{\mathbf{x}_6} R_6 \mathbf{x}_1 & \widehat{\mathbf{x}_6} T_6 \end{bmatrix} \in \mathbb{R}^{9 \times 2}. \tag{4.44}$$

The geometric configuration of the point and line features corresponding to the condition rank$(M) = 0$ is illustrated in Figure 4.2. But notice that, among all the possible solutions for $L$ and $p$, if they both happen to be at infinity, the incidence condition then would hold for all the images involved.

We now discuss the case 1 in Theorem 1 when rank$(M) \leq 2$. In this case, the matrix $M$ must contains at least one sub-matrix of the type:

$$\begin{bmatrix} \widehat{\mathbf{x}}_i R_i \widehat{\mathbf{l}_1}^T & \widehat{\mathbf{x}}_i T_i \end{bmatrix} \in \mathbb{R}^{3 \times 4} \tag{4.45}$$

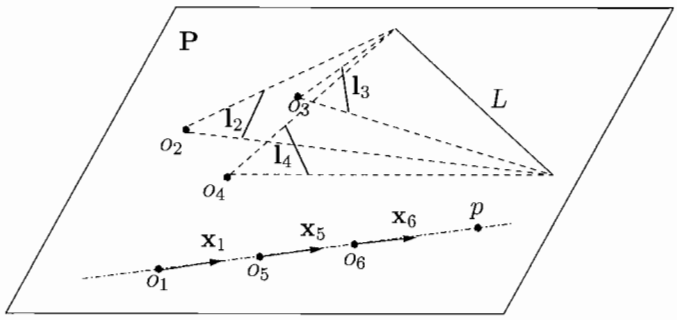


Figure 4.2: A degenerate geometric configuration for the case $\text{rank}(M) = 0$: a point-line-line-line-point-point scenario. From the given rank condition, the line $L$ could be any where on the plane spanned by all the camera centers; the point $p$ could be any where on the line through $o_1, o_5, o_6$.

for some $i \geq 2$. It is easy to verify that such a module can never be zero hence the only sub-cases for the possible rank of $M$ are:

$$1.\ \text{rank}(M) = 2, \quad \text{and} \quad 2.\ \text{rank}(M) = 1. \tag{4.46}$$

When the rank of $M$ is 2, it corresponds to the generic cases that incidence condition among the features is effective. The essential example here is the matrix $M_{lp}$ given in (4.29)

$$M_{lp} = \begin{bmatrix} \widehat{\mathbf{x}_2} R_2 \widehat{\mathbf{l}_1} & \widehat{\mathbf{x}_2} T_2 \\ \vdots & \vdots \\ \widehat{\mathbf{x}_m} R_m \widehat{\mathbf{l}_1} & \widehat{\mathbf{x}_m} T_m \end{bmatrix} \in \mathbb{R}^{3(m-1)\times 4}. \tag{4.47}$$

If $\text{rank}(M_{lp}) = 2$, it can be shown that the point $p$ is only determined up to the plane specified by $o_1$ and $\mathbf{l}_1$ but all the point features $\mathbf{x}_2, \ldots, \mathbf{x}_m$ correspond to the same point $p$. The line $L$ is only determined up to this plane, but the point $p$ does not have to be on this line. This is illustrated in Figure 4.3. Beyond

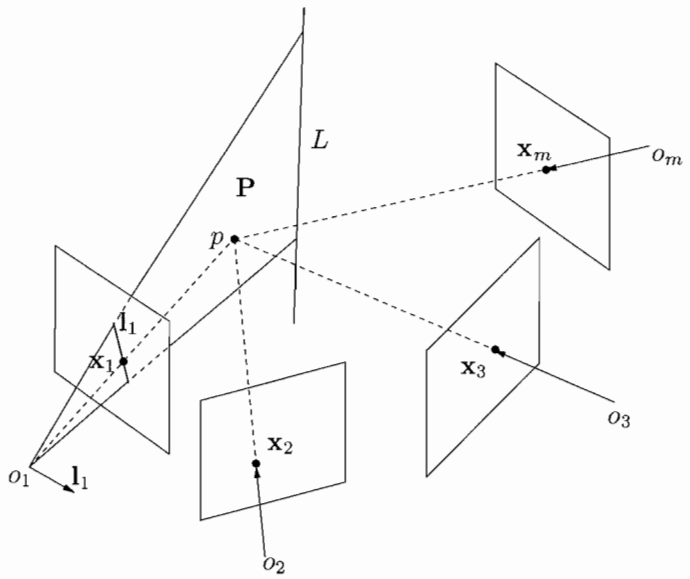


Figure 4.3: Generic configuration for the case $\text{rank}(M_{lp}) = 2$.

$M_{lp}$, if there are more than two line features present in some $M$, the point $p$ then must lie on every plane

associated to every line feature. Hence $p$ must be on the intersection of these planes. Note that even in this case, adding more rows of line features to the $M$ matrix will not be able to uniquely determine the line $L$ in 3-D. Because the incidence condition for multiple line features requires the associated rank to be 1.[7] If we only require rank 2 for the overall matrix $M$, the line can be determined only up to a family of lines - intersections of the planes associated to all the line features - which all should intersect at the same point $p$.

When the rank of $M$ is 1, it corresponds to a set of degenerate cases that the incidence relationship between the point $p$ and the line $L$ will be violated. For example, it is direct to show that $M_{lp}$ is of rank 1 if and only if all the vectors $R_i^{-1}\mathbf{x}_i, i = 2, \ldots, m$ are parallel to each other and they are all orthogonal to $\mathbf{l}_1$, and so are $R_i^{-1}T_i, i = 2, \ldots, m$ orthogonal to $\mathbf{l}_1$. That means all the camera centers lie on the same plane specified by $o_1$ and $\mathbf{l}_1$ and all the images $\mathbf{x}_2, \ldots, \mathbf{x}_m$ (transformed to the reference camera frame) lie on the same plane and are parallel to each other. For example, suppose $m = 5$ and we choose $M$ to be:

$$M := \begin{bmatrix} \widehat{\mathbf{x}_2} R_2 \widehat{\mathbf{l}_1}^T & \widehat{\mathbf{x}_2} T_2 \\ \widehat{\mathbf{x}_3} R_3 \widehat{\mathbf{l}_1}^T & \widehat{\mathbf{x}_3} T_3 \\ \widehat{\mathbf{x}_4} R_4 \widehat{\mathbf{l}_1}^T & \widehat{\mathbf{x}_4} T_4 \\ \mathbf{l}_5^T R_5 \widehat{\mathbf{l}_1}^T & \mathbf{l}_5^T T_5 \end{bmatrix} \in \mathbb{R}^{10\times 4}. \tag{4.48}$$

The geometric configuration of the point and line features corresponding to the condition $\text{rank}(M) = 1$ is illustrated in Figure 4.4. Notice that in this case, we no longer have incidence condition for the point

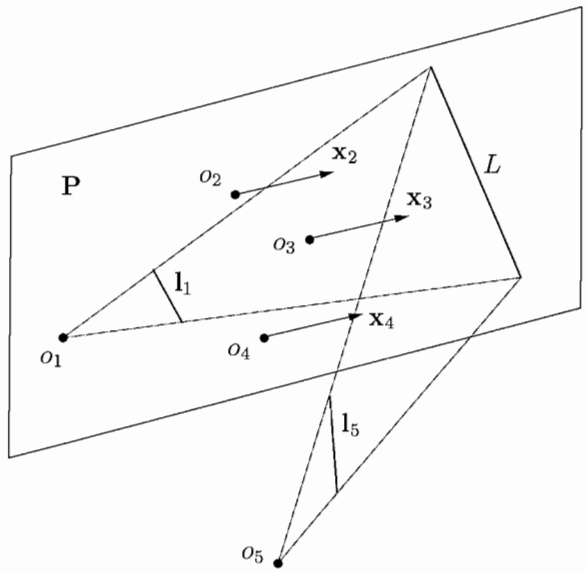


Figure 4.4: A degenerate geometric configuration for the case $\text{rank}(M) = 1$: a line-point-point-point-line scenario.

features. However, one can view them as they intersect at a point $p$ at infinity. In general, we no longer have the incidence condition between the point $p$ and the line $L$, unless both the point $p$ and line $L$ are in the plane at infinity in the first place. But since the rank condition is effective for line features, the incidence condition for all the line features still holds.

As a summary of the above discussion, we see that the rank condition indeed allows us to carry out meaningful geometric analysis on the relationship among multiple point and line features for arbitrarily many views. It without doubt extends existing methods which are based multi-linear tensors and are only good for analyzing up to three views at a time. Since there is yet no systematic way to extend triple-wise analysis to multiple views based on such as trifocal tensors, the multiple view matrix seems to be a more natural tool for multiple-view analysis. Notice that, from examples 1-4 in the preceding section, the rank

[7] Property of the matrix $M_l$ which is obtained from $M$ by substituting the line measurements only.

condition simply implies all previously known multi-linear constraints, but multi-linear constraints do not necessarily imply the rank condition. Hence the use of algebraic equations may introduce certain artificial degeneracy that makes a global analysis much more complicated and sometimes even intractable. On the other hand, the rank condition has no problem in characterizing all the geometrically meaningful degeneracies in a multiple-view mixed-feature scenario. All the degenerate cases simply correspond to a further drop of rank for the multiple view matrix.

## 4.6 Applications in motion and structure recovery

The unified formulation of the rank condition may allow us to solve the problem of motion and structure recovery from multiple views using both point and line features. There are certain advantages for using point and line features together. Incidence constraints among points and lines can now be explicitly taken into account when a global estimation of motion and structure takes place. To demonstrate how this works, lets consider an image of a cube as shown in Figure 4.5. Since the corner $p$ lies on the edges $L^1, L^2$ and $L^3$,

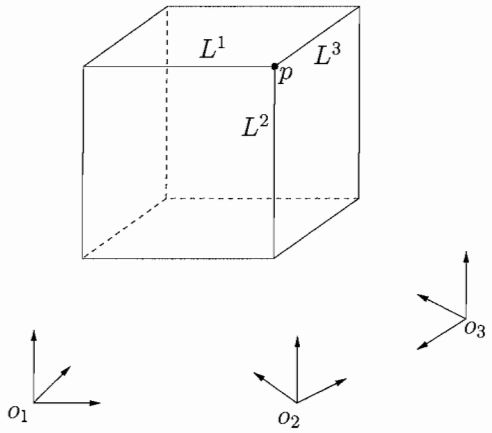


Figure 4.5: A standard cube. The three edges $L^1, L^2, L^3$ intersect at the corner $p$. The three coordinates indicate that three images are taken at these vantage points.

from three images of the cube, we have the multiple view matrix $M$ for $p$:

$$M = \begin{bmatrix} \widehat{\mathbf{x}_2} R_2 \mathbf{x}_1 & \widehat{\mathbf{x}_2} T_2 \\ \mathbf{l}_2^{1T} R_2 \mathbf{x}_1 & \mathbf{l}_2^{1T} T_2 \\ \mathbf{l}_2^{2T} R_2 \mathbf{x}_1 & \mathbf{l}_2^{2T} T_2 \\ \mathbf{l}_2^{3T} R_2 \mathbf{x}_1 & \mathbf{l}_2^{3T} T_2 \\ \widehat{\mathbf{x}_3} R_3 \mathbf{x}_1 & \widehat{\mathbf{x}_3} T_3 \\ \mathbf{l}_3^{1T} R_3 \mathbf{x}_1 & \mathbf{l}_3^{1T} T_3 \\ \mathbf{l}_3^{2T} R_3 \mathbf{x}_1 & \mathbf{l}_3^{2T} T_3 \\ \mathbf{l}_3^{3T} R_3 \mathbf{x}_1 & \mathbf{l}_3^{3T} T_3 \end{bmatrix} \in \mathbb{R}^{12 \times 2} \tag{4.49}$$

where $\mathbf{l}_i^j$ is the image of the $j^{th}$ line $L^j$ in the $i^{th}$ view. It is direct to verify that $[\lambda_1, 1]^T \in \mathbb{R}^2$ is in the kernel of $M$. In addition to the multiple images $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ of the point $p$ itself, the extra rows associated to the line features $\mathbf{l}_i^j, i, j = 1, 2, 3$ also help to determine the depth scale $\lambda_1$ (of $p$ relative to the first camera frame).

We can already see two immediate flexible ways of using the rank condition on the multiple view matrix: 1. it can simultaneously handle multiple incidence conditions associated to the same feature;[8] 2. it can simultaneously handle multiple measurements of the same feature within one view. Since incidence conditions between points and lines occur frequently in practice, especially for man-made objects such as buildings and houses, the use of multiple view matrix for mixed features is going to improve the quality of overall

[8]In fact, any algorithm extracting point feature essentially relies on exploiting local incidence condition on multiple edge features. The structure of the $M$ matrix simply reveals a similar fact within a larger scale.

reconstruction by explicitly taking into account all the geometric relationships among features of various types and with multiple measurements.

Similar to the pure point feature and line feature cases studied in Chapters 1 and 2, one may outline algorithms for motion and structure recovery based on the condition $\text{rank}(M) \leq 1$. In order to make this chapter self-contained, as an example, we here still use the cube as an example to demonstrate the essential steps of an algorithm. Without loss of generality, we here assume the number of views is still $m = 3$. Then for each of the eight corners of the cube, we have a similar multiple view matrix $M$ as given in (4.49), and we enumerate them as $M^j \in \mathbb{R}^{12\times 2}, j = 1, \ldots, 8$. Since $\text{rank}(M^j) \leq 1$, there exist $\alpha^j = [\lambda_1^j, 1] \in \mathbb{R}^2$ such that:

$$M^j\alpha^j = \begin{bmatrix} \widehat{\mathbf{x}_2^j} R_2 \mathbf{x}_1^j & \widehat{\mathbf{x}_2^j} T_2 \\ \mathbf{l}_2^{1jT} R_2 \mathbf{x}_1^j & \mathbf{l}_2^{1jT} T_2 \\ \mathbf{l}_2^{2jT} R_2 \mathbf{x}_1^j & \mathbf{l}_2^{2jT} T_2 \\ \mathbf{l}_2^{3jT} R_2 \mathbf{x}_1^j & \mathbf{l}_2^{3jT} T_2 \\ \widehat{\mathbf{x}_3^j} R_3 \mathbf{x}_1^j & \widehat{\mathbf{x}_3^j} T_3 \\ \mathbf{l}_3^{1jT} R_3 \mathbf{x}_1^j & \mathbf{l}_3^{1jT} T_3 \\ \mathbf{l}_3^{2jT} R_3 \mathbf{x}_1^j & \mathbf{l}_3^{2jT} T_3 \\ \mathbf{l}_3^{3jT} R_3 \mathbf{x}_1^j & \mathbf{l}_3^{3jT} T_3 \end{bmatrix} \begin{bmatrix} \lambda_1^j \\ 1 \end{bmatrix} = 0 \in \mathbb{R}^{12}, \quad j = 1, \ldots, 8, \tag{4.50}$$

where $\mathbf{x}_i^j$ means the image of the $j^{th}$ point in the $i^{th}$ view and $\mathbf{l}_i^{kj}$ means the image of the $k^{th}$ edge associated to the $j^{th}$ point in the $i^{th}$ view. But in order to estimate $\alpha^j$ we need to know the matrix $M^j$, *i.e.*, we need to know the motion $(R_2, T_2)$ and $(R_3, T_3)$. From the geometric meaning of $\alpha^j = [\lambda_1^j, 1]^T$, $\alpha^j$ can be solved already if we know only the motion $(R_2, T_2)$ between the first two views, which can be estimated using the standard 8 point algorithm. Knowing $\alpha^j$'s, the equations (4.50) become linear in $(R_2, T_2)$ and $(R_3, T_3)$. We can use them to solve for the motions (again). Define the vectors $\vec{R}_i = [r_{11}, r_{12}, r_{13}, r_{21}, r_{22}, r_{23}, r_{31}, r_{32}, r_{33}]^T \in \mathbb{R}^9$ and $\vec{T}_i = T_i \in \mathbb{R}^3$, $i = 2, 3$. It is then equivalent to solve the following equations:

$$P_i \begin{bmatrix} \vec{R}_i \\ \vec{T}_i \end{bmatrix} = \begin{bmatrix} \lambda_1^1 \widehat{\mathbf{x}_i^1} * \mathbf{x}_1^{1T} & \widehat{\mathbf{x}_i^1} \\ \lambda_1^1 \mathbf{l}_i^{11T} * \mathbf{x}_1^{1T} & \mathbf{l}_i^{11T} \\ \lambda_1^1 \mathbf{l}_i^{21T} * \mathbf{x}_1^{1T} & \mathbf{l}_i^{21T} \\ \lambda_1^1 \mathbf{l}_i^{31T} * \mathbf{x}_1^{1T} & \mathbf{l}_i^{31T} \\ \vdots & \vdots \\ \lambda_1^8 \widehat{\mathbf{x}_i^8} * \mathbf{x}_1^{8T} & \widehat{\mathbf{x}_i^8} \\ \lambda_1^8 \mathbf{l}_i^{18T} * \mathbf{x}_1^{8T} & \mathbf{l}_i^{18T} \\ \lambda_1^8 \mathbf{l}_i^{28T} * \mathbf{x}_1^{8T} & \mathbf{l}_i^{28T} \\ \lambda_1^8 \mathbf{l}_i^{38T} * \mathbf{x}_1^{8T} & \mathbf{l}_i^{38T} \end{bmatrix} \begin{bmatrix} \vec{R}_i \\ \vec{T}_i \end{bmatrix} = 0 \in \mathbb{R}^{48}, \quad i = 2, 3, \tag{4.51}$$

where $A * B$ is the *Kronecker product* of $A$ and $B$. In general, if we have more than 6 feature points (here we have 8) or equivalently 12 feature lines, the rank of the matrix $P_i$ is 11 and there is a unique solution to $(\vec{R}_i, \vec{T}_i)$.

Let $\tilde{T}_i \in \mathbb{R}^3$ and $\tilde{R}_i \in \mathbb{R}^{3\times 3}$ be the (unique) solution of (4.51) in matrix form. Such a solution can be obtained numerically as the eigenvector of $P_i$ associated to the smallest singular value. Let $\tilde{R}_i = U_i S_i V_i^T$ be the SVD of $\tilde{R}_i$. Then the solution of (4.51) in $\mathbb{R}^3 \times SO(3)$ is given by:

$$T_i = \frac{\text{sign}(\det(U_i V_i^T))}{\sqrt[3]{\det(S_i)}} \tilde{T}_i \in \mathbb{R}^3, \tag{4.52}$$

$$R_i = \text{sign}(\det(U_i V_i^T))\, U_i V_i^T \in SO(3). \tag{4.53}$$

We then have the following linear algorithm for motion and structure estimation from three views of a cube:

**Algorithm 7 (Multiple view motion and structure from mixed features).** *Given $m(= 3)$ images $\mathbf{x}_1^j, \ldots, \mathbf{x}_m^j$ of $n(= 8)$ points $p^j$, $j = 1, \ldots, n$ (as the corners of a cube), and the images $\mathbf{l}_i^{kj}, k = 1, 2, 3$ of the three edges intersecting at $p^j$, estimate the motions $(R_i, T_i)$, $i = 2, \ldots, m$ as follows:*

1. *Initialization:* $s = 0$

   (a) *Compute* $(R_2, T_2)$ *using the 8 point algorithm for the first two views [16].*

   (b) *Compute* $\alpha_s^j = [\lambda_1^j/\lambda_1^1, 1]^T$ *where* $\lambda_1^j$ *is the depth of the* $j^{th}$ *point relative to the first camera frame.*

2. *Compute* $(\tilde{R}_i, \tilde{T}_i)$ *as the eigenvector associated to the smallest singular value of* $P_i$, $i = 2, \ldots, m$.

3. *Compute* $(R_i, T_i)$ *from (4.52) and (4.53) for* $i = 2, \ldots, m$.

4. *Compute the new* $\alpha_{s+1}^j = \alpha^j$ *from (4.50). Normalize so that* $\lambda_{1,s+1}^1 = 1$.

5. *If* $||\alpha_s - \alpha_{s+1}|| > \epsilon$, *for a pre-specified* $\epsilon > 0$, *then* $s = s + 1$ *and goto 2. Else stop.*

The camera motion is then $(R_i, T_i), i = 2, \ldots, m$ and the structure of the points (with respect to the first camera frame) is given by the converged depth scalar $\lambda_1^j, j = 1, \ldots, n$.

We have a few comments on the proposed algorithm:

1. The reason to set $\lambda_{1,s+1}^1 = 1$ is to fix the universal scale. It is equivalent to putting the first point at a relative distance of 1 to the first camera center.

2. Although the algorithm is based on the cube, considers only three views, and utilizes only one type of multiple view matrix, it can be easily generalized to any other objects and arbitrarily many views whenever incidence conditions among a set of point features and line features are present. One may also use the rank conditions on different types of multiple view matrix provided by Theorem 1. The reader may refer to Chapter 2 for the case when $D_1^\perp$ is chosen to be $\widehat{\mathbf{l}_1}^T$.

3. The above algorithm is a straightforward modification of the algorithm proposed for the pure point case in Chapter 1. All the measurements of line features directly contribute to the estimation of the camera motion and the structure of the points. Throughout the algorithm, there is no need to initialize or estimate the 3-D structure of lines.

## 4.7 Simulations

The simulation parameters are as follows: the camera's field of view is $90^o$, image size is $500 \times 500$, everything is measured in units of the focal length of the camera, and features typically are suited with a depth variation is from 100 to 400 units of focal length away from the camera center, *i.e.*, they locate in the truncated pyramid specified by the given field of view and depth variation (see Figure 1.5). Camera motions are specified by their translation and rotation axes. For example, between a pair of frames, the symbol $XY$ means that the translation is along the $X$-axis and rotation is along the $Y$-axis. If $n$ such symbols are connected by hyphens, it specifies a sequence of consecutive motions. We always choose the amount of total motion such that all feature points will stay in the field of view for all frames. In all simulations, independent Gaussian noise with a standard deviation (std) given in pixels is added to each image point, and each image line is perturbed in a random direction of a random angle with a corresponding std given in degrees. [9] Error measure for rotation is $\arccos\left(\frac{tr(R\tilde{R}^T)-1}{2}\right)$ in degrees where $\tilde{R}$ is an estimate of the true $R$. Error measure for translation is the angle between $T$ and $\tilde{T}$ in degrees where $\tilde{T}$ is an estimate of the true $T$. Error measure for the scene structure is the percentage of $||\alpha - \tilde{\alpha}||/||\alpha||$ where $\tilde{\alpha}$ is an estimate of the true $\alpha$.

### 4.7.1 Simulations on a structured scene

In this simulation, we apply the algorithm to a scene which consists of (four) cubes only. Cubes are good objects to test the algorithm since the relationships between their corners and edges are easily defined and they represent a fundamental structure of many objects in real-life. It is certainly a first step to see how the multiple view matrix based approach is able to take into account point and line features as well as their

[9] Since line features can be measured more reliably than point features, lower noise level is added to them in simulations.

inter-relationships to facilitate the overall recovery. The length of the four cube edges are 30, 40, 60 and 80 units of focal length, respectively. The cubes are arranged such that the depth of their corners ranges from 75 to 350 units of focal length. The three motions (relative to the first view) are an XX-motion with -10 degrees rotation and 20 units translation, a YY-motion with 10 degrees rotation and 20 units translation and another YY-motion with -10 degrees rotation and 20 units translation, as shown in Figure 4.6.

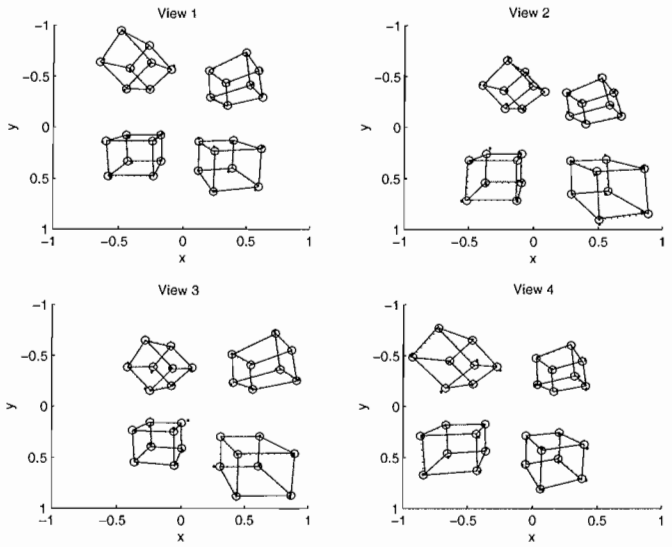


Figure 4.6: Four views of four 3-D cubes in (normalized) image coordinates. The circle and the dotted lines are the original images, the dots and the solid lines are the noisy observations under 5 pixels noise on point features and 0.5 degrees noise on line features.

We run the algorithm for 1000 trials with the noise level on the point features from 0 pixel to 5 pixels and a corresponding noise level on the line features from 0 degree to 1 degrees. Relative to the given amount of translation, 5 pixels noise is rather high because we do want to compare how all the algorithms perform over a large range of noise levels. The results of the motion estimate error and structure estimate error are given in Figure 4.7 and 4.8 respectively. The "Point feature only" algorithm is the one proposed in Chapter 1 which essentially uses the multiple view matrix $M$ in (4.50) without all the rows associated to the line features; and the "Mixed features" algorithm uses essentially the same $M$ as in (4.50). Both algorithms are initialized by the standard 8 point algorithm. The "Mixed features" algorithm gives a significant improvement in all the estimates as a result of the use of both point and line features in the recovery. Also notice that, at a high noise levels, even though the 8 point algorithm gives rather off initialization values, the two iterative algorithms manage to converge back to reasonable estimates.

### 4.7.2 Simulations on a random scene

Here we run the algorithm for 500 trials on a randomly chosen scene for each trial. Each random scene is generated by randomly generating 24 points in the truncated pyramid as shown in Figure 1.5. They are then connected by 40 randomly chosen lines. The two consecutive XX-motion and YY-motion with an incremental 10 degrees rotation and the translation is given by the so-called $T/R$ *ratio*, which is the ratio between the magnitude of translation $\|T\|$ and rotation angle $\theta$ compared at the center of truncated pyramid (see Figure 1.5). In following simulations, the ratio is 2. Comparing to the motion with previous simulations on the cubes, here the amount of translation is much bigger. That results in improved estimates for translation and structure, as shown by Figure 4.9 and 4.10. In Figure 4.10, we also compared structure estimates obtained from using point features only and those from using both point and line features. The improvement is expected.

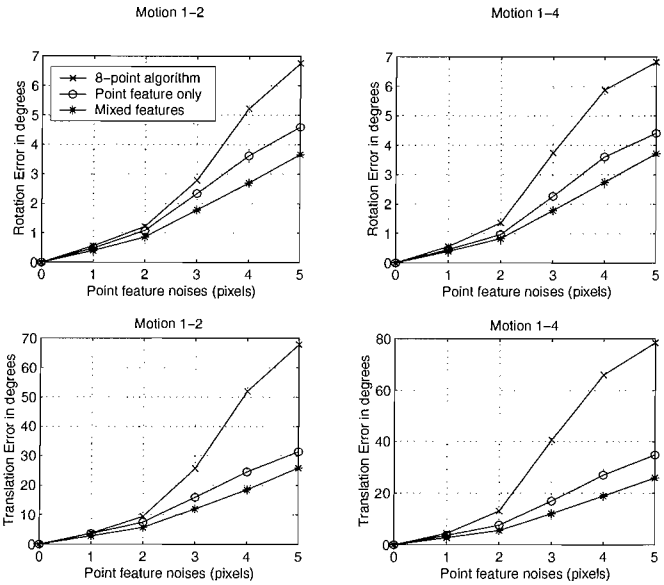


Figure 4.7: Motion estimates error versus level of noises. "Motion x-y" means the estimate for the motion between image frames x and y. Since the results are very much similar, we only plotted "Motion 1-2" and "Motion 1-4".

## 4.8 Discussions and conclusions

This chapter has proposed a unified paradigm to synthesize results and experiences in the study of multiple views of point and line features. It is shown that all relationships among multiple images of a point on a line are captured through a single rank condition on an associated multiple view matrix. All previously known or unknown constraints on multiple images simply become its instantiations. To a large extent, this condition simplifies and unifies multiple view geometry. In addition, we can now carry out meaningful geometric analysis for arbitrarily many images altogether without going through pair-wise, triple-wise and quadruple-wise analysis. Compared to conventional multiple view analysis based on trifocal tensors, the multiple view matrix based approach clearly separates meaningful geometric degeneracy from degeneracy which may be artificially introduced by the use of algebraic equations to describe constraints. In particular, as shown in this chapter, any configurations which cause a further drop of rank in the multiple view matrix correspond exactly to certain geometric degeneracy. Combined with previous results on pure point, line and planar feature cases, results on the mixed feature case given in this chapter make a coherent set of theory and algorithms for multiple view geometry and have extended the study to its full generality.

The proposed approach aims to provide people a new perspective to multiple view geometry. It will certainly have impact on both theoretical analysis and algorithm development. The linear algorithms given in this chapter and others only show a straight-forward way to use the rank condition. There are many other ways to improve them: 1. One can use any other error measures in the 3-D space or in the 2-D image to recover the motion and structure optimally subject to the rank condition; 2. Slight variations of the rank condition can handle occlusions and take into account *a priori* knowledge in the 3-D motion or scene structure; 3. Similar ideas can be applied to the study of multiple views of 3-D curves or even surface; 4. It is also straight-forward to generalize the rank condition to orthographic or many other types of projection models; 5. Better numerical methods need to be investigated to impose the rank condition; and so on. While we are still in the process of digesting the full implication of these new results, there are plenty of reasons for us to believe that we are still at a very early stage of understanding the full extent of multiple view geometry: either its theory or its practice.

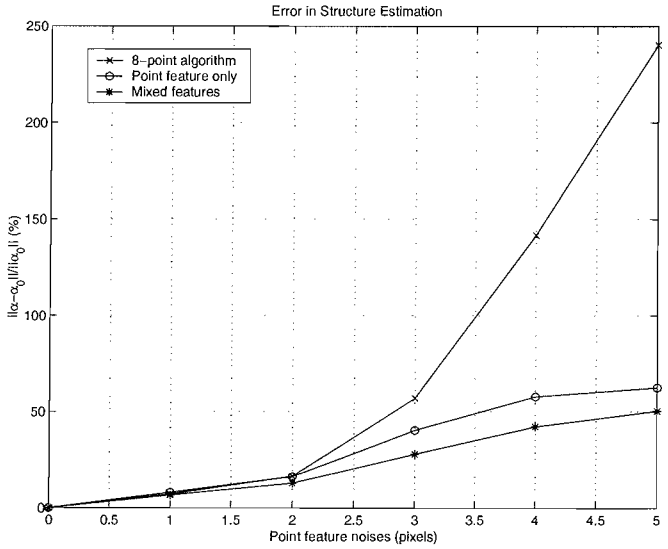


Figure 4.8: Structure estimates error versus level of noises. Here $\alpha_0$ represents the true structure and $\alpha$ the estimated.

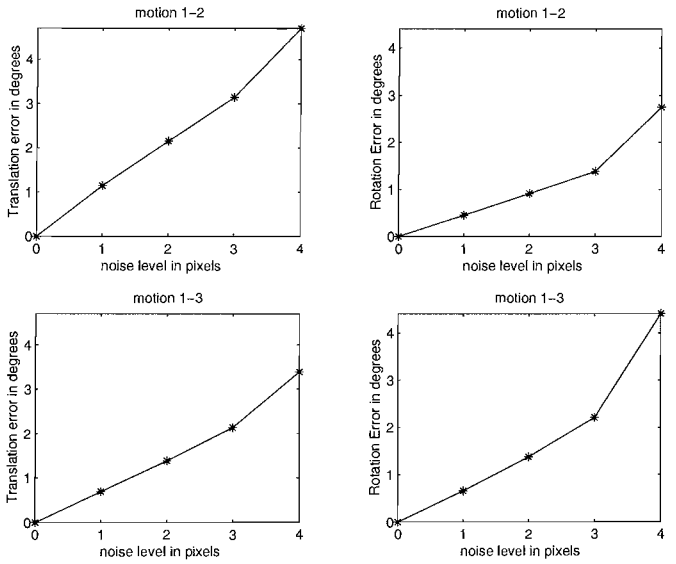


Figure 4.9: Motion estimates error versus level of noises for random scenes. "motion x-y" means the estimate for the motion between image frames x and y.

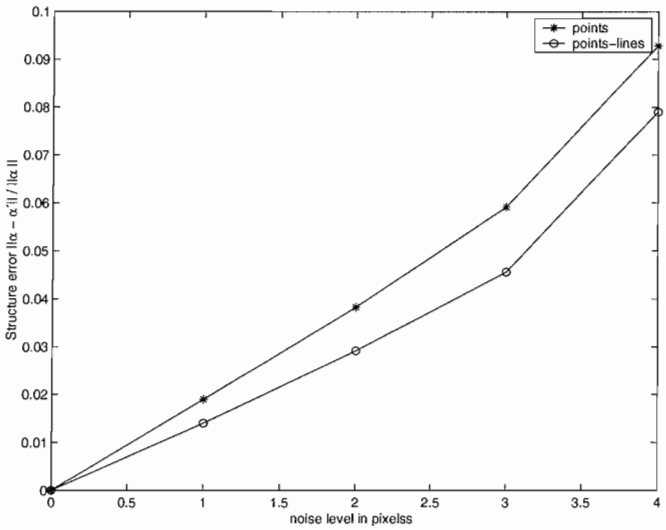


Figure 4.10: Structure estimates error versus level of noises for random scenes. Here $\alpha$ represents the true structure and $\alpha'$ the estimated.

# Bibliography


[1] S. Avidan and A. Shashua. Novel view synthesis by cascading trilinear tensors. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 4(4), 1998.

[2] B. Boufama, R. Mohr, and F. Veillon. Euclidean constraints for uncalibrated reconstruction. In *ICCV*, pages 466–470, Berlin, Germany, 1993.

[3] O. Faugeras. *There-Dimensional Computer Vision.* The MIT Press, 1993.

[4] O. Faugeras and Q.-T. Luong. *Geometry of Multiple Images.* The MIT Press, 2001.

[5] O. Faugeras and F. Lustman. Motion and structure from motion in a piecewise planar environment. *International Journal of Pattern Recognition and Artificial Intelligence*, 2(3):485–508, 1988.

[6] O. Faugeras and B. Mourrain. On the geometry and algebra of the point and line correspondences between N images. In *Proceedings 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 *Proceedings of 1994 Image Understanding Workshop*, pages 1006–1016, Monterey, CA USA, 1994. OMNIPRESS.

[8] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge, 2000.

[9] A. Heyden. Reduced multilinear constraints – theory and experiments. *International Journal of Computer Vision*, 30(2):5–26, 1998.

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

[11] T. Huang and O. Faugeras. Some properties of the E matrix in two-view motion estimation. *IEEE PAMI*, 11(12):1310–12, 1989.

[12] F. Kahl and A. Heyden. Affine structure and motion from points, lines and conics. *International Journal of Computer Vision*, 33(3):163–180, 1999.

[13] K. Kanatani. Detecting the motion of a planar surface by line & surface integrals. In *Computer Vision, Graphics, and Image Processing*, volume 29, pages 13–22, 1985.

[14] E. Kruppa. Zur ermittlung eines objecktes aus zwei perspektiven mit innerer orientierung. *Sitz.-Ber.Akad.Wiss., Math.Naturw., Kl.Abt.IIa, 122:1939-1948*, 1913.

[15] Y. Liu, T. Huang, and O. Faugeras. Determination of camera location from 2-D to 3-D line and point correspondences. *IEEE Transactions on PAMI*, pages 28–37, 1990.

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

[17] H. C. Longuet-Higgins. The reconstruction of a plane surface from two perspective projections. In *Proceedings of Royal Society of London*, volume 227 of *B*, pages 399–410, 1986.

[18] L. Quan and T. Kanade. A factorization method for affine structure from line correspondences. In *Proceedings of the CVPR*, pages 803–808, 1996.

[19] C. Schimdt and A. Zisserman. The geometry and matching of lines and curves over multiple views. *International Journal Of Computer Vision*, 40(3):199–234, 2000.

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

[21] A. Shashua. Trilinear tensor: The fundamental construct of multiview geometry and its applications. In *International Workshop on Algebraic Frames For The Perception Action Cycle (AFPAC)*, 1997.

[22] A. Shashua and L. Wolf. On the structure and properties of the quadrifocal tensor. In *the Proceedings of ECCV, Volume I*, pages 711–724. Springer-Verlag, 2000.

[23] M. Spetsakis and Y. Aloimonos. Structure from motion using line correspondences. *International Journal of Computer Vision*, 4(3):171–184, 1990.

[24] M. Subbarao and A. M. Waxman. On the uniqueness of image flow solutions for planar surfaces in motion. *Third IEEE workshop on computer vision: representation and control*, pages 129–140, 1985.

[25] C.J. Taylor and D. Kriegman. Structure and motion from line segments in multiple images. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 17(11), 1995.

[26] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography. *Intl. Journal of Computer Vision*, 9(2):137–154, 1992.

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

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

[29] B. Triggs. Autocalibration from planar scenes. In *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, 1998.

[30] R. Vidal, Y. Ma, S. Hsu, and S. Sastry. Optimal motion estimation from multiview normalized epipolar constraint. In *ICCV'01*, Vancouver, Canada, 2001.

[31] A. Waxman and S. Ullman. Surface structure and three-dimensional motion from image flow kinematics. *Int. J. Robotics Research*, 4(3):72–94, 1985.

[32] J. Weng, T. Huang, and N. Ahuja. Motion and structure estimation from line correspondences: Closed-form solution, uniqueness and optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(3):318–336, March 1992.

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

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

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