# Rank Classification of Linear Line Structure in Determining Trifocal Tensor

ZHAO, Ming

A Thesis Submitted in Partial Fulfillment

of the Requirements for the Degree of

Master of Philosophy

in

Automation and Computer-Aided Engineering



July 2008

Thesis/Assessment Committee

Prof. Yun-hui Liu(Chair)

Prof. Chung, Chi-kit Ronald(Thesis Supervisor)

Prof. Jiaya Jia(Committee Member)

Dr. Bebis George(External Examiner)

# Abstract

This thesis addresses how the motion of a camera can be tracked and determined from the video data captured by the camera. For reliability of the tracking and motion determination tasks we consider three consecutive image frames at a time in the video data, and examine how the camera motion as determined by such a triplet of consecutive images can be tracked and determined temporally. A concise quantity that captures the essentials of the camera motion in a triplet of images is the trifocal tensor. There has been much work on how the trifocal tensor can be determined from point correspondences over the image triplet. In man-made scenes lines are often abundant and are as indispensable as point features. This work focuses on how the tracking and motion determination tasks can be achieved by the use of line correspondences in the video data.

This thesis first considers the degeneracy in camera motion estimation and projective reconstruction when image line correspondences over three views, and in particular those projected from linear line structures in space, are used. The observed set of lines in space are called critical if there are multiple projectively non-equivalent configurations of the camera positions that can picture the same image triplet of the lines. In other words, the line set is critical if their image projections to three views do not allow the three associated camera positions to be uniquely determined. This work examines the degeneracy of the camera positions from the under-determination of the associated trifocal tensor. It is shown here that the matrix that leads to the determination of the trifocal tensor, termed the tensor estimation matrix, is useful in classifying the linear line structures. We show that the rank of the tensor estimation matrix is reduced to 7, 11, 15 if the observed lines come from a line pencil, a line bundle, and a line field respectively, which are line families belonging to linear line space; and 12, 19, 23 if the lines come from a general ruled surface, a general linear congruence, and a general linear line complex, which are subclasses of

linear line structures. In addition, for each of the above line structures, we provide how many lines of the structure must at least be observable so that the full information of the line structure is displayed in their image data. Such critical structures are quite typical in reality, and thus the findings are important to the validity and stability of practically all algorithms related to structure from motion and projective reconstruction using line correspondences.

This thesis also provides a framework of how the camera motion can be tracked and determined over time should the observed line structure be general enough to allow a unique trifocal tensor solution over each image triplet to be attained. The framework consists of the following: a simple line tracking algorithm that can establish line correspondences from the image sequence, and an extended Kalman Filter whose application can recover the camera motion parameters up to projectivity.

Synthetic data and real image experiments are both presented to illustrate the validity of the findings.

# 摘要

本論文主要考量如何從視頻信息中跟蹤和估計攝像機運動的問題。基於跟蹤和運動估計的可靠性，我們考慮利用視頻信息中的三張連續圖像，研究如何從三張連續圖像估計攝像機運動。三視圖張量是一個從實質上描述三張圖像裏攝像機運動的簡潔方式。已經有許多工作關於三視圖張量如何從三圖像點對應中得到。在人造實景中，綫的數量通常比較多，和點特徵一樣不可或缺。我們的工作集中在如何通過視頻信息中的綫對應完成攝像機跟蹤和運動估計。

本論文首先考慮給定三視圖綫對應情形下，攝像機運動估計和投影重搆中的退化問題，尤其是使用這些綫性結構的圖像時。如果多個投影不等效的攝像機位置可以照出相同的圖像，這組空間中的綫被稱爲*退化的*。也就是說，這個綫集合是退化的，如何從它們的三個圖像不能唯一確定攝像機位置。這個工作考察了由於相關聯的三視圖張量不能完全被決定的退化情形。用來估計三視圖張量的張量估計矩陣在分類綫性結構上非常有用。我們指出張量估計矩陣的秩減少到7、11、15，如果觀察到的綫分別來自一個綫組、綫群和綫場，它們是屬於綫性綫空間的綫族；減少到12、19、23，如果綫來自一個一般規範曲面、一般合同綫束和一般複合綫束，它們是綫性結構的子類。此外，對於上述每個綫結構，我們給出最少需要綫的數量，使得該結構的全部秩的性質都能從圖像信息中反映出來。這些退化的結構在現實中很常見，因此這個工作對於實踐中使用對應綫的基於運動的結構估計和投影重搆算法的有效性和穩定性都很重要。

本文也提供了一個跟蹤並估計攝像機隨時間運動的框架，從三張圖像觀察到的綫結構足夠得到唯一的三視圖張量。該框架包括下面這些部分：首先，一個簡單的綫跟蹤算法從圖像序列中找出對應綫；其次，應用擴展卡曼濾波器來找出攝像機運動參數。

我們同時給出了模擬數據和實際圖像試驗來證明該工作的正確性。

# Acknowledgments

Here I wish to express my sincere gratitude to my supervisor Prof. Chi-kit Ronald Chung who has guided me on identifying fruitful research directions, inspired me on generating new ideas, and encouraged me to explore deep problems throughout the two years of my study. Looking back, his demand of extreme precision and systematic treatment to problems benefited not only my research but also my life. I would also like to thank my thesis committee members Prof. Leo Jiaya Jia and Prof. Yun-hui Liu who have made invaluable comments on my research.

The support from my colleagues in the Computer Vision Laboratory has also been very helpful to my work. In particular, I would like to thank Ding Yuan, Zhan Song, Homing Chim, Wei Wang, and Yong He for their selfless help to me. I also greatly appreciate the contributions of a number of key researchers in Computer Vision; with their work I can stand on the shoulders of the giants and see further. Last but not least, my parents have always been my ultimate source of energy. They gave me everything in their lives without claiming anything.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

This chapter begins with the motivation of this work and outlines the objectives and challenges in exploring the degeneracy in using line correspondences to estimate the trifocal tensor and camera motion. Next, the original contributions of this work are listed. Finally, we give a brief overview of the organization of the thesis with forward pointers to various chapters.

## 1.1 Motivation

The topic of this thesis is degeneracy of trifocal tensor which captures all geometric constraints of three uncalibrated images as its counterpart-fundamental matrix for two-view, and camera motion estimation using line correspondences. As for line features, several structure and motion estimation algorithms that use line correspondences have been proposed in the literature [1], [2, 3, 4]. However, the ambiguity issue possibly involved in the use of line correspondences has not been given as much attention as to the use of point correspondences. In fact, compared to points, the advantage of using lines are that they can be located accurately in the image and tracked reliably from one image to the next; available evidence suggests that the construction from lines is more accurate than reconstruction from points, at least if the camera calibration is known [5].

The rank classification of tensor estimation matrix is based on different linear line structures, including line pencil, point star, ruled plane, linear ruled surface, linear line congruence and linear line complex. These critical structures are quite typical in real-

ity, and thus the findings are important to the validity and stability of practically all algorithms related to structure from motion and projective reconstruction using line correspondences.

Next, a minor part of the thesis is about camera motion estimation, which has a wide range of applications, such as real-time robot localization, navigation for camera-enabled handheld devices (mobile phones, PDAs), automatic CAD modeling, vehicle navigation. It is more useful to estimate motion in urban scenario where the signals from Global Positioning System (GPS) may be blocked by buildings [6]. As mobile phone and embedded camera are widely available, people and automobiles in downtown of city can be located more accurately using visual information and 3-D scene model than the use of GPS, not to mention the case that GPS signal is blocked by buildings. The scene characteristics of typical urban environment consist of regular components such as buildings with distinctive points and parallel line segments, and undesired components such as moving people and cars (non-rigidity), trees (homogeneity), and vitreous window (lighting).

## 1.2 Objective of the study

This work aims at providing a more comprehensive treatment of the various critical configurations of line structures that can be present in real scenes, and how they impact the determination of trifocal tensor from image data.

We address the question of, given line correspondences over three views, what is the condition of the observed lines for the relative geometry of the three associated camera positions in space to be uniquely recoverable? There are at least two issues: what particular structure the lines in space belong to, and whether there are enough of the lines. In this work we deal with both the structure issue and the quantity issue, and we deal with them from the perspective of trifocal tensor – the $3 \times 3 \times 3$ quantity that captures the relative positions of the cameras in relation to the views. We investigate whether the observed lines are of a too specific structure in 3-D space to allow the trifocal tensor to be uniquely recoverable, and if they indeed are, by how much they are shy of a general enough structure. We also investigate, for each of the investigated line structures, how many lines are minimally needed to allow the full information of the line structure to be

reflected in the image projections.

A set of observed lines in space are called *critical* if there are multiple projectively non-equivalent configurations of the camera positions that can picture the same image triplet of the lines. We shall examine the different classes of structure of the line set that are critical. In all cases, we assume that the camera positions are general with respect to the set of lines, meaning that the images are not degenerate images like those having multiple lines projected to the same image line, a result of accidental alignment.

Trifocal tensor, an important quantity in multiple view geometry, is the counterpart of fundamental matrix; trifocal tensor to three views is as fundamental matrix to two views. Trifocal tensor has drawn much attention in the last decade or so, for it can be computed directly from image correspondences, and its determination would allow camera matrices and in turn projective structure of the imaged scene to be reconstructed.

In the case of four views, quadrifocal tensor [7] captures the relationship between a set of four lines that provide 9 linear equations. The reason that trifocal tensor is frequently used in contrast to quadrifocal tensor is the following: first, quadrifocal tensor contains 80 degrees of freedom with 51 admissibility constraints (nonlinear constraints), which is much more complicated than trifocal tensor with 26 degrees of freedom and 8 admissibility constraints. Second, it is known that the quadrilinear tensor is algebraically dependent on the associated fundamental matrices and trifocal tensor, and thus does not add independent new constraints. Finally, four views is the maximum number for which such an object exists (for 5 views or more such multi-linear constraints cannot be conveniently captured using a single tensor), which means the tensor method does not extend to more than four views [7]. The above explains why there are only limited number of works that use quadrifocal tensor.

Existing algorithms of determining trifocal tensor are mainly iterative and require initialization typically from a linear method. The degeneracy of camera motion estimation using linear method has been studied in a number of works. However, the works are mainly about the cases where point correspondences are used. There have been a few earlier works on the cases where line correspondences are used, but the treatment has been conducted with the use of direct geometric primitives namely the relative displacements and orientations of the cameras [8]. Since trifocal tensor has the essence of expressing the

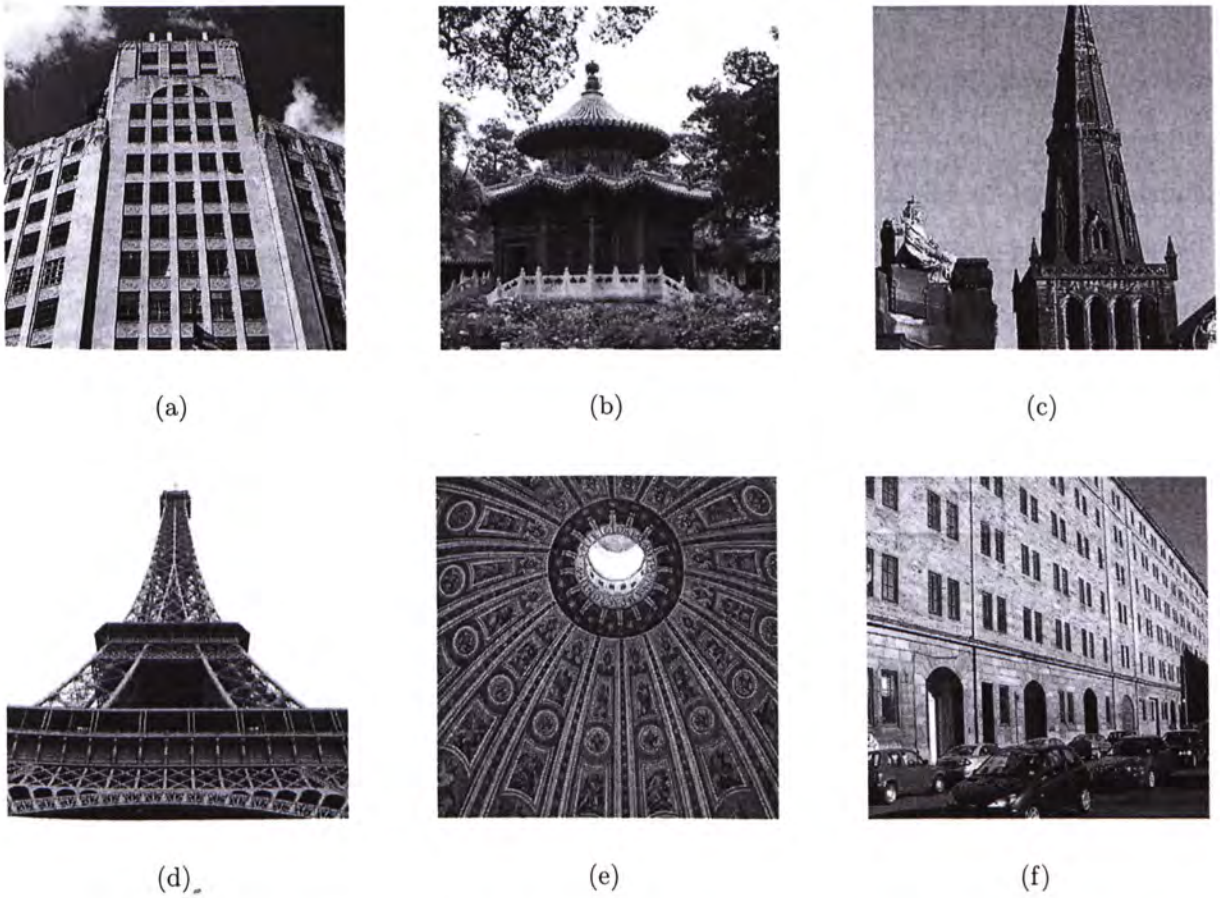
relative positions of all cameras in relation to the image observables in a single collective term, we believe the study would be simpler and the degeneracy be more easily quantified if the issue is pursued from the angle of trifocal tensor, and this work aims at doing that.

Line correspondences are in many ways as indispensable as point correspondences. There is evidence [9, 10] that 3-D reconstruction from lines could be more accurate than from points. Moreover, linear line structure as the structures to be addressed in this paper do appear quite often in real scenes, as exemplified in Fig. 1.1. Putting it simply, linear line space includes three subspaces: line pencil as a family of lines that contain the same point and that intersect at a non-incident line (in other words, the lines are both co-planar and co-intersecting at a point on the plane); point-star (also named bundle of lines) as a family of lines that are concurrent at a point; ruled plane (also named field of lines) as a family of lines incident on a particular plane. More general than linear line space, linear line structures include linear line complex which are lines satisfying one linear constraint, linear line congruence satisfying two linear constraints, and linear ruled surface-lines satisfying three linear constraints. In Fig. 1.1(a) a typical linear line complex structure is shown. The line sets extracted from the roof of the pavilion in Fig. 1.1(b) and Fig. 1.1(c) are bundles of lines that belong to linear line congruence. Eiffel tower in Fig. 1.1(d) and the roof of the church in Fig. 1.1(e) can also be approximated as several sets of line bundle. The building in Fig. 1.1(f) consists of a field of lines that are coplanar and they also belong to linear line congruence.

Next, it provides a framework for determining camera motion integrating Kalman Filter and trifocal tensor. The goal is the following, compute the position and velocity of the camera from a image sequence capturing a static scene. Generally the 3-D information is not directly accessible, thus, we can foresee the advantage to estimate camera motion without knowing any 3-D structure using trifocal tensor.

## 1.3 Challenges and our approach

The degeneracy problem is challenging because there are multiple perspectives considering degeneracy, such as degenerate motion of the camera, and degenerate structures (point, lines). Besides, line correspondences are in many ways harder to handle than



(a) (b) (c)

(d) (e) (f)

Figure 1.1: Scenes consisting of linear line complex and linear line congruence (in particular the sub-class point-star, and ruled plane) respectively.

point correspondences, for the reason that their representation, say the Plücker's coordinates, contain nonlinear constraint (Plücker equality) between the coordinates. And the projection of lines makes the nonlinearity in Plücker's coordinates even more complicated. However, the key issue is the way that we treat the degeneracy of tensor estimation - a novel perspective from the rank of tensor estimation matrix, which allows a systematic classification of degenerate linear structures.

Linear features have several advantages for tracking, however, line tracking is difficult primarily because there is no strong geometric constraint available, compared to point tracking where epipolar constraint decreases the searching dimension. Matching lines as individual or group proves to be difficult. Applying epipolar constraint to ending points of line segment is not reliable. Secondly, in engineering practices, the extraction of lines has some deficiencies in connectivity and the piecewise linearity of the real line in 3-D. Thirdly,

the partial occlusion gives considerable trouble as the 3-D object model is unknown. In data acquisition process, we may only have the blurred images in video due to the finite shutter speed. Lastly, the robustness and computational complexity are always the keen considerations for real time video applications.

For camera motion estimation, the scene complexity in reality comes from non-rigidity, lighting effect, homogeneity and so on. The problem of computational stability and degenerate motion and structures also effect the algorithm. Additional challenges rise from projective to Euclidian $(\mathbf{R}, \mathbf{t})$ measurement and similarity between small translation and rotation. To overcome such difficulties, we have made the following assumptions to simplify the problem: 1) Target scene: regular urban scene consists of buildings and roads (straight lines); 2) Scene rigidity and smooth motion (for filtering); 3) Dense sample rate with respect to motion velocity; 4) Near planar motion.

Our approach integrates the use of trifocal tensor, extended Kalman Filter (EKF). The process for camera motion estimation can be decomposed to three steps: 1) Feature tracking across views; 2) Relative camera motion estimation with respect to the first camera position; 3) Model matching for motion in Euclidean space. The proposed approach holds the following advantages. First, linear feature is usually sparse, allowing more accurate tracking result than point feature [9, 10]. Secondly, different lines detected in a single image are likely to be relatively distant from each other, in contrast to distinctive point features that are usually locally concentrated. Therefore, line correspondences across multiple views may capture more accurate information than points.

## 1.4 Original contributions

The major contributions presented in the thesis are listed in Table. 1.1. The author published two papers [11, 12] related to this topic.

## 1.5 Organization of this dissertation

The thesis is organized as follows:

We start in Chapter 2 by giving an introduction about the ideas behind degeneracy of

Table 1.1: Original contributions.

1. Rank classification of tensor estimation based on line geometry;
2. Proof of rank properties in 5 general Lemmas in algebra;
3. Proof of rank $7, 11, 15$ in tensor estimation for line pencil, point star and ruled plane respectively;
4. Proof of rank $12, 19, 23$ in tensor estimation for linear ruled surface, linear line congruence and linear line complex respectively;

trifocal tensor estimation and camera motion estimation, and present a survey of previous research related to our problem. Chapter 3 encloses three-view geometry and trifocal tensor, including trilinearities, line transfer, and computation of the trifocal tensor. Algebraic line geometry related to this work is described in Chapter 4, including Plücker line coordinates, line projection matrix and classification of linear line structures. We describe the critical three-view configurations for trifocal tensor determination in Chapter 5. Line tracking and camera motion estimation framework are proposed in Chapter 6. Experimental results of both simulation and real data are presented in Chapter 7. Finally we summarize our findings and highlight possible future work in Chapter 8.

# Chapter 2

# Related Work

In this chapter, we will first outline some of the fundamental concepts in degeneracy of trifocal tensor estimation, and give a brief survey of previous work related to the problem, specifically the work regarding line correspondences and line geometry. The survey is brief, as the number of related work is limited. It is next shown about the related work on camera motion estimation. The intention here is to outline some of the shortcomings in previous work and to gain some insight about possible solutions to the problem.

## 2.1 Critical configuration for motion estimation and projective reconstruction

In this thesis we are concerned primarily with recovering camera displacement or camera motion from visual observations. The problem has been studies both in photogrammetry and computer vision community. Point and line features can both be used in motion estimation from multiple views. Certain algebraic properties of correspondences over multiple views are summarized in [13] and [7].

### 2.1.1 Point feature

Critical configuration of point features, in particular, has already drawn much attention in the literature, and the studies have been about the condition of the point features that do not allow the camera motion to be determined uniquely.

**Two-View.** In the two-view case, the only critical setting is that all 3-D points and camera centers in space lie on a ruled quadric [14]. However, [15] has shown that not all ruled quadric are critical. The critical configurations and their conjugates for two-view only include the following: Hyperboloid $\mathbf{XY} = \mathbf{ZT}$ - centers on a common generator and Cone $\mathbf{X}^2 + \mathbf{Y}^2 = \mathbf{Z}^2$ - camera centers on different generators, Hyperboloid $\mathbf{XY} = \mathbf{ZT}$ - centers not on a common generator and itself, Cone $\mathbf{X}^2 + \mathbf{Y}^2 = \mathbf{Z}^2$ - one camera center at the vertex and itself, two planes $\mathbf{XY} = 0$ - cameras on one plane, but not on the intersection and itself, two planes $\mathbf{XY} = 0$ - one camera on the intersection, the other not and itself, single plane $\mathbf{X}^2 = 0$ - both cameras on the plane and itself, cone $\mathbf{X}^2 + \mathbf{Y}^2 = \mathbf{Z}^2$ - both cameras on the same generator, but not the vertex, two planes $\mathbf{XY} = 0$ - camera centers not on the same plane. Another important result is that with two perspective views of 5 point correspondences, there are 10 possible solutions to structure and motion [16].

**Three-View.** In contrast with two-view, there is no such single "critical surface" for point correspondences in three-view [17]. [18] has shown that in general there is a unique solution for relative orientation of three-view using 4 point correspondences on a fixed object. However, multiple solutions are possible in rare cases, even when the 4 points are not coplanar. They expressed the rotations in quaternion form and use an equivalent form of "Longuet-Higgins equations" for motion. The result is that given the perspective projections of four points on an object at three views, then there is almost always a unique solution for the motion and structure of the object.

[19] pointed out that three images of 6 points is subject to a three way ambiguity, then in [20], the author proposed that the three way ambiguity is preserved as long as the optical center of camera remains on a certain quadric surface and the ambiguity does not hold only if the optical center leaves the quadric surface. The proof depends on the properties of certain Cremona transformation of the plane. The Cremona transformation relevant to reconstruction from images of six points is of type $C^{10}[6^1, e^7]$. In brief, for noncollinear sets of six points from three views, Cremona transformation of type $C^{10}[6^1, e^7]$ yields a linked triple which corresponds to three way ambiguity. In addition, the three way ambiguity can be resolved by a further motion of camera.

Generally 7 points are enough to allow the trifocal tensor to be retrieved uniquely.

However, it has been shown that though a single "critical surface" is not present, there are still critical configurations in the three-view case, which include the configuration of the cameras and 3-D points all lying on the intersection of two quadric surfaces, the configuration of all 3-D points being coplanar and the three cameras being off the plane, and the configuration of all points lying on a twisted cubic and at least one of the three cameras being embedded on the cubic [21]. [17] proved that the critical configuration is a finite set of 10 points (including the camera centers) that are the intersection of a linear system of cubic surfaces. The 10 points must satisfy the three-view geometry, hence a general set of 7 points from 3 general cameras does not lead to any ambiguity.

**N-View.** As for the $n$-view case, the critical configuration is that all 3-D points and cameras lie on the intersection of two distinct ruled quadrics [22]. [23] investigated critical configuration for projective reconstruction from multiple images taken by a camera moving along a straight line. It is shown that the image sequences from a configuration consisting of any number of cameras lying on a straight line and any number of points lying on a twisted cubic, and a set of points and cameras all lying on a rational quartic curve.

A classification of multiple-view critical configurations [23] is provided here for an overview.

- If the three quadrics meet in isolated points, then the configuration is critical of 7 of those points, and the three cameras defining the quadrics

- If the three quadrics are the same, then all the cameras in the conjugate configuration are collinear. The critical set does not consist of the complete quadric, but only of a rational quartic curve lying on the quadric. The conjugate configuration is a twisted cubic/line configuration.

- If the three quadrics are linearly dependent (pencil of quadrics), then the critical configuration is elliptic quartic curve (the intersection of the quadrics) and any number of points and cameras lying on this curve.

- If three quadrics meet in a twisted cubic, the critical configuration is any number of points lying on the twisted cubic and cameras on an arbitrary line.

- If the three quadrics meeting in conics or lines, then all these possible intersections are critical configurations.

[15] summarized the latest results on critical configurations from multiple views.

The reason of reviewing the previous work about critical configurations on point features is to gain some insight about the methodologies and research perspectives on the subject, which would definitely enhance our studies to line feature.

### 2.1.2 Line feature

A pioneering work was done by Buchanan [24, 25] who pointed out that $(3, 6, 5)$-congruence is a critical line set (A congruence of order n, class m, and rank r is referred to as a $(n, m, r)$-congruence; the number of lines that is constant for almost all points of 3-space is defined as the order of line congruence, the finite number of lines that a general plane in 3-space contains is defined as its class, and rank of congruence is defined as the finite number of points $\mathbf{P}$ on a line $\mathbf{l}$ in 3-space with the property that $\mathbf{l}$ together with two elements of congruence through $\mathbf{P}$ lie on a plane, for example, $(1, 0, 0)$-congruence is called the point-star), and suggested a few line sets that defeat the Liu-Huang algorithm [26] of recovering motion and structure from line correspondences.

Maybank [5] continued Buchanan's work by choosing to use Semple's representation to describe lines so as to parameterize the line congruence by surfaces of degree 6 in $\mathcal{P}^4$, while Grassmann coordinates express the line set by a surface of degree 9 in $\mathcal{P}^5$. However, Semple's representation does not have a direct way of expressing the projection of 3-D lines to image lines. In Maybank's work it is also shown that line congruence can be represented by a general plane of $\mathcal{P}^4$ or a quadric hypersurface of $\mathcal{P}^5$. It is shown that $(3, 6, 5)$-congruence is a critical set of lines for reconstruction from three images taken by uncalibrated cameras. In Semple's parameterisation of lines of $\mathcal{P}^3$, a line congruence of type $(3, 6, 5)$ is represented by a surface in $\mathcal{P}^4$ of degree 6 known as a *Bordiga surface.* Then, a general Bordiga surface containing the rational normal quartic parameterises a critical line congruence. Next, if three uncalibrated cameras are used to obtain images of 10 lines in general positions, then reconstruction from these images is unique up to a collineation ambiguity in space. If cameras are calibrated, then it is known that six lines

are sufficient for a unique reconstruction.

Navab and Faugeras [27] addressed the analysis of the critical sets of lines in the Euclidean domain using two equations from Liu and Huang [26] and related the degeneracy analysis to the Liu-Huang algorithm. The treatment is however in terms of geometric primitives explicitly related to individual camera positions and orientations, not the collective quantity trifocal tensor.

As discussed earlier, trifocal tensor is a concise quantity that captures how the positions and orientations of all cameras are related to image observations. We believe there is advantage in treating the degeneracy issue in terms of trifocal tensor, especially when observables are not direct 3-D but only image data. In particular, as we shall elaborate in this report, the rank of a matrix – the matrix that leads to the determination of the trifocal tensor, which hereafter we shall refer to as the *tensor estimation matrix* – could be a measure of the degeneracy of the observed line set, and the parallel measure in direct geometric primitives is not as accessible.

The work of Stein and Shashua [28] is among the very few that ever look at the degeneracy of line structures from the viewpoint of trifocal tensor. The work focuses on the case that the observed line set is a linear line complex (LLC), and points out that in such a case the tensor estimation matrix has the rank reduced to at most 23. It is shown that the ambiguity of tensor estimation from LLC structure is at most an 8-fold ambiguity. As the null space span the admissible tensors, three standard correlation matrices of the admissible tensors can be spanned by three 1-parameter family of matrices. They also observed that only one of the 8 solutions satisfies all the admissibility constraints, as the rank-2 constraint of the standard correlation matrices is closed under linear superposition. An algorithm for recovering trifocal tensor in the LLC case was proposed as in Table. 2.1. Besides, they found that the rank of estimation matrix from three-view of coplanar points is at most 21. However, the null space includes admissible tensors, thus a unique solution is not available.

There are however line structures other than LLC that can also show up in reality. This work aims at providing a more comprehensive treatment of the various critical configurations of line structures that can be present in real scenes, and how they impact the determination of trifocal tensor from image data. The problem is challenging because

Table 2.1: Algorithm for recovering trifocal tensor in the LLC case.

1. Robust estimation to determine line correspondences which belong to the LLC.
2. Create 3 'ghost tensors'.
3. Use point-line-line correspondences from three view to compute the tensor estimation matrix **A**.
4. Find the 4th vector spanning the null space of **A**.
5. Find three scalars so that an admissible tensor is obtained using the admissibility constraints.

line correspondences are in many ways not as easy to handle as point correspondences, for the reason that their representation, say the Plücker's coordinates, contain nonlinear constraint (Plücker equality) over the coordinates.

In terms of the 9 admissibility constraints, we do not address this issue to every line structures for the following reasons: first, many degenerate line structures only provide linear constraints on tensor estimation from which trifocal tensor can not be recovered even with the full use of admissibility constraints; secondly, the tensor estimation from LLC has been addressed already in Stein and Shashua [28]. These constraints can not applied in a explicit way because they are nonlinear constraints of higher degrees.

## 2.2 Camera motion estimation

Camera motion estimation consists of three parts: feature extraction (not included below), feature tracking and camera motion determination. A brief summary of different approaches is given below. Features that can be used in determining camera motion include points, junctions and lines.

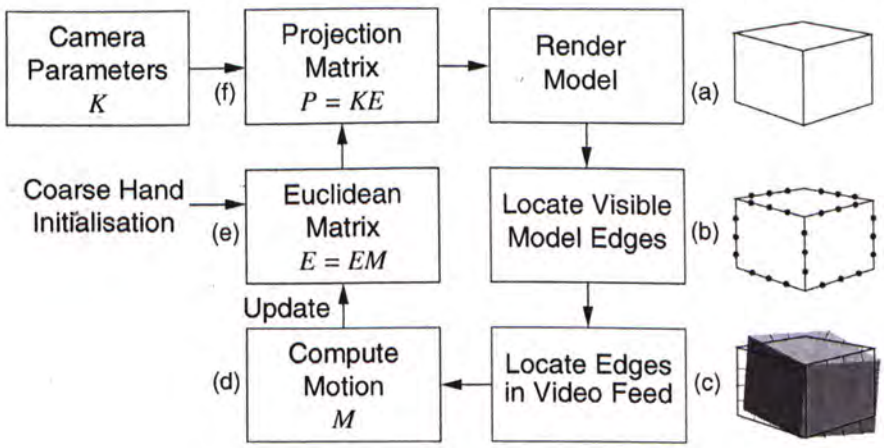
To track features across multiple images, point is the most common feature to establish correspondences [29]. The well-known KLT tracker is among this category. However, point features may not be available in large quantities and generally, the mismatch and unreliability of point features are larger than lines. As lines are determined by a set of pixels, it is possible to locate lines more accurately than points [30] given the homogeneous attribute for points within a small region. Thus, We primarily focus on lines.

### 2.2.1 Line tracking

Although there are many literature in line tracking, it is still a difficult task. The end points and connectivity of line feature are not stable. And there is no strong disambiguating geometric constraint available for lines [31]. Early approaches to tracking were mostly edge-based because they are computationally efficient and stable to lighting changes. The existing methods can be separated into two groups [32]: one looks for gradients in the image around a first estimation of object pose and it is fast and general; another approach extracts image contours and fits the model outlines to the contours, and this is slow, less general, and more robust.

First, we summarize some model-based tracking methods. Drummond and Cipolla [33] described a robust real-time wire-frame tracking system that combined the graphical rendering with constrained active contour tracking. Exponential map is used to represent of transformation of structure between two frames and this inter-frame transformation can be approximated by a linear term. Based on the strong feature of the object model, the search for intensity discontinuities is limited to one dimensional. Binary space partition trees are used to dynamically determine the visible features of the model in real-time. The whole system is shown in Fig. 2.2.1. Least-squares algorithm minimizes the transformed edge position and the actual edge position when estimating the transformation across frames. Instead of using RANSAC or M-estimator, it incorporates iterative reweighted least squares (IRLS) to improve its robustness to occlusion.

Figure 2.1: Drummond and Cipolla's model-based tracking system.




In contrast to the methods that rely on matching points sampled on edges, an alterna-

tive is to globally match model with straight line segments [34] extracted from the image based on the "Mahalanobis distance". Given the attribute vector $\mathbf{X}_m$ of a model and the attribute vector $\mathbf{X}_d$ of a tentative segment, the Mahalanobis distance between $\mathbf{X}_m$ and $\mathbf{X}_d$ can be defined as

$$d = (\mathbf{X}_m - \mathbf{X}_d)^T(\Lambda_m + \Lambda_d)^{-1}(\mathbf{X}_m - \mathbf{X}_d) \tag{2.1}$$

where $\Lambda_m$ and $\Lambda_d$ are the covariance matrix of $\mathbf{X}_m$ and $\mathbf{X}_d$ respectively. The minimization is performed using the Levenberg-Marquardt algorithm.

[35] proposed to apply direct optimization on gradients, for example, maximize the gradient norm along the model reprojection. Some chose to maximize the correlation between the predicted and measure gradient norm plus an additional term to constrain the motion. They require a very good initial estimate to converge to a correct pose.

Next, let us move on to tracking without 3-D models. It is possible to simultaneously estimated both camera motion and the scene geometry without priori 3-D models, known as Simultaneous Localization and Mapping (SLAM) that concerns about global representation, in contrast to Structure from Motion (SFM) that focuses more on local reconstruction. It robustly estimates the relative poses between three consecutive cameras using RANSAC in three-view.

Fitzgibbon and Zisserman [36] described an automatic camera tracking method involving the following steps: (1) compute interest points in each frame; (2) match the interest points between neighboring frames using robust estimation of multiple view tensors; (3) compute a projective reconstruction from the interest point matches; (4) compute a metric reconstruction. They used an occupancy matrix to identify whether 3-D points are visible in certain views. The minimization of re-projection error is achieved using bundle adjustment. If the prior knowledge about the camera parameters, camera motion and scene constraints are available, the optimal estimate must minimize the bundle error subject to these constraints, thus, they formulated a penalty function considering the priors.

Schmid and Zisserman [31] used both grey level information and the multiple view geometric relations for image triplets. The line segments are extracted by applying Canny edge detector. Their method treats short range motion and long range motion differently.

For short range motion, each segment is treated as a list of points so that neighborhood correlation is a measurement of similarity. The corresponding points could be obtained by searching along each line segment with a winner takes all strategy. In the F-guided matching, the matching score for a line pair is computed as the average of the individual correlation score for the points of the line.

Last but not least, the tracking by detection methods make the system more realistic in applications. In [37], the line detector/tracker can be incorporated together. Candidate points (edgels) for lines can be found from sparse sampling of image data by applying either the isotropic or oriented detector to a randomly chosen small region of the image, and it uses a RANSAC grouper to find the line segments based on the edgels. The line segment is tracked by a small number of linear searches normal to the line at its predicted position. Given two previous views, line position in the third view can be predicted. The robust estimate of the true line segment is obtained from RANSAC optimization. The unguided tracking continues until the structure based tracking can be established.

Rosten and Drummond [38] proposed a real-time 3-D model-based tracking by combing point and line features, which have demonstrated high performance. They integrated the two classes of features in a two component Gaussian Mixture Model (GMM) with preconditions and postconditions that are useful in understanding the behavior of corresponding features as shown in Table 2.2.

Their fusion algorithm is as the following: 1. New frame arrives and point features are detected. 2. Correspondence are found on the model. 3. The correct-match probability is computed from the correspondence score. 4. Pose is robustly computed for both modes, and the most probable mode is kept. 5. The new pose is used to initialize an edge tracker. 6. Features are back-projected on to the model. 7. The learned relationship between matching score and matching probability is updated. Points are analyzed using Scale-Invariant Feature Transform (SIFT) [39]. The maximum-likelihood is found not by iterative reweighted least squares (IRWLS) , which will not succeed because of local maxima, but by robust technique - Generalized Monte-Carlo based technique such as simulated annealing.

[40] provided a comprehensive study about the geometry and match of lines over multiple views. They introduced a geometric constraints in two-view matching - epipolar beam

Table 2.2: Point based tracking and edge based tracking.

| |
|---|
| **Point based tracking** |
| Preconditions |
| 3-D point cloud/model |
| Postconditions |
| Produces robust differential measurements |
| with approximately Gaussian posterior |
| Posterior measurement covariance is inaccurate |
| **Edge based tracking** |
| Preconditions |
| Geometric 3-D edge model |
| Accurate pose prior |
| Postconditions |
| Non-Gaussian posterior |
| Drift-free measurement |

as shown in Fig. 2.2. However, as the end points of line segments are not reliable, there is very limited usage of this property. Their short baseline matching between two views is: 1. Determine the curves in the second image which satisfy the epipolar constraints as putative matches. 2. For each putative match, determine the similarity score based on normalized cross-correlation - photometric constraint. 3. The matches are decided by a winner takes all scheme based a similarity scores. Each curve in the first image can only match one curve in the second. In implementation, the linear features are extracted by Canny edge detector. The similarity score for a pair of curve segments is computed as the average of the individual edgel correlation values.

[41] described an algorithm for matching line segments when camera motion is approximately known. They suggested that tracking usually involves two steps. First is a initialization process for a first estimate, and second is the use of Kalman filter to track frame by frame. [42] presented a robust model-based tracking algorithm based on EKF, and they used a test of mean criterion for initiating backtracking. [43] considered the propagation of uncertainty for matching junctions. The feature extraction consists of edge detection using an optimal filter [44], followed by a Hough transform line finder. The uncertainty of end point location of line segment is measured by a probability density

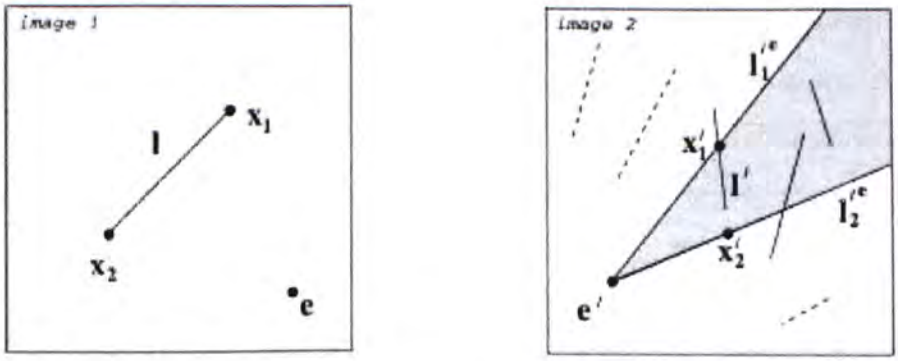



Figure 2.2: The epipolar geometry reduces the search space for line segments corresponding to **l**.

function. Then, the uncertainty propagates to epipolar lines. In matching junctions, the similarity is based on two junctions, junction topologies, and physical constraints.

In summarize, although tracking itself is by and large a solved problem, tracking lines using three-view geometry still deserve further exploration, and we aim at doing that.

### 2.2.2 Determining camera motion

In the early days, people estimated camera motion using point features from two images even when the relative position of the two cameras was unknown.

The algorithms fall into two classes: batch algorithms and recursive algorithms. In batch algorithms, one is linear algorithm with close form solution, the other is the optimization of nonlinear objective function. Recursive algorithms require models for motion and measurement, and the application of extended Kalman Filter.

**Linear algorithm, close form solution.** [30] proposed a closed-form solution to motion and structure from line correspondences. It requires rigid scene and a minimum of 13 lines over three perspective views. Essentially, their derivation is in Euclidean space, and the three *intermediate parameters* $(\mathbf{E}, \mathbf{F}, \mathbf{G})$ can be regarded as the "essential trifocal tensor", compared to essential matrix in two-view case. In the algorithm, $\mathbf{E}, \mathbf{F}, \mathbf{G}$ is solved first, and then two translation vectors can be determined up to scale, and finally, rotation matrices is determined. It can be used as a initial guess for an iterative process.

[30, 45] are some other linear algorithms.

**Factorization.** Tomasi and Kanade [46] is a pioneering work in this category, which

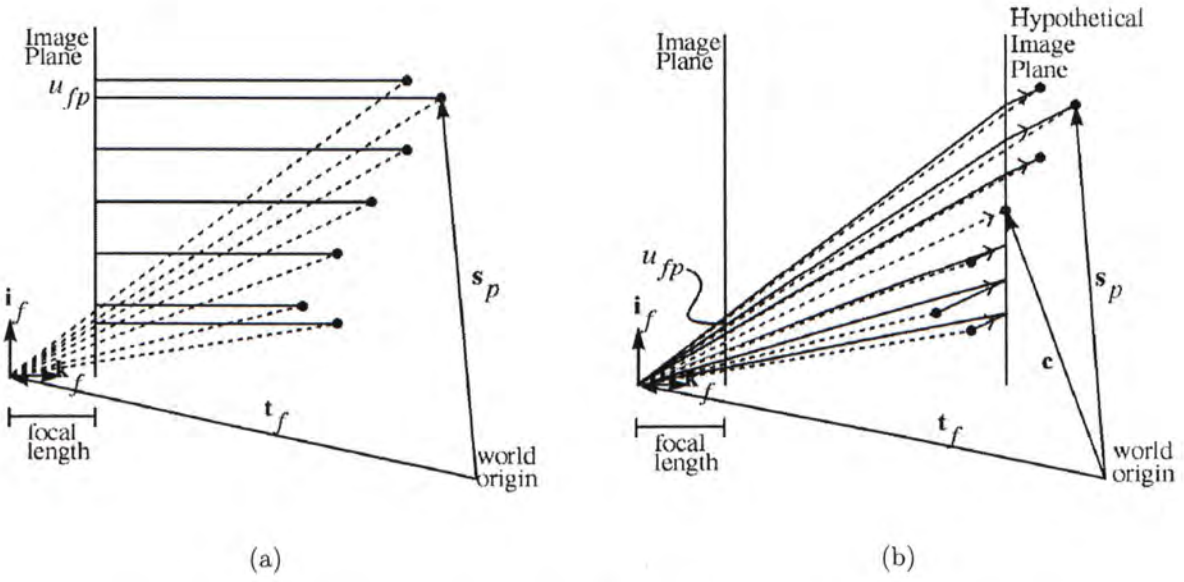



Figure 2.3: Orthogonal projection and paraperspective projection.

assumes orthographic projection as in Fig. 2.3(a). Given $P$ feature points over $F$ frames in an image stream, then horizontal feature coordinates are incorporated into an $F \times P$ matrix $\mathbf{U}$ with one row per frame and one column per feature point. Similarly, matrix $\mathbf{V}$ is built. $\mathbf{U}$ and $\mathbf{V}$ form a $2F \times P$ matrix $\mathbf{W}$, called *measurement matrix*. Then, it can be converted to *registered measurement matrix* $\tilde{\mathbf{W}}$, which is fully based on image measurements. In the algorithm, first is to compute the singular value decomposition of $\tilde{\mathbf{W}}$, then rotation matrix $\mathbf{R}$ and shape matrix $\mathbf{S}$ can be computed based on SVD of $\tilde{\mathbf{W}}$. In the presence of noise, a steepest descent minimization can be applied to a objective function of $\mathbf{W}$. The basis of the factorization method is the rank theorem - without noise, the registered measurement matrix $\tilde{\mathbf{W}}$ is at most of rank three. The advantage of factorization algorithm is that it does not require any prior knowledge about camera motion and object shape. But its orthographic projection model limits its applications to certain types of camera motions, because orthographic model contains no notion of the distance from the camera to the object.

[47] loosened the orthographic projection model in [46] to para-perspective projection model as in Fig. 2.3(b). They also provided an iterative minimization method for perspective refinement of para-perspective solution.

**Bundle adjustment.** Bundle adjustment in recovering motion and structure mini-

mizes the reprojection error between the estimated model and image measurements and it can produce jointly optimal 3-D structure and camera pose parameters. A variation of it is the interleaved bundle adjustment [48]. It separates the minimization process into two steps to speed up the algorithm.

[49] was a recent work applying bundle adjustment to estimate camera motion. It computes the trifocal tensor, triangulate the 3-D points using optimal backprojection, resection cameras and bundle adjust the reconstruction over all views in the sequence.

[50] gave a comprehensive survey about the theory and methods of bundle adjustment. Classically, bundle adjustment is formulated as nonlinear least squares problem, and the cost function is quadratic in feature reprojection errors. Modern bundle adjustment uses non-quadratic M-estimator-like probabilistic models to handle outliers more integrally, such as additional penalties to overcome overfitting. Bundle adjustment is essentially a parameter estimation problem. Nonlinear least squares (NLS) is the most basic parameter estimation method, and it minimizes the weighted Sum of Squared Error (SSE) cost function. For Gaussian noise, it provides the optimal solution; for non-Gaussian noise, Gauss-Markov theorem states that least squares gives the Best Linear Unbiased Estimator (BLUE) for linear model. Other models include robustified least squares (more realistic likelihood model), intensity-based methods (error model is based on intensity residuals), implicit models. Numerical optimization methods include Newton's method, Gauss-Newton method, Levenberg-Marquardt method. Constrained Problems can be solved by Sequential Quadratic Programming (SQP). Network optimization is applicable to solve complex parameters.

**Recursive, point feature, trifocal tensor.** A recent work integrated extended Kalman Filter and trifocal tensor in camera motion estimation [51]. Kanade-Lucas-Tmasi (KLT) tracker is used to extract point features and track them across views. In the initialization stage, eight-point algorithm using epipolar geometry plus the RANSAC algorithm provides a rough estimate of the relative pose between the first two frames. Next, in the EKF stage, constant velocity model is used as the motion model of the dynamic system, and measurement model is based on trifocal tensor constraints. It requires no prior information about the scene structure. In particular, it is the trifocal tensor point transfer function. They also proposed an additional structure updating step consisting of

$N$ identical EKFs for each model point.

In [52], the object motion model is separated as translational motion model and rotational motion model. The translational model assumes $n$th derivative of object position is constant, then, the model can be described by $3n$ parameters. The rotation matrix is written as the unit quaternion,

$$\mathbf{q} = (q_1, q_2, q_3, q_4)^T = (cos(\theta/2), \vec{\omega} sin(\theta/2)) \tag{2.2}$$

where $\vec{\omega}$ is the axis of rotation. The solution can be obtained with a filter such as an IEKF.

[53, 54, 55] are some other EKF-based algorithms.

To conclude, although camera motion estimation has been investigated for a couple of years, there is still a room to integrate feature tracking and motion determination together with the use of trifocal tensor and line features.

# Chapter 3

# Preliminaries on Three-View Geometry and Trifocal Tensor

This chapter is about the fundamentals of camera geometry, two-view geometry and three-view geometry. It begins with projective spaces and transformations, and summarizes the relationship of camera matrix, fundamental matrix and trifocal tensor. It also covers the trilinearities among three-view and line transfer. An linear algorithm for trifocal tensor based on line correspondences will be discussed as the foundation of later chapters.

## 3.1 Projective spaces $\mathcal{P}^3$ and transformations

*Projective space* is a set of elements constructed from a vector space consists of all vectors that are equal up to a non-zero scalar multiplication. A 3-D point in space is represented by a homogeneous 4-vector in $\mathcal{P}^3$ as

$$\mathbf{X} \sim [x_1, x_2, x_3, x_4]^T \tag{3.1}$$

while its Euclidean representation in $\mathbf{E}^3$ is $\mathbf{X} = [x_1/x_4, x_2/x_4, x_3/x_4]^T$. All points with $x_4 = 0$ in $\mathcal{P}^3$ refer to points at infinity.

**Projective representation of 3-D plane** is also a 4-vector $\mathbf{\Omega}$. Suppose a 3-D plane is given by $p_1x_1 + p_2x_2 + p_3x_3 + p_4x_4 = 0$, then the plane $\mathbf{\Omega}$ is expressed by homogeneous

vector

$$\Omega \sim [p_1, p_2, p_3, p_4]^T \tag{3.2}$$

**Projective representation of 3-D line** is covered in Chapter 4.

**Projective transformation of points.** A $4 \times 4$ matrix $\mathbf{H}$ captures the projective transformation on $\mathcal{P}^3$

$$\mathbf{X}' = \mathbf{H}\mathbf{X} \tag{3.3}$$

The hierarchy of transformations. The 15 degrees of freedom of projective transformation are accounted for as follows: 7 for similarity transformation, 5 for affine scalings, and 3 for projective part of the transformation. Meanwhile, 7 degrees of freedom are as 3 for rotation, 3 for translation, and 1 for isotropic scaling.

$$\mathbf{P}_{proj} = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{v}^T & v \end{bmatrix}, \quad \mathbf{P}_{aff} = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$$

$$\mathbf{P}_{sim} = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}, \quad \mathbf{P}_{Euclidean} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$$

## 3.2 The trifocal tensor

In this section, we discuss briefly about the geometry of single view, two-view and concentrate on three-view geometry and trifocal tensor.

**Camera Geometry.**

*Pinhole cameral model* captures the relation between 2-D image of an object taken by a camera and the 3-D geometry of this object shown in Fig. 3.1. With this, any 3-D point $\mathbf{X}$ can be projected to a image point $\mathbf{x}$ in Eqn. 3.4.

**Camera matrix.** Camera matrix is the projection matrix that captures how a 3-D point (expressed in homogeneous coordinates of $\mathcal{P}^3$) is related to its image projection (expressed in homogeneous coordinates of $\mathcal{P}^2$). It is the $3 \times 4$ matrix $\mathbf{P}$ in

$$\mathbf{x} = \mathbf{P}\mathbf{X} = \mathbf{K}\mathbf{M}\mathbf{X} \tag{3.4}$$

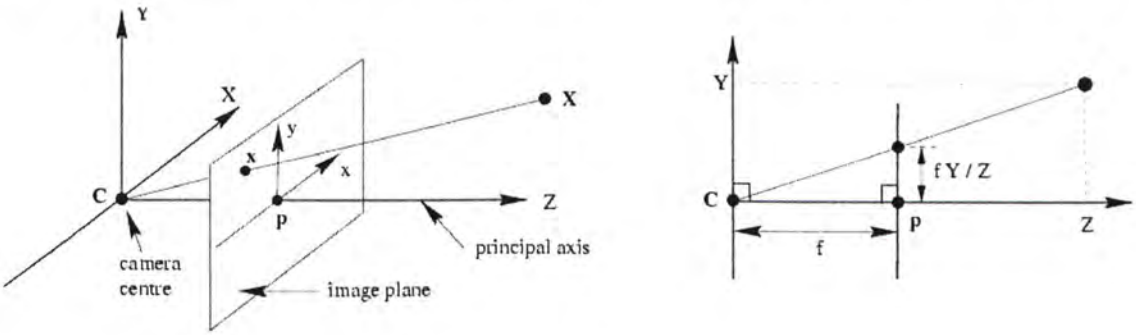



Figure 3.1: Camera Model.

$\mathbf{P}$ has 11 degree of freedom. $\mathbf{K}$ and $\mathbf{M}$ presents the intrinsic and extrinsic parameters matrix of the camera respectively.

$$\mathbf{K} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.5}$$

is called the *camera calibration matrix.* $f$ is the focal length, and $p_x$, $p_y$ are the coordinates of the principle point on image plane.

Assume that camera motion is rigid, $\mathbf{M}$ can be extended to

$$\mathbf{M} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \tag{3.6}$$

where $\mathbf{R}$, $\mathbf{t}$ are respectively the rotation matrix and translation vector that define the camera's position in space with respect to the reference coordinate frame. In particular, $\mathbf{R}$ can be expressed by the *Euler angles* $\alpha, \beta, \gamma$ as Eqn. (3.7):

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\alpha) & -sin(\alpha) \\ 0 & sin(\alpha) & cos(\alpha) \end{bmatrix} \begin{bmatrix} cos(\beta) & 0 & sin(\beta) \\ 0 & 1 & 0 \\ -sin(\beta) & 0 & cos(\beta) \end{bmatrix} \begin{bmatrix} cos(\gamma) & -sin(\gamma) & 0 \\ sin(\gamma) & cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.7}$$

**Camera-Motion.** $\mathbf{R}$ and $\mathbf{t}$ can be parameterized by $\alpha$, $\beta$, $\gamma$ and $x$,$y$,$z$ respectively. To compute camera-motion means to determine $\mathbf{R}$ and $\mathbf{t}$ (up to scale), or equivalently, determine $\alpha, \beta, \gamma, x, y, z$ directly from the image observations.

**Decomposition of the camera matrix.** Suppose camera matrix $\mathbf{P}$ is given, then the

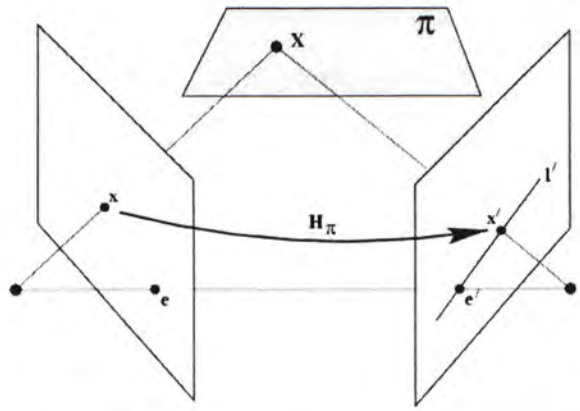



Figure 3.2: Two-view geometry and fundamental matrix.

camera center, orientation, and internal parameters $\mathbf{K}$ can be retrieved from $\mathbf{P}$. Camera center $\mathbf{C}$ satisfies $\mathbf{PC} = 0$. Applying SVD to $\mathbf{P}$ (see Appendix.), the right null-vector can be obtained. For the camera orientation $\mathbf{R}$ and internal parameters $\mathbf{K}$, from

$$\mathbf{P} = [\mathbf{M}| - \mathbf{M}\tilde{\mathbf{C}}] = \mathbf{K}[\mathbf{R}| - \mathbf{R}\tilde{\mathbf{C}}] \tag{3.8}$$

where $\tilde{\mathbf{C}}$ is the camera center in the world coordinate frame. Then, $\mathbf{K}$ and $\mathbf{R}$ can be found as $\mathbf{M} = \mathbf{KR}$ using the $\mathbf{RQ}$-decomposition, which decomposes a matrix to the product of an upper-triangular matrix and an orthogonal matrix.

**Camera line projection matrix.** Suppose the camera matrix that projects 3-D point to image point is the $3 \times 4$ matrix $\mathbf{P} \sim (\bar{\mathbf{P}}|\mathbf{p})$. Then the $3 \times 6$ line projection matrix [3, 4] that projects 3-D line (in the Plücker coordinates as a 6-vector) to image line (in the projective coordinates of $\mathcal{P}^2$ as a 3-vector) is

$$\tilde{\mathbf{P}} \sim [det(\bar{\mathbf{P}})\bar{\mathbf{P}}^{-\mathbf{T}}|[\mathbf{p}]_{\mathbf{x}}\bar{\mathbf{P}}]. \tag{3.9}$$

**Two-View Geometry.** In two-view geometry, fundamental matrix captures all constraints of epipolar geometry. Geometrically as in Fig. 3.2, *epipoles* $\mathbf{e}, \mathbf{e}'$ are the points of intersection of the line joining camera centers with the image plane. *Baseline* $\mathbf{CC}'$is the line joining two camera centers $\mathbf{C}$, $\mathbf{C}$'. *Epipolar plane* $\mathbf{H}_\pi$ is a plane containing the baseline. *Epipolar line* $\mathbf{l}$, $\mathbf{l}'$ is the intersection of epipolar plane with the image plane.

Suppose the point correspondence $\mathbf{x}$, $\mathbf{x}'$ satisfy

$$\mathbf{x}' = \mathbf{H}_\pi \mathbf{x} \tag{3.10}$$

we have

$$\mathbf{l}' = \mathbf{e}' \times \mathbf{x}' = [\mathbf{e}']_\times \mathbf{H}_\pi \mathbf{x} = \mathbf{F}\mathbf{x} \tag{3.11}$$

Fundamental matrix is defined as $\mathbf{F} = [\mathbf{e}']_\times \mathbf{H}_\pi$. The rank-2 matrix $\mathbf{F}$ maps from 2-D to 1-D family.

**Definition 3.1.** *Fundamental matrix* $\mathbf{F}$ *is the* $3 \times 3$ *rank-2 homogeneous matrix that satisfies*

$$\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0 \tag{3.12}$$

*for all point correspondences* $\mathbf{x}, \mathbf{x}'$ *in two images.*

**Properties.**

1. Epipolar lines $\mathbf{l}' = \mathbf{F}\mathbf{x}$, $\mathbf{l} = \mathbf{F}^T\mathbf{x}'$;
2. Epipole $\mathbf{F}\mathbf{e} = \mathbf{0}$, $\mathbf{e}'^T\mathbf{F} = \mathbf{0}$;
3. $\mathbf{F}$ has 7 degrees of freedom;
4. Pure translation, $\mathbf{F} = [\mathbf{e}']_\times$ has only 2 degrees of freedom; pure planar motion, $\mathbf{F}$ has 6 degrees of freedom.

$\mathbf{F}$ can also be calculated from epipole and camera matrices,

$$\mathbf{F} = [\mathbf{e}']_\times \mathbf{P}'\mathbf{P}^+ \tag{3.13}$$

where $\mathbf{P}^+$ is the pseudo-inverse of $\mathbf{P}$, i.e. $\mathbf{P}\mathbf{P}^+ = \mathbf{I}$.

Given fundamental matrix $\mathbf{F}$, projective ambiguity is the only ambiguity for camera matrices. Camera matrices may be chosen as $\mathbf{P} = [\mathbf{I}|\mathbf{0}]$ and $\mathbf{P}' = [[\mathbf{e}']_\times \mathbf{F}|\mathbf{e}']$. The general representation is given by

$$\mathbf{P} = [\mathbf{I}|\mathbf{0}], \quad \mathbf{P}' = [[\mathbf{e}']_\times \mathbf{F} + \mathbf{e}'\mathbf{v}^T|\lambda \mathbf{e}']$$

where $\mathbf{v}$ is any 3-vector and $\lambda$ is a non-zero scalar.

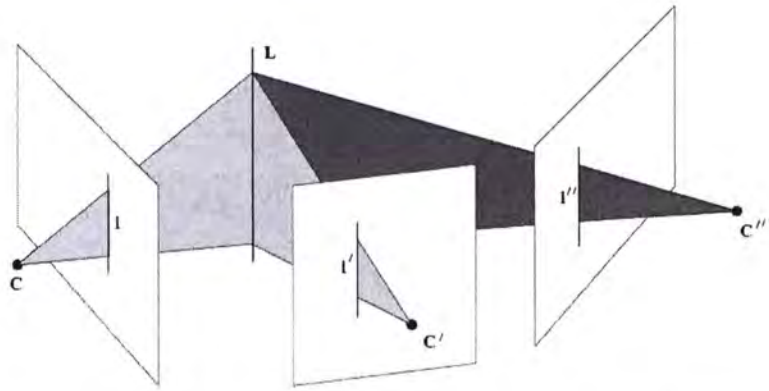



Figure 3.3: Three-view geometry and trifocal tensor.

**Three-View Geometry.** Compared to two-view geometry, where fundamental matrix captures all geometric relations across two views, trifocal tensor encodes the geometric relationship for three views.

In three-view, three back-projected lines meet in a single line in 3-D, and it provides constraints on lines. We denote the line correspondence as $l_i$, $l'_i$, $l''_i$ and camera matrices for three views as $\mathbf{P} = [\mathbf{I}|\mathbf{0}]$, $\mathbf{P}' = [\mathbf{A}|\mathbf{a}_4]$, $\mathbf{P}'' = [\mathbf{B}|\mathbf{b}_4]$, where $\mathbf{A}$, $\mathbf{B}$ are $3 \times 3$ matrices that represents infinite homographies from the first to the second and the third cameras respectively. Then, the incidence relation is

$$\mathbf{l}_i = \mathbf{l}'^T \mathbf{T}_i \mathbf{l}'' \tag{3.14}$$

where

$$\mathbf{T}_i = \mathbf{a}_i \mathbf{b}_4^T - \mathbf{a}_4 \mathbf{b}_i^T \tag{3.15}$$

where $\mathbf{a}_i, \mathbf{b}_i$ are columns in camera matrix $\mathbf{P}'$, $\mathbf{P}''$ respectively.

**Definition 3.2.** $\{\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3\}$ *is the matrix notation of trifocal tensor.*

$$\mathbf{l}^T = \mathbf{l}'^T \begin{bmatrix} \mathbf{T}_1 & \mathbf{T}_2 & \mathbf{T}_3 \end{bmatrix} \mathbf{l}'' \tag{3.16}$$

*Note: trifocal tensor only involves image coordinates, not 3-D structure.*

Based on trifocal tensor, line-line-line, point-line-line, point-line-point, point-point-point relations can be derived [7].

**Tensor slices.**

The properties of tensor slices can be found in [56]. Here only a brief summary is stated. Three matrices $\mathbf{E}_\delta$, $\mathbf{W}_\delta$, $\mathbf{G}_\delta$ are obtained from the dot products of the tensor with some arbitrary vector $\delta$. Three types of dot products produce three types of families: $\mathbf{E}_\delta$, $\mathbf{W}_\delta$ are *homography* matrices where $\mathbf{E}_\delta$ is a collineation mapping from view-1 to view-2 via the plane determined by camera center $C''$ of the third camera and the line $\delta$ in view-3; likewise, $\mathbf{W}_\delta$ is the homography mapping from view-1 to view-3 via the plane formed by camera center C' of the second camera and the line $\delta$ in view-2. Matrix $\mathbf{G}_\delta$ is a *correlation* matrix mapping the dual image plane of view-2 to points in view-3.

When $\delta$ is chosen to be $(1,0,0),(0,1,0),(0,0,1)$, $\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3$ are call *standard homography slices* of the tensor. Similarly, $\mathbf{G}_1, \mathbf{G}_2, \mathbf{G}_3$ are called *standard correlation slices* of the tensor.

**Admissibility constraints of the trifocal tensor.**

The tensor with 27 coefficients while the number of degrees of freedom is only 18 (It can be accounted as the three camera matrices (33 parameters) minus the degrees of freedom associated with any arbitrary projective frame (15)). Therefore, there must be 9 algebraic constraints for trifocal tensor. There is no other constraint for trifocal tensor. We used the correlation contraction form for the tensor. Three groups of admissibility constraints are:

1. (7 constraints) $Rank(\delta^i \mathcal{T}_i^{jk}) = 2$ for all $\delta$. $\mathbf{G}_1, \mathbf{G}_2, \mathbf{G}_3$ are of rank-2.
2. (1 constraint) $Rank(null(\mathbf{G}_1), null(\mathbf{G}_2), null(\mathbf{G}_3)) = 2$.
3. (1 constraint) $Rank(null(\mathbf{G}_1^T), null(\mathbf{G}_2^T), null(\mathbf{G}_3^T)) = 2$.

**Camera matrix, fundamental matrix and trifocal tensor.** Suppose $\mathbf{P}$, $\mathbf{F}$ and $\mathbf{T}$ denote camera matrix, fundamental matrix and trifocal tensor respectively.

**Calculating T from P.** Given camera matrices for three views $\mathbf{P} = [\mathbf{I}|\mathbf{0}]$, $\mathbf{P}' = [\mathbf{A}|\mathbf{a}_4]$, $\mathbf{P}'' = [\mathbf{B}|\mathbf{b}_4]$, in which $3 \times 3$ matrices $\mathbf{A}$ and $\mathbf{B}$ represents infinite homographies from the first to the second and the third cameras respectively.

Then the matrix notation of trifocal tensor $\{\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3\}$ can be written as

$$\mathbf{T}_i = \mathbf{a}_i \mathbf{b}_4^T - \mathbf{a}_4 \mathbf{b}_i^T \tag{3.17}$$

**Extracting F from T.** $3 \times 3$ matrices $\mathbf{F}_{21}, \mathbf{F}_{31}$ are fundamental matrices between

the first and the second and third views respectively, then

$$\begin{aligned}\mathbf{F}_{21} &= [\mathbf{e}']_\times[\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3]\mathbf{e}'' \\ \mathbf{F}_{31} &= [\mathbf{e}'']_\times[\mathbf{T}_1^T, \mathbf{T}_2^T, \mathbf{T}_3^T]\mathbf{e}'\end{aligned} \tag{3.18}$$

**Extracting P from T.** Camera matrices can be retrieved from trifocal tensor only up to projective ambiguity. The first camera can be chosen arbitrarily as $\mathbf{P} = [\mathbf{I}|\mathbf{0}]$. Then

$$\mathbf{P}' = [[\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3]\mathbf{e}''|\mathbf{e}'] \tag{3.19}$$

Taking the projective ambiguity in to account, we have

$$\mathbf{P}' = [[\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3]\mathbf{e}'' + \mathbf{e}'\mathbf{v}^T|\lambda\mathbf{e}'] \tag{3.20}$$

for some $3 \times 1$ vector $\mathbf{v}$ and scalar $\lambda$. Obviously the fundamental matrix between camera pair $\mathbf{P}, \mathbf{P}'$ is $\mathbf{F}_{21}$. Suppose $\mathbf{P}'$ is chosen as Eqn. (3.19), then the projective world frame is fixed so that $\mathbf{P}''$ can be defined uniquely up to scale,

$$\mathbf{P}'' = [(\mathbf{e}''\mathbf{e}''^T - \mathbf{I})[\mathbf{T}_1^T, \mathbf{T}_2^T, \mathbf{T}_3^T]\mathbf{e}'|\mathbf{e}''] \tag{3.21}$$

*NOTE: Details can be found in [7].*

**Trilinearities and transfer.** Line transfer from corresponding lines in the second and third views to the first is given by [7].

$$l_i = l'_j l''_k \mathcal{T}_i^{jk} \tag{3.22}$$

where $l_i, l'_j, l''_k = 1, 2, 3$ are image coordinates of $\mathbf{l}$, $\mathbf{l}'$, $\mathbf{l}''$ respectively.

## 3.3 Computation of the trifocal tensor-Normalized linear algorithm.

To calculate trifocal tensor from line correspondences in three-view, it is easy to verify the line-line-line correspondence from line transfer,

$$(\mathbf{l}_r\epsilon)\mathbf{l}'_j\mathbf{l}''_k\mathcal{T}_i^{jk} = 0^s \tag{3.23}$$

The equation contains three scalar equations, but only two of them are independent. Given $N \geq 13$ image line correspondences across three views, the trifocal tensor can be computed as follows.

**Algorithm.**

1). Transform lines with respect to the center of the image as its origin.

2). Expand Eqn. (3.23) to two scalar equations and form a set of linear equations of the form $\mathbf{At} = \mathbf{0}$.

$$(\mathbf{l}'^T \begin{bmatrix} \mathbf{T}_1 & \mathbf{T}_2 & \mathbf{T}_3 \end{bmatrix} l'')[\mathbf{l}]_\times = \mathbf{0}^T \tag{3.24}$$

where the set of three matrices $\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3$ constitutes the trifocal tensor in matrix notation. It is expanded as,

$$(\begin{bmatrix} \mathbf{l}'_1 & \mathbf{l}'_2 & \mathbf{l}'_3 \end{bmatrix} \begin{bmatrix} \mathbf{T}_1 & \mathbf{T}_2 & \mathbf{T}_3 \end{bmatrix} \begin{bmatrix} \mathbf{l}''_1 & \mathbf{l}''_2 & \mathbf{l}''_3 \end{bmatrix}) \begin{bmatrix} 0 & -\mathbf{l}_3 & \mathbf{l}_2 \\ \mathbf{l}_3 & 0 & -\mathbf{l}_1 \\ -\mathbf{l}_2 & \mathbf{l}_1 & 0 \end{bmatrix} = \mathbf{0}^T \tag{3.25}$$

It can then be written as,

$$\begin{aligned}
\mathbf{l}'^T\mathbf{T}_1\mathbf{l}'' &= (l'_1T_1^{11} + l'_2T_1^{21} + l'_3T_1^{31})l''_1 + (l'_1T_1^{12} + l'_2T_1^{22} + l'_3T_1^{32})l''_2 + (l'_1T_1^{13} + l'_2T_1^{23} + l'_3T_1^{33})l''_3 \\
\mathbf{l}'^T\mathbf{T}_2\mathbf{l}'' &= (l'_1T_2^{11} + l'_2T_2^{21} + l'_3T_2^{31})l''_1 + (l'_1T_2^{12} + l'_2T_2^{22} + l'_3T_2^{32})l''_2 + (l'_1T_2^{13} + l'_2T_2^{23} + l'_3T_2^{33})l''_3 \\
\mathbf{l}'^T\mathbf{T}_3\mathbf{l}'' &= (l'_1T_3^{11} + l'_2T_3^{21} + l'_3T_3^{31})l''_1 + (l'_1T_3^{12} + l'_2T_3^{22} + l'_3T_3^{32})l''_2 + (l'_1T_3^{13} + l'_2T_3^{23} + l'_3T_3^{33})l''_3
\end{aligned} \tag{3.26}$$

$\mathbf{l}_3, \mathbf{l}'_3, \mathbf{l}''_3$ can be normalized to 1. The first two columns of $[\mathbf{l}]_\times$ are selected as two independent scalar equations,

$$l_3 \mathbf{l}'^T \mathbf{T}_2 \mathbf{l}'' - l_2 \mathbf{l}'^T \mathbf{T}_3 \mathbf{l}'' = 0$$
$$-l_3 \mathbf{l}'^T \mathbf{T}_1 \mathbf{l}'' + l_1 \mathbf{l}'^T \mathbf{T}_3 \mathbf{l}'' = 0 \tag{3.27}$$

Using these two equations, $N$ ($N \geq 13$) line correspondences can be packed into $2 \times N$ linear equations as the form $\mathbf{At} = \mathbf{0}$, where

$$\mathbf{t} = \begin{bmatrix} \mathbf{T}_1^{11} & \mathbf{T}_1^{12} & \mathbf{T}_1^{13} & \dots & \mathbf{T}_3^{33} \end{bmatrix} \tag{3.28}$$

3). Use least-squares method to get the solution. Given $\mathbf{At} = \mathbf{0}$, the solution $\mathbf{t}$ that minimizes $\|\mathbf{At}\|$ subject to $\|\mathbf{t}\| = 1$ is the last column of $\mathbf{V}$, where $\mathbf{A} = \mathbf{UDV}^T$ is the SVD of $\mathbf{A}$.

4). Reshape the solution to trifocal tensor form.

Besides, algebraic minimization algorithm and geometric minimization algorithm are two other kinds of methods for trifocal tensor estimation in real applications, which we will not discuss here.

# Chapter 4

# Linear Line Structures

This chapter will describe the model of line space and the classification of linear line structures, including linear line space (line pencil, point star, ruled plane) and linear ruled surface, linear line congruence, linear line complex.

## 4.1 Models of line space

A line is defined by joining two points or intersecting two planes. Unlike point and plane, line can not be represented using homogenous 4-vector, since it has 4 degree of freedom that corresponds to a homogenous 5-vector. Some of the line representations are listed.

1. **Orthogonal plane coordinates.** Suppose a line intersects two arbitrary orthogonal planes $\alpha, \beta$ at $\mathbf{X}_1$, $\mathbf{X}_2$ respectively. Two additional plane coordinates can be established with these two planes to index $\mathbf{X}_1$ and $\mathbf{X}_2$ as $(x_1, y_1), (x_2, y_2)$. Generally, a line is uniquely determined by four parameters $x_1, y_1, x_2, y_2$, giving four degree of freedom. This representation has the disadvantage of changing the degree under projective representation, but it still helps in classifying line structure that will be shown later.

2. **Plücker line coordinates**

**First Plücker line coordinates.** This representation was first introduced by Plücker in 1846 [57]. Suppose $x, y, z$ are cartesian coordinates, a line can be represented by the following equations

$$x = rz + \varrho, y = sz + \rho \tag{4.1}$$

where $r, s, \varrho, \rho$ are four parameters of the line. However, if a projective transformation is applied, the result can not be precisely stated, as the numerators and denominators are linear functions of the original line. The degree of any equation between $r, s, \varrho, \rho$ would be changed by projective transformation. It can be avoided by a new formulation below.

**Plücker matrices.** Line is represented by a $4 \times 4$ skew-symmetric homogeneous matrix. Suppose there are two points $\mathbf{A}, \mathbf{B}$, line $\mathbf{L}$ is

$$\mathbf{L} = \mathbf{A}\mathbf{B}^T - \mathbf{B}\mathbf{A}^T \tag{4.2}$$

**Properties.**

1. $\mathbf{L}$ has rank 2.
2. Line $\mathbf{L}$ has 4 degrees of freedom.

**Dual Plücker representation $\mathbf{L}^*$.** It is formed by the intersection of two planes $\mathbf{P}, \mathbf{Q}$,

$$\mathbf{L}^* = \mathbf{P}\mathbf{Q}^T - \mathbf{Q}\mathbf{P}^T \tag{4.3}$$

**Properties.** Similar to $\mathbf{L}$.

Join of two points and intersection of two planes form the self-dual for line-geometry.

**Plücker line coordinates.** We adopt the Plücker line coordinates for representing lines because not only do they capture a concise description of any 3-D line but also allow the relationship between the line and its image projection to be related by a rather simple projection matrix.

Given two 3-D points $\mathbf{M^T} \sim (\bar{\mathbf{M}}^\mathbf{T}|m)$ and $\mathbf{N^T} \sim (\bar{\mathbf{N}}^\mathbf{T}|n)$ of a 3-D line to be represented, the Plücker line coordinates of the line are the homogeneous 6-vector $\mathbf{L} \sim (\mathbf{a^T}|\mathbf{b^T})^T$, where $\mathbf{a} = \bar{\mathbf{M}} \times \bar{\mathbf{N}}$ refers to the surface normal of the plane defined by the described 3-D line and the origin of the coordinate frame, $\mathbf{b} = m\bar{\mathbf{N}} - n\bar{\mathbf{M}}$ refers to the direction of the 3-D line in that plane, and $\mathbf{a}$ and $\mathbf{b}$ in $\mathbf{L}$ have their respective norms scaled relative to each other so that if $\mathbf{b}$ is scaled to be a unit vector the norm of $\mathbf{a}$ is the distance of the 3-D line from the coordinate frame origin. The Plücker coordinates also satisfy the Plücker equality which is given by

$$\Omega_{\mathbf{L},\mathbf{L}} = \mathcal{C}\{\mathbf{L}\} = \mathbf{a}^T\mathbf{b} = 0 \tag{4.4}$$

Equivalently, the Plücker coordinates can also be obtained as six particular coefficients of the following $4 \times 4$ skew-symmetric Plücker matrix $\bar{\mathbf{L}}$:

$$\bar{\mathbf{L}} = \mathbf{M}\mathbf{N}^T - \mathbf{N}\mathbf{M}^T$$

with the six coefficients being

$$\mathcal{L} \sim \{l_{23}, -l_{13}, l_{12}, -l_{14}, l_{42}, -l_{34}\}$$

Thus a general 3-D line has four degrees of freedom: two captured by the direction of $\mathbf{a}$ which is the normal vector to the plane $\mathbf{\Pi}$ defined by the 3-D line and the origin, one on vector $\mathbf{a}$'s norm - the distance of the 3-D line from the origin, and one on the direction of vector $\mathbf{b}$ as the orientation of the 3-D line on plane $\mathbf{\Pi}$.

With the Plücker representation, a line in 3-D can be represented as a point of $\mathbf{P^4}$ [5], or a point that lies on a non-singular quadric $\mathbf{\Omega}$ (Eqn. (4.4)) in $\mathbf{P^5}$. The latter representation is called the *Klein model* [58].

**Definition 4.1.** *The Klein mapping assigns a line* $\mathbf{L}$ *of* $\mathcal{P}^3$ *to the point* $\{l_{23}, -l_{13}, l_{12}, -l_{14}, l_{42}, -l_{34}\}$ *of* $\mathcal{P}^5$*, where* $\{l_{23}, -l_{13}, l_{12}, -l_{14}, l_{42}, -l_{34}\}$ *are the line's Plücker coordinates.*

**Theorem 4.1.** *The lines of projective* 3*-space are in one-to-one correspondence with the points of the quadric* $\mathbf{\Omega} = 0$ *of projective* 5*-space.*

There is advantage of using the latter representation because with it the relationship between a line and its image projection can be captured by a simple pre-multiplication of a projection matrix.

## 4.2 Line structures

No matter which representation is adopted, the lines of four degrees of freedom form a quadruply infinite system [59], which are line complex, line congruence, ruled surfaces.

As discussed earlier, general 3-D lines have four degrees of freedom, and they can be regarded as forming a quadruple infinite system $\mathcal{P}^4$ in the 3-D line space [59]. Overally speaking, specific family of lines in the 4-dimensional line space that possess one degree

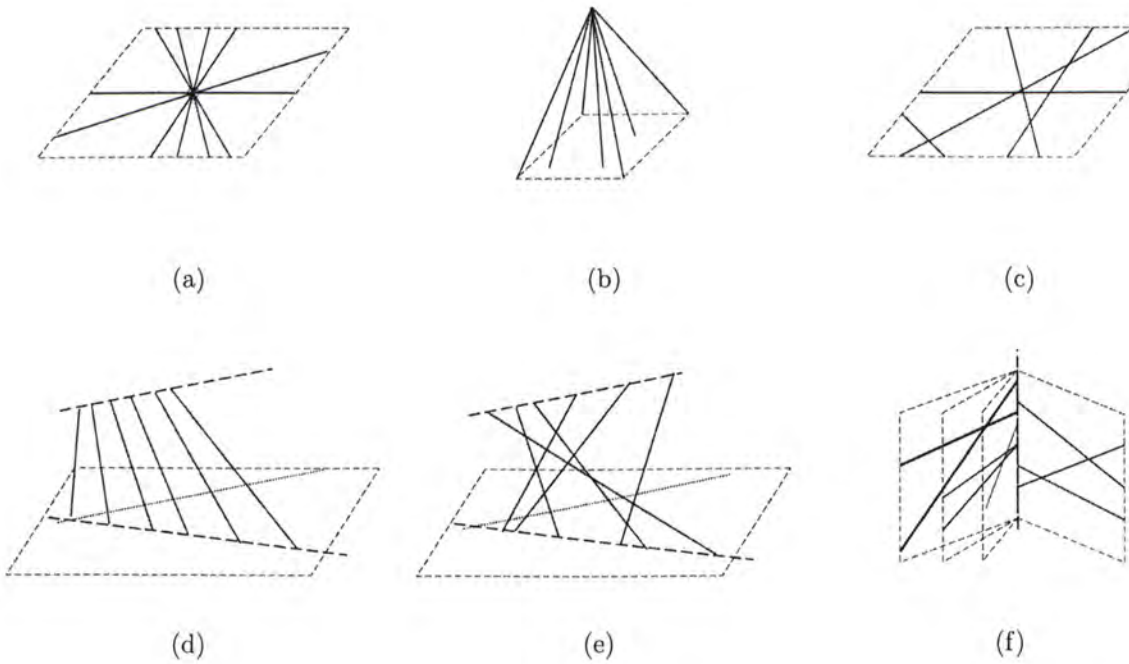



Figure 4.1: Illustration of the various linear line structures. (a)-(c): linear line spaces including line pencil, point-star and ruled plane; (d)-(e): linear ruled surface, linear line congruence and linear line complex.

of freedom, two degrees of freedom, and three degrees of freedom are respectively called ruled surface, line congruence, and line complex in the terminologies of classical line geometry. In particular, if the constraints that reduce the degrees of freedom (DoFs) of the line set are all linear, the above structures are called *linear ruled surface*, *linear line congruence*, and *linear line complex* respectively, and as a collection they are called **linear line structures**. In this work only line structures with linear constraint(s) are investigated; in other words we only consider linear line structures. On top of the above, if the Plücker equality that governs the 6 line coordinates happens to be linear (for example, in the case where the $\mathbf{a}$ (or $\mathbf{b}$) in the Plücker coordinates $\mathbf{L} \sim [\mathbf{a}^\mathbf{T}, \mathbf{b}^\mathbf{T}]^\mathbf{T}$ of every line in a line set is a constant vector), then the linear line structure is reduced to the **linear line space**, which has linear ruled surface reduced to the sub-structure *line pencil*, and linear line congruence reduced to *point-star* or *ruled plane*.

Below we describe the respective line structures in more detail.

### 4.2.1 Linear line space

The only linear line spaces in 3-D are line pencil (with 1 arbitrary DoF), point-star , and ruled plane (both with 2 arbitrary DoFs) [59], which are visually illustrated in Fig. 4.1(a),(b),(c).

**Lemma 4.1.** *The only one-dimensional subspaces contained in the Klein quadric* $\mathbf{M}_2^4$ *is line pencil.*

**Lemma 4.2.** *The Klein quadric contains two two-dimensional subspaces, which corresponds to point-star, and ruled plane.*

**Remark 4.1.** *A regular quadric in* $\mathcal{P}^5$ *can not contain three-dimensional subspaces, for Plücker equality can not appear to be linear under 3-DoF.*

If the Plücker equality has to appear linear, there is no line space with 3 DoFs [58], for a regular quadric in $\mathcal{P}^5$ cannot contain three-dimensional subspaces. Point-star and ruled plane are dual to each other in the linear space (because they are both specializations of the 2-DoF linear line congruence when the Plücker equality happens to be linear), and the line pencil is self-dual. Line pencil can be obtained as the intersection of point-star and ruled plane.

### 4.2.2 Ruled surface

*Ruled surface* is a family of lines that has 1-DoF generated by a straight line moving through 3-D space by a single parameter. The straight lines in the ruling are called *generators* of the surface. The *order* of the surface is the number of lines that meet a general line.

**Linear ruled surface.** *Linear ruled surface* (Fig. 4.1(d)) is a line system constrained by three linear equations. The *order* $n$ of the line system is the number of lines in the system that meet at a general line [59].

In Plücker coordinates with the norm scaled to be 1, a line $\mathbf{L} \sim (\mathbf{a}^T|\mathbf{b}^T)^T$ in a ruled surface has its coordinates constrained by three linear constraints plus the Plücker equality,

as expressed by the following:

$$\begin{aligned} \mathbf{a}_1^T\mathbf{L} &= \mathbf{a}_2^T\mathbf{L} = \mathbf{a}_3^T\mathbf{L} = 0; \\ \mathbf{a}^T\mathbf{b} &= 0;\ |\mathbf{L}| = 1 \end{aligned} \tag{4.5}$$

**Some analytical ruled surfaces.** Typical ruled surfaces include ruled hyperboloid, quartic ruled surface, sextic ruled surface. twisted cubic, helicoid, ruled cone.

### 4.2.3 Line congruence

*Line congruence* is defined by two independent equations, as the intersection of two complexes. The *order* of line congruence is the number of lines that meet at a general point, and the *class* is the number of lines that lie in a general plane [59].

**Linear line congruence.** *Linear line congruence* (Fig. 4.1(e)) is line congruence with the two constraints being linear. It is a $(1,1)$ congruence that is the intersection of two distinct linear line complexes $L_1$, $L_2$. In Plücker coordinates with the norm scaled to be 1, a line $\mathbf{L} \sim (\mathbf{a}^T|\mathbf{b}^T)^T$ in a linear line congruence is bounded by the following:

$$\begin{aligned} \mathbf{a}_1^T\mathbf{L} &= \mathbf{a}_2^T\mathbf{L} = 0; \\ \mathbf{a}^T\mathbf{b} &= 0;\ |\mathbf{L}| = 1 \end{aligned} \tag{4.6}$$

**Quadric congruence.** *Quadric congruence* is a $(2,2)$ congruence that is the intersection of a linear complex and a quadric complex.

### 4.2.4 Line complex

In Plücker coordinates a line $\mathbf{L}$ in a *line complex* is constrained by a single equation:

$$f(l_{12}, l_{13}, l_{14}, l_{23}, l_{42}, l_{34}) = 0$$

leaving only 3-DoF for any line in the line system. The *order* of the line complex is defined by the order of the constraint $f(\cdot) = 0$ over the six line coordinates. However, we are reminded that the quadric constraint (Plücker equality), is always satisfied, which makes

the line geometry more complicated than its counterpart of points.

**Linear line complex.** *Linear line complex* (Fig. 4.1(f)) is line complex with the governing constraint being linear. It is the three-dimensional linear manifold of lines defined by one linear equation. In Plücker coordinates with the last coordinate scaled to be 1, a line $\mathbf{L} \sim (\mathbf{a}^T|\mathbf{b}^T)^T$ in a linear line complex is:

$$\begin{aligned} \mathbf{a}_1^T\mathbf{L} &= 0; \\ \mathbf{a}^T\mathbf{b} &= 0; \ |\mathbf{L}| = 1. \end{aligned} \tag{4.7}$$

If the 6-vector $\mathbf{a}_1$ satisfies the Plücker equality ($\mathbf{\Omega}_{\mathbf{a}_1,\mathbf{a}_1} = 0$), meaning that $\mathbf{a}_1$ is the Plücker coordinates of a 3-D line, then all lines of the linear line complex (LLC) meet the line $\mathbf{a}_1$ as its axis, and the LLC is called special or singular. If the 6-vector $\mathbf{a}_1$ does not satisfy the Plücker equality ($\mathbf{\Omega}_{\mathbf{a}_1,\mathbf{a}_1} \neq 0$), the LLC is called general or regular. The degeneracy study of Stein and Shashua in [28] is about special linear line complex.

**Quadratic line complex.** If $f(\cdot) = 0$ is second order, the line complex is quadratic.

# Chapter 5

# Critical Configurations of Three Views Revealed by Line Correspondences

Degeneracy can come from degenerate motion (straight line motion, planar motion) and degenerate structure (quadrics). Our perspective is to investigate the degenerate line structure for trifocal tensor estimation. Next, it describes the empirical observation regarding the line transfer under degenerate structure. Finally, rank properties of nonlinear line structures are proposed.

## 5.1 Two-view degeneracy

As line correspondences do not provide any constraint for two-view geometry, only point feature will be summarized in brief.

**Point feature.** Two-view ambiguity is summarized in [15]. Critical configuration is defined by two projectively inequivalent configurations $\mathbf{P}^i, \mathbf{P}_j$ and $\mathbf{Q}^i, \mathbf{Q}_j$ such that

$$\mathbf{P}^i\mathbf{P}_j = \mathbf{Q}^i\mathbf{Q}_j \tag{5.1}$$

$\mathbf{P}^i, \mathbf{P}_j$ and $\mathbf{Q}^i, \mathbf{Q}_j$ are called conjugate configurations. It is well-known that critical configurations of two cameras occur when two cameras and all 3-D points lie on a ruled

quadric.

**Theorem 5.1.** *Suppose two pairs of cameras* $(\mathbf{P}^0, \mathbf{P}^1)$, $(\mathbf{Q}^0, \mathbf{Q}^1)$ *with inequivalent fundamental matrix* $\mathbf{F}_P^{10}, \mathbf{F}_Q^{10}$. *Define two quadrics*

$$\mathbf{S}_P = \mathbf{P}^{1T}\mathbf{F}_Q^{10}\mathbf{P}^0, \mathbf{S}_Q = \mathbf{Q}^{1T}\mathbf{F}_P^{10}\mathbf{Q}^0 \tag{5.2}$$

*1. If* $\mathbf{P}$ *and* $\mathbf{Q}$ *satisfy* $\mathbf{P}^0\mathbf{P} = \mathbf{Q}^0\mathbf{Q}$ *and* $\mathbf{P}^1\mathbf{P} = \mathbf{Q}^1\mathbf{Q}$, *then* $\mathbf{P}$ *lies on the quadric* $\mathbf{S}_P$ *and* $\mathbf{Q}$ *lies on* $\mathbf{S}_Q$.

*2. If* $\mathbf{P}$ *lies on* $\mathbf{S}_P$, *then there exists critical configuration* $\mathbf{Q}$ *lies on* $\mathbf{S}_Q$

*3. If* $\mathbf{e}_Q^{01}$ *is the epipole defined by* $\mathbf{F}_Q^{10}\mathbf{e}_Q^{01} = \mathbf{0}$, *then the ray passing through* $\mathbf{C}_P^0$ *consists of points* $\mathbf{X}$ *such that* $\mathbf{P}^0\mathbf{X} = \mathbf{e}_Q^{01} = \mathbf{Q}^0\mathbf{C}_Q^1$ *lies on the ruled quadric* $\mathbf{S}_P$.

**Proof.**

1.

$$\mathbf{P}^T\mathbf{S}_P\mathbf{P} = \mathbf{P}^T\mathbf{P}^{1T}\mathbf{F}_Q^{10}\mathbf{P}^0\mathbf{P} = \mathbf{Q}^T(\mathbf{Q}^{1T}\mathbf{F}_Q^{10}\mathbf{Q}^0)\mathbf{Q} = 0 \tag{5.3}$$

Thus, $\mathbf{P}$ lies on $\mathbf{S}_P$. Likewise, $\mathbf{Q}$ lies on $\mathbf{S}_Q$

3. For point $\mathbf{X}$ such that $\mathbf{e}_Q^{01} = \mathbf{P}^0\mathbf{X}$, then $\mathbf{S}_P\mathbf{X} = \mathbf{P}^{1T}\mathbf{F}_Q^{10}\mathbf{P}^0\mathbf{X} = \mathbf{P}^{1T}\mathbf{F}_Q^{10}\mathbf{e}_Q^{10} = 0$. $\mathbf{X}$ lies on $\mathbf{S}_P$.

## 5.2 Three-view degeneracy

In three views, both point feature and line feature can be used. We only focus on the cases where line structures are used. As we have seen that the critical configurations using point correspondences are well-known; however, only a few attention has been drawn to the critical sets of lines, see [28] and [27]. There is still much room to explore in this area.

### 5.2.1 Introduction

**Line representation.** In this work, we use the Plücker coordinates to represent lines, see Chapter 3.

In this work we examine how the determination of the trifocal tensor from an image triplet would be affected should the observed line correspondences over the three views

be from particular line systems in 3-D. In particular, we examine how the rank of the tensor estimation matrix $\mathbf{A}$ – the matrix that leads to the determination of the trifocal tensor – would be affected, and how distant each of the particular line systems is from the necessary system that allows the trifocal tensor to be uniquely determined.

**Line Projection.** For any 3-D line $\mathbf{L}$ visible in three given views, its image projections $\mathbf{l}, \mathbf{l}', \mathbf{l}''$ in the three views are related to $\mathbf{L}$ in Plücker coordinates by the following:

$$\bar{\mathbf{K}} = \begin{bmatrix} \mathbf{l} \\ \mathbf{l}' \\ \mathbf{l}'' \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{P}} \\ \tilde{\mathbf{P}}' \\ \tilde{\mathbf{P}}'' \end{bmatrix} \mathbf{L} \tag{5.4}$$

where $\tilde{\mathbf{P}}, \tilde{\mathbf{P}}', \tilde{\mathbf{P}}''$ are the three cameras' line projection matrices.

In the above, $\mathbf{L}$ can be taken to include the coordinates of all $N$ 3-D lines visible in the image data. In that case (which hereafter we assume), $\mathbf{L}$ will be a $6 \times N$ matrix, $\mathbf{l}, \mathbf{l}', \mathbf{l}''$ will be each a $3 \times N$ matrix, and the entire left side of the above equation, collectively referred to as the $\bar{\mathbf{K}}$ matrix, is about a $9 \times N$ matrix that captures all image line observations in all views of all $N$ lines. As we shall present in detail below, this $9 \times N$ observation matrix $\bar{\mathbf{K}}$ is all it takes to define the estimation matrix $\mathbf{A}$ – the key to the determination of the trifocal tensor.

In the rest of paper we also require to access the individual rows of the image line observations $\mathbf{l}$ or $\mathbf{l}'$ or $\mathbf{l}''$ that are visible in View 1, 2, and 3 respectively. We shall define the $N$-vector $\mathbf{l}_i$ as the $i^{th}$ row of $\mathbf{l}$, and scalar $l_{ij}$ as the $j$th entry of that row $\mathbf{l}_i$ of $\mathbf{l}$. In other words, $\mathbf{l} = [\mathbf{l}_1{}^T, \mathbf{l}_2{}^T, \mathbf{l}_3{}^T]^T$, and $\mathbf{l}_i = [l_{ij}]$ for all $i = 1, 2, 3$. Similarly, we have $N$-vector $\mathbf{l}'_i$ and scalar $l'_{ij}$ for image line observations $\mathbf{l}'$ in the second view, and $N$-vector $\mathbf{l}''_i$ and scalar $l''_{ij}$ for image line observations $\mathbf{l}''$ in the third view.

**Tensor Estimation.** Each line correspondence over three views gives two independent linear equations for the $3 \times 3 \times 3$ trifocal tensor [7] whose entries can be listed out in a column fashion as a $27 \times 1$ vector $\mathbf{t}$. Given $N$ line correspondences, we have $2N$ linear equations for $\mathbf{t}$, which can be collectively captured by a $2N \times 27$ matrix $\mathbf{A}$. The tensor-equivalent vector $\mathbf{t}$ and the estimation matrix $\mathbf{A}$ are related by:

$$\mathbf{At} = \mathbf{0} \tag{5.5}$$

where

$$\mathbf{A} = \begin{bmatrix} & \vdots & \vdots \\ \mathbf{0}_{N\times 9} & l_{3i}\mathbf{B}_i & -l_{2i}\mathbf{B}_i \\ & \vdots & \vdots \\ \vdots & & \vdots \\ -l_{3i}\mathbf{B}_i & \mathbf{0}_{N\times 9} & l_{1i}\mathbf{B}_i \\ \vdots & & \vdots \end{bmatrix} \tag{5.6}$$

and $\mathbf{B}_i$ is the $i$th row ($i = 1, 2, \ldots, N$) of the $N \times 9$ matrix $\mathbf{B}$ defined by the image line observations in Eqn. (5.7).

$$\mathbf{B} = \begin{bmatrix} & \vdots & \\ l'_{1i}l''_{1i},\ l'_{1i}l''_{2i},\ l'_{1i}l''_{3i},\ l'_{2i}l''_{1i},\ l'_{2i}l''_{2i},\ l'_{2i}l''_{3i},\ l'_{3i}l''_{1i},\ l'_{3i}l''_{2i},\ l'_{3i}l''_{3i} & \\ & \vdots & \end{bmatrix} \tag{5.7}$$

Notice that once the line correspondences are established, $\mathbf{B}$ is known, and so is the estimation matrix $\mathbf{A}$.

Hereafter we shall in many places require the use of a simple property about the rank of a matrix, and we state it below.

**Rank invariance property.** Given any matrix $\mathbf{M} = [\mathbf{m}_i]$, where $\mathbf{m}_i$ is the $i$-th row of $\mathbf{M}$,

$$rank([a_i\mathbf{m}_i]) \leq rank(\mathbf{M}) \tag{5.8}$$

for all set of scalars $a_i \in \mathcal{R}$.

### 5.2.2 Linear line space

Linear line space includes only line pencil (with 1-DoF in the line set), point-star (2-DoF in the line set), and ruled-plane (2-DoF in the line set).

As captured by Eqn. (5.4), with $N$ 3-D lines included into the 3-D line set matrix $\mathbf{L}$, the image observations will collectively become a $9 \times N$ matrix $\bar{\mathbf{K}}$. Below we work out the rank property of the estimation matrix $\mathbf{A}$ through the rank property of this observation matrix $\bar{\mathbf{K}}$.

Our presentation will be conducted in this manner. We first show that if the $\bar{\mathbf{K}}$ matrix defined by the line correspondences owns certain rank properties, the rank of the estimation matrix $\mathbf{A}$ will be capped at a certain value. Different sets of rank properties of the $\bar{\mathbf{K}}$ matrix will lead to different caps of the estimation matrix $\mathbf{A}$. These we present in three separate lemmas, one for line-pencil, another for point-star, and yet another for ruled plane. We then in a theorem show that the required sets of rank properties of the $\bar{\mathbf{K}}$ matrix are indeed there should the observed line structure be a line-pencil, point-star, and ruled plane respectively.

**Lemma 5.1.** *Given any* $9 \times N$ *matrix* $\mathbf{K} = [{\mathbf{k}_1}^T, {\mathbf{k}_2}^T, {\mathbf{k}_3}^T]^T$, *where* $\mathbf{k}_i$ *is each a* $3 \times N$ *matrix, , suppose that for all* $i, j = 1, 2, 3$ *and* $i \neq j$, *we have*

- $rank(\mathbf{k}_i) = 2$,
- $rank(\mathbf{K}) = 2$.

*Then if* $\mathbf{K}$ *is taken as the image line observation matrix* $\bar{\mathbf{K}}$ *(and thus all* $\mathbf{k}_i$, $i = 1, 2, 3$, *as* $\mathbf{l}, \mathbf{l}', \mathbf{l}''$ *respectively) to calculate the estimation matrix* $\mathbf{A}$, *the rank of* $\mathbf{A}$ *is at most* 7.

*Proof.* Let $\mathbf{A}_{*,j}$ be the $j$th column of the estimation matrix $\mathbf{A}$, for all $j = 1, 2, \ldots, 27$.

The proof consists of 3 parts: one for the 1st 9 columns of $\mathbf{A}$, the other for its second 9 columns, and the third for its last 9 columns. In each part, we examine how many of the 9 columns are linearly independent and contribute to the overall rank of $\mathbf{A}$.

I **First 9 columns of A:**

Consider the 1st to 9th columns of $\mathbf{A}$, i.e., the columns $\mathbf{A}_{*,1}$ to $\mathbf{A}_{*,9}$.

I(a) As pointed out by Eqn. (5.6), these columns are all about the $N \times 9$ sub-matrix $[-l_{3i}\mathbf{B}_i]$. Both the three sets of columns $\{\mathbf{A}_{*,1}, \mathbf{A}_{*,2}, \mathbf{A}_{*,3}\}$, $\{\mathbf{A}_{*,4}, \mathbf{A}_{*,5}, \mathbf{A}_{*,6}\}$, and $\{\mathbf{A}_{*,7}, \mathbf{A}_{*,8}, \mathbf{A}_{*,9}\}$ are about $[a_i[\mathbf{l}''_{1,i}, \mathbf{l}''_{2,i}, \mathbf{l}''_{3,i}]]$ for some $a_i$. By the Rank Invariance property stated above, the rank of each of such 3-column sets is no more than $rank([\mathbf{l}''_{1,i}, \mathbf{l}''_{2,i}, \mathbf{l}''_{3,i}]) = rank([{\mathbf{l}''_1}^T, {\mathbf{l}''_2}^T, {\mathbf{l}''_3}^T]^T) = rank(\mathbf{k}_3)$ which is 2 by the first stated rank property of $\mathbf{K}$. In other words, in counting the linearly independent columns, we can rule out $\mathbf{A}_{*,3}$, $\mathbf{A}_{*,6}$, and $\mathbf{A}_{*,9}$, and keep only $\mathbf{A}_{*,1}$, $\mathbf{A}_{*,2}$, $\mathbf{A}_{*,4}$, $\mathbf{A}_{*,5}$, $\mathbf{A}_{*,7}$, and $\mathbf{A}_{*,8}$.

I(b) Similar to the above, by the first stated rank property of $\mathbf{K}$ and the rank invariance property, again it can be deduced that $rank([\mathbf{A}_{*,j}, \mathbf{A}_{*,j+3}, \mathbf{A}_{*,j+6}]) \leq rank([\mathbf{l}_1'^T, \mathbf{l}_2'^T, \mathbf{l}_3'^T]^T) = rank(\mathbf{k}_2) = 2$ for $j = 1, 2, 3$. In other words, $\mathbf{A}_{*,7}$ is dependent of $\mathbf{A}_{*,1}$ and $\mathbf{A}_{*,4}$, and $\mathbf{A}_{*,8}$ is dependent of $\mathbf{A}_{*,2}$ and $\mathbf{A}_{*,5}$.

With the above, within the first 9 columns of $\mathbf{A}$, we have at most 4 columns: $\mathbf{A}_{*,1}, \mathbf{A}_{*,2}, \mathbf{A}_{*,4}, \mathbf{A}_{*,5}$, that are linearly independent.

I(c) Next we shall show that $\mathbf{A}_{*,4}$ is linearly independent of $\{\mathbf{A}_{*,1}, \mathbf{A}_{*,2}\}$.

Observing the second rank property of $\mathbf{K}$, we can pick $\{\mathbf{l}_1, \mathbf{l}_2\}$ as the basis of $\{\mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3, \mathbf{l}_1', \mathbf{l}_2', \mathbf{l}_3', \mathbf{l}_1'', \mathbf{l}_2'', \mathbf{l}_3'\}$. In other words, we have

$$\begin{aligned} \mathbf{l}_1' &= c_1\mathbf{l}_1 + c_2\mathbf{l}_2 \\ \mathbf{l}_2' &= c_3\mathbf{l}_1 + c_4\mathbf{l}_2 \\ \mathbf{l}_1'' &= c_5\mathbf{l}_1 + c_6\mathbf{l}_2 \\ \mathbf{l}_2'' &= c_7\mathbf{l}_1 + c_8\mathbf{l}_2 \end{aligned} \tag{5.9}$$

for some $c_i \in \mathbf{R}$.

Suppose that $\mathbf{A}_{*,4}$ is linearly dependent of $\{\mathbf{A}_{*,1}, \mathbf{A}_{*,2}\}$. Then there should exist $X, Y \in \mathbf{R}$ such that $\mathbf{A}_{*,4} = X\mathbf{A}_{*,1} + Y\mathbf{A}_{*,2}$ or simply

$$[l_{2i}' l_{1i}''] = X[l_{1i}' l_{1i}''] + Y[l_{1i}' l_{2i}'']$$

where [*] is a vector with * as its column entry. Using Eqn. (5.9) to expand the above in terms of $\mathbf{l}_{1i}, \mathbf{l}_{2i}$, we have

$$\begin{aligned} [(c_3 l_{1i} + c_4 l_{2i})(c_5 l_{1i} + c_6 l_{2i})] = X[(c_1 l_{1i} + c_2 l_{2i})(c_5 l_{1i} + c_6 l_{2i})] \\ + Y[(c_1 l_{1i} + c_2 l_{2i})(c_7 l_{1i} + c_8 l_{2i})] \end{aligned}$$

or for all $i = 1, 2, \cdots, N$,

$$\begin{aligned} & l_{1i}^2(c_3c_5) + l_{1i}l_{2i}(c_3c_6 + c_4c_5) + l_{2i}^2(c_4c_6) \\ = & l_{1i}^2(Xc_1c_5 + Yc_1c_7) + l_{1i}l_{2i}(Xc_1c_6 + Xc_2c_5 \\ & + Yc_1c_8 + Yc_2c_7) + l_{2i}^2(Xc_2c_6 + Yc_2c_8). \end{aligned}$$

Equating the coefficients of both sides, the above equation can be reduced to

$$\begin{bmatrix} c_1c_5 & c_1c_7 \\ c_1c_6 + c_2c_5 & c_1c_8 + c_2c_7 \\ c_2c_6 & c_2c_8 \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} c_3c_5 \\ c_3c_6 + c_4c_5 \\ c_4c_6 \end{bmatrix}$$

which is an over-constraining set of equations for $X, Y$, showing that $X, Y$ do not exist in general – a contradiction to the supposition. Thus $\mathbf{A}_{*,4}$ is independent of $\mathbf{A}_{*,1}, \mathbf{A}_{*,2}$.

I(d) By a similar "Proof by Contradiction" step, we can show that $\mathbf{A}_{*,5}$ is linearly dependent of $\{\mathbf{A}_{*,1}, \mathbf{A}_{*,2}, \mathbf{A}_{*,4}\}$.

Suppose $\mathbf{A}_{*,5}$ is linearly independent of $\{\mathbf{A}_{*,1}, \mathbf{A}_{*,2}, \mathbf{A}_{*,4}\}$. It means $\{l'_{2i}l''_{2i}\} - X\{l'_{1i}l''_{1i}\} - Y\{l'_{1i}l''_{2i}\} - Z\{l'_{2i}l''_{1i}\} = 0$ has no solution in $X, Y, Z$. But $X, Y, Z$ in

$$\begin{bmatrix} c_1c_5 & c_1c_7 & c_3c_5 \\ c_1c_6 + c_2c_5 & c_1c_8 + c_2c_7 & c_3c_6 + c_4c_5 \\ c_2c_6 & c_2c_8 & c_4c_6 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} c_3c_7 \\ c_3c_8 + c_4c_7 \\ c_4c_8 \end{bmatrix}$$

do supply a solution – a contradiction to the supposition. Thus $\mathbf{A}_{*,5}$ is linearly dependent of $\mathbf{A}_{*,1}, \mathbf{A}_{*,2}, \mathbf{A}_{*,4}$.

Putting together all the above linear dependencies, $\mathbf{A}_{*,3}, \mathbf{A}_{*,5}, \mathbf{A}_{*,6}, \mathbf{A}_{*,7}, \mathbf{A}_{*,8}, \mathbf{A}_{*,9}$ are linearly dependent of $\{\mathbf{A}_{*,1}, \mathbf{A}_{*,2}, \mathbf{A}_{*,4}\}$, and $rank([\mathbf{A}_{*,1}, \ldots, \mathbf{A}_{*,9}]) \leq 3$ .

II **Second 9 columns of A:**

$\{\mathbf{A}_{*,1}, \ldots, \mathbf{A}_{*,9}\}$ and $\{\mathbf{A}_{*,10}, \ldots, \mathbf{A}_{*,18}\}$ are all about $l_{3i}\mathbf{B}_i$. Thus by the same reason as the first 9 columns of $\mathbf{A}$, we have $rank([\mathbf{A}_{*,10}, \ldots, \mathbf{A}_{*,18}]) = rank([\mathbf{A}_{*,1}, \ldots, \mathbf{A}_{*,9}]) \leq$

3.

$\{\mathbf{A}_{*,1}, \mathbf{A}_{*,2}, \mathbf{A}_{*,4}, \mathbf{A}_{*,10}, \mathbf{A}_{*,11}, \mathbf{A}_{*,13}\}$ can be picked as the basis of the first 18 columns of $\mathbf{A}$.

III **Last 9 columns of A:**

Again by the rank invariance property, the rank of the $2N \times 9$ matrix $\begin{bmatrix} [-l_{2i}\mathbf{B}_i] \\ [l_{1i}\mathbf{B}_i] \end{bmatrix}$ is no more than the rank of $[\mathbf{B}_i]$. In other words, the rank property of $\{\mathbf{A}_{*,19}, ..., \mathbf{A}_{*,27}\}$ is the same as that of $\{\mathbf{A}_{*,1}, ..., \mathbf{A}_{*,9}\}$ and $\{\mathbf{A}_{*,10}, ..., \mathbf{A}_{*,18}\}$. Thus, from the proof in Step **I**, $\mathbf{A}_{*,21}, \mathbf{A}_{*,23}, ..., \mathbf{A}_{*,27}$ are linearly dependent of $[\mathbf{A}_{*,19}, \mathbf{A}_{*,20}, \mathbf{A}_{*,22}\}$, and $rank([\mathbf{A}_{*,19}, \ldots, \mathbf{A}_{*,27}]) \leq 3$.

Finally, we need to determine the linear dependency between the 3 column vectors $\mathbf{A}_{*,19}, \mathbf{A}_{*,20}, \mathbf{A}_{*,22}$ and the basis of $\{\mathbf{A}_{*,1}, \ldots, \mathbf{A}_{*,18}\}$. Again, "Proof by contradiction" schemes similar to those above can be used to show that $\mathbf{A}_{*,19}$ is independent of $\{\mathbf{A}_{*,1}, \mathbf{A}_{*,2}, \mathbf{A}_{*,4}, \mathbf{A}_{*,10}, \mathbf{A}_{*,11}, \mathbf{A}_{*,13}\}$, but $\{\mathbf{A}_{*,20}, \mathbf{A}_{*,22}\}$ is dependent of $\{\mathbf{A}_{*,1}, \mathbf{A}_{*,2}, \mathbf{A}_4,$ $\mathbf{A}_{*,10}, \mathbf{A}_{*,11}, \mathbf{A}_{*,13}, \mathbf{A}_{*,19}\}$. Due to page limit, we omit the elaboration of such trivial steps here. To conclude, the rank of $\mathbf{A}$ is at most 7.

□

**Lemma 5.2.** *Given any* $9 \times N$ *matrix* $\mathbf{K} = [\mathbf{k}_1{}^T, \mathbf{k}_2{}^T, \mathbf{k}_3{}^T]^T$, *where* $\mathbf{k}_i$ *is each a* $3 \times N$ *matrix, suppose that for all* $i, j = 1, 2, 3$ *and* $i \neq j$, *we have*

- $rank(\mathbf{k}_i) = 2$,
- $rank([\mathbf{k}_i,\ \mathbf{k}_j]^T) = 3$,
- $rank(\mathbf{K}) = 3$.

*Then if* $\mathbf{K}$ *is taken as the image line observation matrix* $\bar{\mathbf{K}}$ *(and thus all* $\mathbf{k}_i$, $i = 1, 2, 3$, *as* $\mathbf{l}, \mathbf{l}', \mathbf{l}''$ *respectively) to calculate the estimation matrix* $\mathbf{A}$, *the rank of* $\mathbf{A}$ *is at most* 11.

*Proof.* We follow the same notations and the proof structure as in Lemma 5.1.

I **First 9 columns of A:**

I(a) As $rank(\mathbf{l}_i) = 2$, by Step **I** of the proof for Lemma 5.1, it can be deduced that $rank([\mathbf{A}_{*,j}\ \mathbf{A}_{*,j+1}\ \mathbf{A}_{*,j+2}]) \leq 2$ for $j = 1, 4, 7$, and $rank([\mathbf{A}_{*,j}\ \mathbf{A}_{*,j+3}\ \mathbf{A}_{*,j+6}]) \leq 2$ for $j = 1, 2, 3$.

I(b) Next we show that $\mathbf{A}_{*,4}$ is linearly independent of $\{\mathbf{A}_{*,1}, \mathbf{A}_{*,2}\}$.

Observing the second and third rank properties of $\mathbf{K}$, we can pick $\{\mathbf{l}_1, \mathbf{l}_2, \mathbf{l}'_2\}$ as the basis of $\{\mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3, \mathbf{l}'_1, \mathbf{l}'_2, \mathbf{l}'_3, \mathbf{l}''_1, \mathbf{l}''_2, \mathbf{l}'_3\}$. In other words, we have

$$\begin{aligned} \mathbf{l}'_2 &= c_1\mathbf{l}_1 + c_2\mathbf{l}_2 + c_3\mathbf{l}'_1 \\ \mathbf{l}''_1 &= c_4\mathbf{l}_1 + c_5\mathbf{l}_2 + c_6\mathbf{l}'_1 \\ \mathbf{l}''_2 &= c_7\mathbf{l}_1 + c_8\mathbf{l}_2 + c_9\mathbf{l}'_1 \end{aligned}$$

for some $c_i \in \mathbf{R}$.

Suppose that $\mathbf{A}_{*,4}$ is linearly dependent of $\{\mathbf{A}_{*,1}, \mathbf{A}_{*,2}\}$, then there should exist $X, Y \in \mathbf{R}$, such that

$$[l'_{2i} l''_{1i}] = X[l'_{1i} l''_{1i}] + Y[l'_{1i} l''_{2i}]$$

Expanding the above in terms of $\mathbf{l}_{1i}, \mathbf{l}_{2i}$ and equating the coefficients of both sides, the above equation can be reduced to

$$\begin{bmatrix} c_4 & c_7 \\ c_5 & c_8 \\ c_6 & c_9 \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} c_1c_6 + c_3c_4 \\ c_2c_6 + c_3c_5 \\ c_3c_6 \end{bmatrix}$$

and

$$\begin{bmatrix} c_1c_4 \\ c_1c_5 + c_2c_4 \\ c_2c_5 \end{bmatrix} = 0$$

which is an over-constraining set of equations for $X, Y$, showing that $X, Y$ do not exist in general – a contradiction to the supposition. Thus, $\mathbf{A}_{*,4}$ is independent of $\mathbf{A}_{*,1}, \mathbf{A}_{*,2}$.

I(c) By a similar "Proof by contradiction" treatment, we can show that $\mathbf{A}_{*,5}$ is linearly independent of $\{\mathbf{A}_{*,1}, \mathbf{A}_{*,2}, \mathbf{A}_{*,4}\}$.

Putting together all the above linear dependencies, $\mathbf{A}_{*,3}, \mathbf{A}_{*,6}, \mathbf{A}_{*,7}, \mathbf{A}_{*,8}, \mathbf{A}_{*,9}$ are linearly dependent of $\{\mathbf{A}_{*,1}, \mathbf{A}_{*,2}, \mathbf{A}_{*,4}, \mathbf{A}_{*,5}\}$, and $rank([\mathbf{A}_{*,1} \ldots \mathbf{A}_{*,9}]) \leq 4$.

II **Second 9 columns of A:**

By Step **II** of the proof for Lemma 5.1, we have $rank([\mathbf{A}_{*,10}, \ldots, \mathbf{A}_{*,18}]) = rank([\mathbf{A}_{*,1}, \ldots, \mathbf{A}_{*,9}]) \leq 4$.

$\{\mathbf{A}_{*,1}, \mathbf{A}_{*,2}, \mathbf{A}_{*,4}, \mathbf{A}_{*,5}, \mathbf{A}_{*,10}, \mathbf{A}_{*,11}, \mathbf{A}_{*,13}, \mathbf{A}_{*,14}\}$ can be picked as the basis of the first 18 columns of $\mathbf{A}$.

III **Last 9 columns of A:**

By Step **III** of the proof for Lemma 5.1, the rank property of $\{\mathbf{A}_{*,19}, ..., \mathbf{A}_{*,27}\}$ is the same as that of $\{\mathbf{A}_{*,1}, ..., \mathbf{A}_{*,9}\}$ and $\{\mathbf{A}_{*,10}, ..., \mathbf{A}_{*,18}\}$. Then by Step **I** of the proof for Lemma 5.1, $\mathbf{A}_{*,21}, \mathbf{A}_{*,23}, ..., \mathbf{A}_{*,27}$ is linearly dependent of $\{\mathbf{A}_{*,19}, \mathbf{A}_{*,20}, \mathbf{A}_{*,22}, \mathbf{A}_{*,23}\}$, and $rank([\mathbf{A}_{*,19}, \ldots, \mathbf{A}_{*,27}]) \leq 4$.

Finally, we need to determine the linear dependency between the 4 column vectors $\mathbf{A}_{*,19}, \mathbf{A}_{*,20}, \mathbf{A}_{*,22}, \mathbf{A}_{*,23}$ and the basis of $\{\mathbf{A}_{*,1}, \ldots, \mathbf{A}_{*,18}\}$. Again, a "Proof by contradiction" step can be used to show that $\mathbf{A}_{*,19}, \mathbf{A}_{*,20}, \mathbf{A}_{*,22}$ are independent of $\{\mathbf{A}_{*,1}, \mathbf{A}_{*,2}, \mathbf{A}_{*,4}, \mathbf{A}_{*,5}, \mathbf{A}_{*,10}, \mathbf{A}_{*,11}, \mathbf{A}_{*,13}, \mathbf{A}_{*,14}\}$, but $\{\mathbf{A}_{*,23}\}$ is dependent of them. Due to page limit, here we omit the elaboration. To conclude, the rank of $\mathbf{A}$ is at most 11. □

**Lemma 5.3.** *Given any* $9 \times N$ *matrix* $\mathbf{K} = [\mathbf{k}_1{}^T, \mathbf{k}_2{}^T, \mathbf{k}_3{}^T]^T$, *where* $\mathbf{k}_i$ *is each a* $3 \times N$ *matrix, suppose that for all* $i, j = 1, 2, 3$ *and* $i \neq j$, *we have*

- $rank(\mathbf{k}_i) = 3$,
- $rank(\mathbf{K}) = 3$.

*Then if* $\mathbf{K}$ *is taken as the image line observation matrix* $\bar{\mathbf{K}}$ *(and thus all* $\mathbf{k}_i$, $i = 1, 2, 3$, *as* $\mathbf{l}, \mathbf{l}', \mathbf{l}''$ *respectively) to calculate the estimation matrix* $\mathbf{A}$, *the rank of* $\mathbf{A}$ *is at most* 15.

*Proof.* Again we adopt the same notations and the proof structure as used in Lemma 5.1.

I **First 9 columns of A:**

I(a) As $rank(\mathbf{l}_i) = 3$, by Step **I** of the proof for Lemma 5.1, we have $rank([\mathbf{A}_{*,j}\ \mathbf{A}_{*,j+1}\ \mathbf{A}_{*,j+2}]) = 3$ for $j = 1, 4, 7$, and $rank([\mathbf{A}_{*,j}\ \mathbf{A}_{*,j+3}\ \mathbf{A}_{*,j+6}]) = 3$ for $j = 1, 2, 3$.

I(b) Next we show that $\mathbf{A}_{*,4}$ is linearly independent of $\{\mathbf{A}_{*,1}, \mathbf{A}_{*,2}, \mathbf{A}_{*,3}\}$.

Observing the second rank property of $\mathbf{K}$, we can pick $\mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3$ as the basis of $\{\mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3, \mathbf{l}'_1, \mathbf{l}'_2, \mathbf{l}'_3, \mathbf{l}''_1, \mathbf{l}''_2, \mathbf{l}'_3\}$. In other words, we have

$$\begin{aligned}
\mathbf{l}'_1 &= c_1\mathbf{l}_1 + c_2\mathbf{l}_2 + c_3\mathbf{l}'_1 \\
\mathbf{l}'_2 &= c_4\mathbf{l}_1 + c_5\mathbf{l}_2 + c_6\mathbf{l}'_1 \\
\mathbf{l}''_1 &= c_7\mathbf{l}_1 + c_8\mathbf{l}_2 + c_9\mathbf{l}'_1 \\
\mathbf{l}''_2 &= c_{10}\mathbf{l}_1 + c_{11}\mathbf{l}_2 + c_{12}\mathbf{l}'_1 \\
\mathbf{l}''_3 &= c_{13}\mathbf{l}_1 + c_{14}\mathbf{l}_2 + c_{15}\mathbf{l}'_1
\end{aligned}$$

for some $c_i \in \mathbf{R}$.

Suppose that $\mathbf{A}_{*,4}$ is linearly dependent of $\{\mathbf{A}_{*,1}, \mathbf{A}_{*,2}, \mathbf{A}_{*,3}\}$, then there should exist $X, Y \in \mathbf{R}$, such that

$$[l'_{2i} l''_{1i}] = X[l'_{1i} l''_{1i}] + Y[l'_{1i} l''_{2i}] + Z[l'_{1i} l''_{3i}]$$

Expanding the above in terms of $\mathbf{l}_{1i}, \mathbf{l}_{2i}$ and equating the coefficients of both sides, the above equation can be reduced to

$$\begin{bmatrix}
c_1c_7 & c_1c_{10} & c_1c_{13} \\
c_2c_8 & c_2c_{11} & c_2c_{14} \\
c_3c_9 & c_3c_{12} & c_3c_{15} \\
c_1c_8 + c_2c_7 & c_1c_{11} + c_2c_{10} & c_1c_{14} + c_2c_{13} \\
c_1c_9 + c_3c_7 & c_1c_{12} + c_3c_{10} & c_1c_{15} + c_3c_{13} \\
c_2c_9 + c_3c_8 & c_2c_{12} + c_3c_{11} & c_2c_{15} + c_3c_{14}
\end{bmatrix}
\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} =
\begin{bmatrix}
c_4c_7 \\ c_5c_8 \\ c_6c_9 \\ c_4c_8 + c_5c_7 \\ c_4c_9 + c_6c_7 \\ c_5c_9 + c_6c_8
\end{bmatrix}$$

which is an over-constraining set of equations for $X, Y, Z$, showing that $X, Y, Z$ do not exist in general – a contradiction to the supposition. Thus, $\mathbf{A}_{*,4}$ is independent of $\mathbf{A}_{*,1}, \mathbf{A}_{*,2}, \mathbf{A}_{*,3}$.

I(c) By a similar method, we can show that $\mathbf{A}_{*,5}$ is also linearly independent of $\{\mathbf{A}_{*,1}, \ldots, \mathbf{A}_{*,4}\}$; $\mathbf{A}_{*,6}$ is dependent of $\{\mathbf{A}_{*,1}, \ldots, \mathbf{A}_{*,5}\}$; $\mathbf{A}_{*,7}$ is independent of $\{\mathbf{A}_{*,1}, \ldots, \mathbf{A}_{*,5}\}$; $\mathbf{A}_{*,8}, \mathbf{A}_{*,9}$ are linearly dependent of $\{\mathbf{A}_{*,1}, \ldots, \mathbf{A}_{*,5}, \mathbf{A}_{*,7}\}$.

Putting together all the above linear dependencies, $\mathbf{A}_{*,6}, \mathbf{A}_{*,8}, \mathbf{A}_{*,9}, \mathbf{A}_{*,9}$ are linearly dependent of $\mathbf{A}_{*,1}, ..., \mathbf{A}_{*,5}, \mathbf{A}_{*,7}$, and $rank([\mathbf{A}_{*,1}, \ldots, \mathbf{A}_{*,9}]) = 6$ by Eqn. (5.6).

II **Second 9 columns of A:**

By Step **II** of the proof for Lemma 5.1, we have $rank([\mathbf{A}_{*,10}, \ldots, \mathbf{A}_{*,18}]) = rank([\mathbf{A}_{*,1}, \ldots, \mathbf{A}_{*,9}]) \leq 6$.

III **Last 9 columns of A:** By Step **III** of the proof for Lemma 5.1, the rank property of $\{\mathbf{A}_{*,19}, ..., \mathbf{A}_{*,27}\}$ is the same as that of $\{\mathbf{A}_{*,1}, ..., \mathbf{A}_{*,9}\}$ and $\{\mathbf{A}_{*,10}, ..., \mathbf{A}_{*,18}\}$.

By a few "Proof by contradiction" steps similar to those used in Lemma 5.1 in discussing the linear dependency, it can be proved that rank of $\mathbf{A}$ is at most 15.

□

**Theorem 5.2 (Linear Line Space).** *The rank of the tensor estimation matrix is at most* $7, 11, 15$ *respectively for each of the following line structures of linear line space: line-pencil, point-star, and ruled plane; the minimal number of lines for the full information of each of the line structures to be reflected in image observations is* $4, 6, 8$ *lines respectively.*

*Proof.* Eqn. (5.4)relates the image observations $[\mathbf{l}^T, \mathbf{l}'^T, \mathbf{l}''^T]^T$ to the 3-D line set $\mathbf{L}$ through the cameras' line projection matrices $\tilde{\mathbf{P}}, \tilde{\mathbf{P}}', \tilde{\mathbf{P}}''$. The relationship leads to the fact that:

$$rank(\begin{bmatrix} \mathbf{l} \\ \mathbf{l}' \\ \mathbf{l}'' \end{bmatrix}) \leq min(rank(\begin{bmatrix} \tilde{\mathbf{P}} \\ \tilde{\mathbf{P}}' \\ \tilde{\mathbf{P}}'' \end{bmatrix}), rank(\mathbf{L})) \tag{5.10}$$

Since $[\tilde{\mathbf{P}}^T, \tilde{\mathbf{P}}'^T, \tilde{\mathbf{P}}''^T]^T$ is of full column rank (=6) in general, we have $rank([\mathbf{l}^T, \mathbf{l}'^T, \mathbf{l}''^T]^T) = rank(\mathbf{L})$.

I **Line Pencil:**

By Eqn. (4.5), $rank(\mathbf{L}) = 6-3-1 = 2$, with the 3 coming from three linear constraints on $\mathbf{L}$ and the 1 from the Plücker equality (which is degraded to be linear, or else the structure does not belong to linear line space). By the assumption that the projection is general, plus Eqn. (5.10), it can be deduced that $rank([\mathbf{l}^T, \mathbf{l}'^T, \mathbf{l}''^T]^T]) = 2$. Since $rank(\mathbf{l}) = rank(\mathbf{l}') = rank(\mathbf{l}'') = 2$, we have $rank([\mathbf{l}^T, \mathbf{l}'^T]^T) = rank([\mathbf{l}^T, \mathbf{l}''^T]^T) =$

$rank([\mathbf{l}'^T, \mathbf{l}''^T]^T) = 2$. All constraints on image lines have thus been considered, and they give rise to the rank properties of $\bar{\mathbf{K}}$ required by Lemma 5.1. By Lemma 5.1, the rank of $\mathbf{A}$ is at most 7 for line pencil. As each line correspondence provides 2 linear constraints on $\mathbf{A}$, the minimum number of line correspondences needed is $\lceil 7/2 \rceil = 4$.

II **Point star and ruled plane:**

Similar to the above, for such line structures, $rank(\mathbf{L}) = 6 - 2 - 1 = 3$, with the 2 coming from two linear constraints and the 1 from the Plücker equality. The image lines of point-star co-intersect in each image, so $rank(\mathbf{l}) = rank(\mathbf{l}') = rank(\mathbf{l}'') = 2$, but $rank([\mathbf{l}^T, \mathbf{l}'^T]^T) = rank([\mathbf{l}^T, \mathbf{l}''^T]^T) = rank([\mathbf{l}'^T, \mathbf{l}''^T]^T) = 3$ because the points of intersection in different images are generally not the same. In contrast, the image lines of ruled plane have $rank(\mathbf{l}) = rank(\mathbf{l}') = rank(\mathbf{l}'') = 3$ and $rank([\mathbf{l}^T, \mathbf{l}'^T]^T) = rank([\mathbf{l}^T, \mathbf{l}''^T]^T) = rank([\mathbf{l}'^T, \mathbf{l}''^T]^T) = 3$. The rank properties governing the image lines of point star and ruled plane thus satisfy the condition of Lemma 5.2 and Lemma 5.3 respectively. By Lemmas 5.2 and 5.3, the rank of $\mathbf{A}$ is at most 11 and 15 respectively for these two linear line structures. The minimum number of line correspondences needed is $\lceil 11/2 \rceil = 6$ for point star and $\lceil 15/2 \rceil = 8$ for ruled plane.

□

In the following subsections we tackle the cases of linear ruled surface, linear line congruence, and linear line complex. We adopt a presentation similar to that for linear line space. We first show that if the $\bar{\mathbf{K}}$ matrix defined by the line correspondences owns certain rank properties, the rank of the estimation matrix $\mathbf{A}$ will be capped at a certain value for the cases of linear ruled surface, linear line congruence, and linear line complex respectively. For the case of linear ruled surface, we simply borrow Lemma 5.3 above. For the cases of linear line congruence and linear line complex, we have two separate lemmas to point out the required rank properties of the $\bar{\mathbf{K}}$ matrix and the resultant rank of the estimation matrix $\mathbf{A}$. We then in two theorems show that the rank properties of the $\bar{\mathbf{K}}$ are indeed there should the observed line structure be a linear line congruence and linear line complex respectively.

### 5.2.3 Linear ruled surface

**Theorem 5.3 (Linear Ruled Surface).** *The rank of the tensor estimation matrix is at most* 12 *if the available* 3*-view observations are about a set of at least* 6 *lines that belong to a general linear ruled surface.*

*Proof.* For linear ruled surface, $\text{rank}(\mathbf{L}) = 6 - 3 = 3$, with the "3" coming from three linear constraints – a characteristic of the line structure. The fact that the line structure here does not belong to linear line space means the Plücker equality is no longer a linear constraint. By Eqn. (5.10), as the projection is assumed to be general, $\text{rank}([\mathbf{l}^T\ \mathbf{l}'^T\ \mathbf{l}''^T]^T) = 3$. Since $\text{rank}(\mathbf{l}) = \text{rank}(\mathbf{l}') = \text{rank}(\mathbf{l}'') = 3$, by Lemma 5.3, the rank of $\mathbf{A}$ is at most 15 for ruled surface.

However, the Plücker equality to be satisfied by $\mathbf{L}$, though nonlinear, does lead to constraints on the image lines $\mathbf{l}, \mathbf{l}', \mathbf{l}''$ through the camera projection process (as captured by Eqn. (5.4)), which could induce additional linear constraints on $\mathbf{A}$, as described below. With the three cameras assumed to be in a general configuration, by means of Eqn. (5.4), the image lines can be expressed in terms of the camera matrices and the Plücker lines $\mathbf{L}$. Algebraic manipulations show that as long as the Plücker equality is satisfied, we have

$$\begin{aligned}[l_{3i}l'_{3i}l''_{1i}] = c_1[l_{3i}l'_{3i}l''_{1i}] + c_2[l_{3i}l'_{3i}l''_{2i}] + c_3[l_{3i}l'_{3i}l''_{3i}] \\ + c_4[l_{3i}l'_{2i}l''_{1i}] + c_5[l_{3i}l'_i 2l''_{2i}]\end{aligned}$$

([*] represents a vector with * as its column entry) for some $c_r \in \mathcal{R}, r = 1, ..., 5$, which are solely determined by the three camera matrices and are thus constants. Similar properties hold for $[l_{1i}l'_{2i}l''_{1i}]$ and $[l_{2i}l'_{2i}l''_{1i}]$. With these, it follows that $\mathbf{A}_{*,7}$ is linearly dependent of $\{\mathbf{A}_{*,1}, ..., \mathbf{A}_{*,5}\}$; $\mathbf{A}_{*,16}$ is linearly dependent of $\{\mathbf{A}_{*,10}, ..., \mathbf{A}_{*,14}\}$; and $\mathbf{A}_{*,22}$ is linearly dependent of $\{\mathbf{A}_{*,1}, ..., \mathbf{A}_{*,5}, \mathbf{A}_{*,10}, ..., \mathbf{A}_{*,14}, \mathbf{A}_{*,19}, \mathbf{A}_{*,20}\}$. Thus, three DoFs have to be deducted and the rank of $\mathbf{A}$ is at most $15 - 3 = 12$. As each line correspondence provides 2 linear constraints on $\mathbf{A}$, the minimum number of line correspondences needed is $\lceil 12/2 \rceil = 6$. □

### 5.2.4 Linear line congruence

**Lemma 5.4.** *Given any* $9 \times N$ *matrix* $\mathbf{K} = [\mathbf{k}_1{}^T, \mathbf{k}_2{}^T, \mathbf{k}_3{}^T]^T$, *where* $\mathbf{k}_i$ *is each a* $3 \times N$ *matrix, suppose that for all* $i, j = 1, 2, 3$ *and* $i \neq j$, *we have*

- $rank(\mathbf{k}_i) = 3$,
- $rank([\mathbf{k}_i, \mathbf{k}_j]^T) = 4$,
- $rank(\mathbf{K}) = 4$.

*Then if* $\mathbf{K}$ *is taken as the image line observation matrix* $\bar{\mathbf{K}}$ *(and thus all* $\mathbf{k}_i$, $i = 1, 2, 3$, *as* $\mathbf{l}, \mathbf{l}', \mathbf{l}''$ *respectively) to calculate the estimation matrix* $\mathbf{A}$, *the rank of* $\mathbf{A}$ *is at most* 22.

*Proof.* Again we adopt the same notations and proof structure as used in Lemma 5.1.

I **First 9 columns of A:**

I(a) Since $rank(\mathbf{l}) = rank(\mathbf{l}') = rank(\mathbf{l}'') = 3$, by Step **I** of the proof for Lemma 5.1, we have $rank([\mathbf{A}_{*,j}, \mathbf{A}_{*,j+1}, \mathbf{A}_{*,j+2}]) = 3$ for $j = 1, 4, 7$ and $rank([\mathbf{A}_{*,j}, \mathbf{A}_{*,j+3}, \mathbf{A}_{*,j+6}]) = 3$ for $j = 1, 2, 3$.

I(b) Next we show that $\mathbf{A}_{*,5}$ is linearly independent of $\{\mathbf{A}_{*,1}, ..., \mathbf{A}_{*,4}\}$.

Observing the second and third rank properties of $\mathbf{K}$, we can pick $\mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3, \mathbf{l}'_1$ as the basis of $\{\mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3, \mathbf{l}'_1, \mathbf{l}'_2, \mathbf{l}'_3, \mathbf{l}''_1, \mathbf{l}''_2, \mathbf{l}'_3\}$. In other words, we have

$$\begin{aligned}
\mathbf{l}'_2 &= c_1\mathbf{l}_1 + c_2\mathbf{l}_2 + c_3\mathbf{l}_3 + c_4\mathbf{l}'_1 \\
\mathbf{l}'_3 &= c_5\mathbf{l}_1 + c_6\mathbf{l}_2 + c_7\mathbf{l}_3 + c_8\mathbf{l}'_1 \\
\mathbf{l}''_1 &= c_9\mathbf{l}_1 + c_{10}\mathbf{l}_2 + c_{11}\mathbf{l}_3 + c_{12}\mathbf{l}'_1 \\
\mathbf{l}''_2 &= c_{13}\mathbf{l}_1 + c_{14}\mathbf{l}_2 + c_{15}\mathbf{l}_3 + c_{16}\mathbf{l}'_1 \\
\mathbf{l}''_3 &= c_{17}\mathbf{l}_1 + c_{18}\mathbf{l}_2 + c_{19}\mathbf{l}_3 + c_{20}\mathbf{l}'_1
\end{aligned}$$

for some $c_i \in \mathbf{R}$.

Suppose that $\mathbf{A}_{*,5}$ is linearly dependent of $\{\mathbf{A}_{*,1}, \ldots, \mathbf{A}_{*,4}\}$, then there should exist $X, Y, Z, W \in \mathbf{R}$, such that

$$\{l'_{2i}l''_{2i}\} = W\{l'_{1i}l''_{1i}\} + X\{l'_{1i}l''_{2i}\} + Y\{l'_{1i}l''_{3i}\} + Z\{l'_{2i}l''_{1i}\}$$

It can be expanded similarly to an over-constraining system of equations like Eqn. (5.10), showing that $W, X, Y, Z$ do not exist in general – a contradiction to the supposition. Thus, $\mathbf{A}_{*,5}$ is independent of $\{\mathbf{A}_{*,1}, \ldots, \mathbf{A}_{*,4}\}$.

I(c) By a similar method, we can show that $\mathbf{A}_{*,6}$ is linearly independent of $\{\mathbf{A}_{*,1}, \ldots, \mathbf{A}_{*,5}\}$; $\mathbf{A}_{*,7}$ is independent of $\{\mathbf{A}_{*,1}, \ldots, \mathbf{A}_{*,6}\}$; $\mathbf{A}_{*,8}$ is independent of $\{\mathbf{A}_{*,1}, \ldots, \mathbf{A}_{*,7}\}$; $\mathbf{A}_{*,9}$ is independent of $\{\mathbf{A}_{*,1}, \ldots, \mathbf{A}_{*,8}\}$.

Putting together all the above linear dependencies, $\mathbf{A}_{*,9}$ is independent of $\{\mathbf{A}_{*,1}, \ldots, \mathbf{A}_{*,8}\}$, and $rank([\mathbf{A}_{*,1}, \ldots, \mathbf{A}_{*,9}]) \leq 8$ by Eqn. (5.6).

II **Second 9 columns of A:** By Step **II** of the proof for Lemma 5.1, we have $rank([\mathbf{A}_{*,10}, \ldots, \mathbf{A}_{*,18}]) = rank([\mathbf{A}_{*,1}, \ldots, \mathbf{A}_{*,9}]) \leq 8$.

III **Last 9 columns of A:** By Step **III** of the proof for Lemma 5.1, the rank property of $\{\mathbf{A}_{*,19}, ..., \mathbf{A}_{*,27}\}$ is the same as that of $\{\mathbf{A}_{*,1}, ..., \mathbf{A}_{*,9}\}$ and $\{\mathbf{A}_{*,10}, ..., \mathbf{A}_{*,18}\}$.

By a few "roof by contradiction" steps similar to those used in Lemma 5.1 in discussing the linear dependency, it can be proved that rank of $\mathbf{A}$ is at most 22. □

**Theorem 5.4 (Linear Line Congruence).** *The rank of the tensor estimation matrix is at most* 19 *if the image observables are about a set of at least* 10 *lines that belong to a general linear congruence.*

*Proof.* For linear congruence, $rank(\mathbf{L}) = 6 - 2 = 4$, with the 2 coming from two linear constraints. With the assumption that the linear congruence here does not belong to linear line space, the Plücker equality is no longer a linear constraint. By Eqn. (5.10), since projection is assumed to be general, $rank([\mathbf{l}^T\ \mathbf{l}'^T\ \mathbf{l}''^T]^T) = rank(\mathbf{L}) = 4$. By $rank(\mathbf{l}) = rank(\mathbf{l}') = rank(\mathbf{l}'') = 3$ and Lemma 5.4, the rank of $\mathbf{A}$ is at most 22 for linear congruence.

However, as in the step in proving Theorem (5.3) for ruled surface, three DoFs have to be deducted and the rank of $\mathbf{A}$ is at most $22 - 3 = 19$ for linear congruence. As each line correspondence provides 2 linear constraints on $\mathbf{A}$, the minimum number of line correspondences needed is $\lceil 19/2 \rceil = 10$.

□

**Minimal surface - Helicoid.** Helicoid is defined by the following parametric equations in Cartesian coordinates,

$$x = \rho cos(\alpha\theta) y = \rho sin(\alpha\theta) z = \theta \tag{5.11}$$

where $-\infty < \rho < \infty$, $-\infty < \theta < \infty$, $\alpha$ is a constant.

**Theorem 5.5.** *The rank of estimation matrix is at most* 19 *for the set of* 3*-D lines that belong to a helicoid.*

*Proof.* Just find out that helicoid belongs to linear congruence, and apply Theorem 5.4. □

### 5.2.5 Linear line complex

**Lemma 5.5.** *Given any* $9 \times N$ *matrix* $\mathbf{K} = [\mathbf{k}_1{}^T, \mathbf{k}_2{}^T, \mathbf{k}_3{}^T]^T$, *where* $\mathbf{k}_i$ *is each a* $3 \times N$ *matrix, suppose that for all* $i, j = 1, 2, 3$ *and* $i \neq j$, *we have*

- $rank(\mathbf{k}_i) = 3$,
- $rank([\mathbf{k}_i, \mathbf{k}_j]^T) = 5$,
- $rank(\mathbf{K}) = 5$.

*Then if* $\mathbf{K}$ *is taken as the image line observation matrix* $\bar{\mathbf{K}}$ *(and thus all* $\mathbf{k}_i$, $i = 1, 2, 3$, *as* $\mathbf{l}, \mathbf{l}', \mathbf{l}''$ *respectively) to calculate the estimation matrix* $\mathbf{A}$, *the rank of* $\mathbf{A}$ *is at most* 26.

*Proof.* Again we adopt the notations and the proof structure as used in Lemma 5.1.

I **First 9 columns of A:**

I(a) As $rank(\mathbf{l}) = rank(\mathbf{l}') = rank(\mathbf{l}'') = 3$, by Step **I** of the proof for Lemma 5.1, we have $rank([\mathbf{A}_{*,j}, \mathbf{A}_{*,j+1}, \mathbf{A}_{*,j+2}]) = 3$ for $j = 1, 4, 7$, and $rank([\mathbf{A}_{*,j}, \mathbf{A}_{*,j+3}, \mathbf{A}_{*,j+6}]) = 3$ for $j = 1, 2, 3$.

I(b) Next we show that $\mathbf{A}_{*,6}$ is linearly independent of $\{\mathbf{A}_{*,1}, \ldots, \mathbf{A}_{*,5}\}$.

Observing the second and third rank properties of $\mathbf{K}$, we can pick $\mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3, \mathbf{l}'_1, \mathbf{l}'_2$ as the basis of $\{\mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3, \mathbf{l}'_1, \mathbf{l}'_2, \mathbf{l}'_3, \mathbf{l}''_1, \mathbf{l}''_2, \mathbf{l}'_3\}$. In other words, we have

$$\begin{aligned}
\mathbf{l}'_3 &= c_1\mathbf{l}_1 + c_2\mathbf{l}_2 + c_3\mathbf{l}_3 + c_4\mathbf{l}'_1 + c_5\mathbf{l}'_2 \\
\mathbf{l}''_1 &= c_6\mathbf{l}_1 + c_7\mathbf{l}_2 + c_8\mathbf{l}_3 + c_9\mathbf{l}'_1 + c_{10}\mathbf{l}'_2 \\
\mathbf{l}''_2 &= c_{11}\mathbf{l}_1 + c_{12}\mathbf{l}_2 + c_{13}\mathbf{l}_3 + c_{14}\mathbf{l}'_1 + c_{15}\mathbf{l}'_2 \\
\mathbf{l}''_3 &= c_{16}\mathbf{l}_1 + c_{17}\mathbf{l}_2 + c_{18}\mathbf{l}_3 + c_{19}\mathbf{l}'_1 + c_{20}\mathbf{l}'_2
\end{aligned}$$

for some $c_i \in \mathbf{R}$.

Suppose that $\mathbf{A}_{*,6}$ is linearly dependent of $\{\mathbf{A}_{*,1}, \ldots, \mathbf{A}_{*,5}\}$, then there should exist $V, W, X, Y, Z \in \mathbf{R}$, such that

$$\begin{aligned}
\{l'_{2i}l''_{3i}\} &= V\{l'_{1i}l''_{1i}\} + W\{l'_{1i}l''_{2i}\} + X\{l'_{1i}l''_{3i}\} + Y\{l'_{2i}l''_{1i}\} \\
&+ Z\{l'_{2i}l''_{2i}\}
\end{aligned}$$

It can be expanded similarly to an over-constraining system of equations like Eqn. (5.10), which shows that $V, W, X, Y, Z$ do not exist in general – a contradiction to the supposition. Thus, $\mathbf{A}_{*,6}$ is independent of $\{\mathbf{A}_{*,1}, \ldots, \mathbf{A}_{*,5}\}$.

I(c) By a similar method, we can show that $\mathbf{A}_{*,j+1}$ is linearly independent of $\{\mathbf{A}_{*,1}, \ldots, \mathbf{A}_{*,j}\}$, for $j = 6, 7, 8$.

Putting together all the above linear dependencies, $rank([\mathbf{A}_{*,1}, \ldots, \mathbf{A}_{*,9}]) = 9$ by Eqn. (5.6).

II **Second 9 columns of A:**

By Step **II** of the proof for Lemma 5.1, we have $rank([\mathbf{A}_{*,10}, \ldots, \mathbf{A}_{*,18}]) = rank([\mathbf{A}_{*,1}, \ldots, \mathbf{A}_{*,9}]) = 9$.

III **Last 9 columns of A:**

By Step **III** of the proof for Lemma 5.1, the rank property of $\{\mathbf{A}_{*,19}, ..., \mathbf{A}_{*,27}\}$ is the same as that of $\{\mathbf{A}_{*,1}, ..., \mathbf{A}_{*,9}\}$ and $\{\mathbf{A}_{*,10}, ..., \mathbf{A}_{*,18}\}$.

Table 5.1: Rank classification of linear line structures.

| **Linear line structure** | **Rank property** | **Min. No. lines** |
|---|---|---|
| Line pencil | 7 | 4 |
| Point star | 11 | 6 |
| Ruled plane | 15 | 8 |
| Linear ruled surface | 12 | 6 |
| Linear congruence | 19 | 10 |
| Linear complex | 23 | 12 |

By a few "Proof by contradiction" steps similar to those used in Lemma 5.1 in discussing the linear dependency, it can be proved that rank of $\mathbf{A}$ is at most 26. □

**Theorem 5.6 (Linear Line Complex).** *The rank of the tensor estimation matrix is at most* 23 *if the image observables are about a set of at least* 12 *lines that belong to a general linear line complex.*

*Proof.* For linear complex, $rank(\mathbf{L}) = 6 - 1 = 5$, with the 1 coming from one linear constraint. The Plücker equality is no longer a linear constraint. By Eqn. (5.10), since projection is assumed to be general, $rank([\mathbf{l}^T, \mathbf{l}'^T, \mathbf{l}''^T]^T) = 5$. By $rank(\mathbf{l}) = rank(\mathbf{l}') = rank(\mathbf{l}'') = 3$ and Lemma 5.5, it can be concluded that the rank of $\mathbf{A}$ is at most 26 for linear complex.

However, as in the step in proving Theorem (5.3) for ruled surface, three DoFs have to be deducted and the rank of $\mathbf{A}$ is at most $26 - 3 = 23$ for linear complex. As each line correspondence provides 2 linear constraints on $\mathbf{A}$, the minimum number of line correspondences needed is $\lceil 23/2 \rceil = 12$.

□

An alternative proof for the above has been presented by Stein and Shashua in [28]. The proof here is of a more general nature because it applies not only to special linear line complex but general linear line complex.

Table 5.1 summarizes the rank properties of all line structures this paper has presented.

## 5.3 Retrieving tensor in critical configurations

For linear line structures, such as ruled plane and linear line complex, although they are degenerated linear line structures such that trifocal tensor can not be uniquely determined, the tensor retrieved from the set containing all null vectors of estimation matrix $\mathbf{A}$ can transfer lines that belong to corresponding linear structures correctly, but not so for lines beyond such structures.

The trifocal tensor from linear line congruence can be recovered, as the estimation matrix $\mathbf{A}$ only ranks 19, it is possible to incorporate some of the 9 admissibility constraints to obtain a full recovery of trifocal tensor. From the rank property, linear line congruence is the only linear line structure beyond LLC that allows the computation of true trifocal tensor available, because the rank of tensor estimation matrix should be at least with rank $27 - 9 = 18$ for the unique determination of trifocal tensor.

**Algorithm for recovering trifocal tensor from linear congruence.**

1. Robust estimation to determine line correspondences which belong to the linear line congruence.
2. Use line-line-line correspondences from three view to compute the tensor estimation matrix $\mathbf{A}$.
3. Find the 8 vectors spanning the null space of $\mathbf{A}$.
4. The true trifocal tensor can be spanned by 8 null vectors with 7 scalars as parameters to be determined.

$$\mathbf{t} = \mathbf{t}_0 + \alpha_1\mathbf{t}_1 + \alpha_2\mathbf{t}_2 + \alpha_3\mathbf{t}_3 + \alpha_4\mathbf{t}_4 + \alpha_5\mathbf{t}_5 + \alpha_6\mathbf{t}_6 + \alpha_7\mathbf{t}_7 \tag{5.12}$$

where $\mathbf{t}$ is the true trifocal tensor, and $\mathbf{t}_0, \ldots, \mathbf{t}_7$ are the 8 null vectors of $\mathbf{A}$. $\alpha_1, \ldots, \alpha_7$ are the 7 scalars to be determined.

5. Find the above 7 scalars so that an admissible tensor is obtained using the admissibility constraints.

In practice, the last step starts with the solution from normalized linear method and refine it by numerical Gauss-Newton method until a solution satisfies the admissibility constraints is obtained.

## 5.4 Rank classification of non-linear line structures

By orthogonal plane representation of lines, the following propositions can be derived. Note that the results do not require any specific representation of lines, but the orthogonal plane representation can provide geometric insight of the structures. Here, we will not give the proof for rank classification of non-linear line structures.

The following proposition holds for nonlinear ruled surface,

**Proposition 5.1 (Nonlinear Ruled Surface).** *The rank of the tensor estimation matrix is at most* 26 *if the image observables are about a set of at least* 13 *lines that belong to a nonlinear ruled surface.*

**Remark 5.1.** *Empirical observations include rank* 11, 17, 19 *for* 2*-order ruled surface, rank* 22, 23 *for* 4*-order ruled surface and rank* 26 *for* 8*-order ruled surface respectively.

The following proposition holds for nonlinear line congruence,

**Proposition 5.2 (Nonlinear line congruence).** *The rank of the tensor estimation matrix is at most* 26 *if the image observables are about a set of at least* 13 *lines that belong to a nonlinear line congruence.*

**Remark 5.2.** *Empirical observations include rank* 23, 26 *for* 2*-order line congruence, rank* 26 *for* 4*-order line congruence respectively.*

The following proposition holds for nonlinear line complex,

**Proposition 5.3 (Nonlinear line complex).** *The rank of the tensor estimation matrix is at most* 26 *if the image observables are about a set of at least* 13 *lines that belong to a nonlinear line complex.*

**Remark 5.3.** *Empirical observations include rank* 26 *for* 2*-order line complex.*

# Chapter 6

# Camera Motion Estimation Framework

The goal of camera motion estimation is to determine camera projection matrices $\mathbf{P}$, from unknown motion acquired by uncalibrated camera, given the fact that we do not have direct access to 3-D in most cases. It is the foundation of real-world applications, such as ego-motion-determination (3-D position of the camera). Camera pose can be estimated by several classes of methods, such as stereo, structure and motion. The problem can be break down into two pieces - line tracking and computation of camera pose, given a set of line correspondences. We made the following reasonable assumptions:

- $\|[(\mathbf{R}_{1,f-1}, \mathbf{t}_{1,f-1}) - (\mathbf{R}_{1,f}, \mathbf{t}_{1,f})] - [(\mathbf{R}_{1,f}, \mathbf{t}_{1,f}) - (\mathbf{R}_{1,f+1}, \mathbf{t}_{1,f+1})]\|$ is small for all $f$, which means the camera motion is smooth.
- $\|(\mathbf{R}_{1f}, \mathbf{t}_{1f}) - (\mathbf{R}_{1,f+1}, \mathbf{t}_{1,f+1})\|$ is small - the imaging rate (video capture rate) is much faster than camera motion at all time, which implies that the image line $\mathbf{l}_{fl}$ in frame $f$ is close to the image line $\mathbf{l}_{f+1,l}$ in frame $f+1$.
- Linear features are sparse (only small amount of linear features are selected from the image sequence).
- scene depth/focal length is large - the scene is far away from the camera.

The following sections describe camera motion estimation framework in three parts: line extraction, line tracking and camera motion determination framework, as shown in

Fig. 6.1.

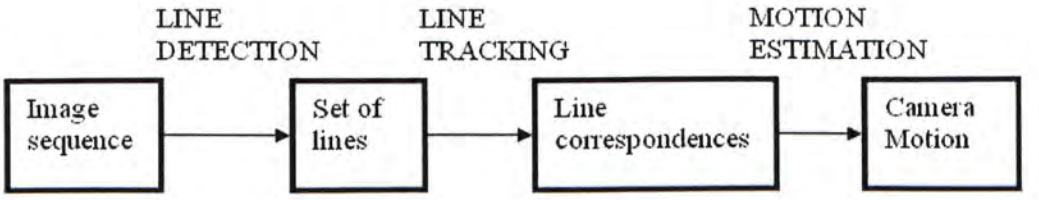



Figure 6.1: Overview of camera motion estimation.

## 6.1 Line extraction

An existing linear feature extraction software, called "*LINEAR*" [1] [60], is used for edge extraction as shown in Table 6.1. Briefly, it undergoes the following operations: edge detection (using convolution operator), edge thinning, edge linking, line fitting (to form super-segments and segments), then anti-parallel extraction. Image sequence can be processed automatically from *command line* input. Details about the output files of LINEAR can be found in its document. In particular, .ASEG file in ASCII format includes detailed information of line segments. "ShowLinear" can be used to display linear images (*.DAT). A powerful characteristic of this program is that disqualified segments and anti-parallels can be filtered out by adjusting the parameters in Filter Options, such as minimum segment length and minimum segment strength.

Table 6.1: LINEAR - Linear Feature Extraction Software.

**Function:**

1. Extracts the edges in the image. (.out files)
2. Finds the axial symmetries (anti-parallels) of the pairs of line segments that have opposite edge contrast.

**Input:** Gray level image

**Output:** Line segments of the image (.aseg files)

[1] Created by Davey Fong under the supervision of Prof. Ronald Chung

## 6.2 Line tracking

Line tracking problem can be formulated as,

**Remark 6.1** (Line tracking). *Given $F$ sets of lines, $\{\mathbf{l}_{1l}\}, \ldots, \{\mathbf{l}_{fl}\}, \ldots, \{\mathbf{l}_{Fl}\}$, it aims to find the mapping, such that $\{(\mathbf{l}_{1l}, \ldots, \mathbf{l}_{fl}, \ldots, \mathbf{l}_{Fl})\} \rightarrow \mathbf{l}$.*

The tracking method depends on the type of object (rigid or deformable), the degrees of freedom of objects and camera as well as the corresponding application. In 2-D tracking, the 3-D displacement is modeled as a 2-D transformation. "Adaptive Model" approach can be used to estimate perspectives or deformation. Recently, 3-D tracking has become the dominate method to recover $\mathbf{R}, \mathbf{t}$ in 3-D. In batch methods, image sequence is considered as a whole. Epipolar geometry plays an important role in matching points in two-view; likewise, trifocal tensor for three-view geometry is one of the fundamental tools of line tracking. As the geometry of lines in $\mathcal{P}^3$ is studied in chapter 3, our tracking method will be proposed directly.

### 6.2.1 Preliminary geometric tracking

Our method considers the homogeneous notion for points and lines on a plane. A line $\mathbf{l}$ in 2-D satisfies $ax + by + c = 0$; then, $\mathbf{l}$ can be represented by the vector $(a, b, c)^T$. As $(ka, kb, kc)$ also satisfies $(ka)x + (kb)y + (kc) = 0$, $(ka, kb, kc)$ and $(a, b, c)$ represent the same line. $(a, b, c)^T$ can be treated as the homogeneous representation of lines in 2-D. The point $\mathbf{x}$ lies on line $\mathbf{l}$ if and only if $\mathbf{x} \cdot \mathbf{l} = 0$. The intersection of two lines $\mathbf{l}$ and $\mathbf{l}'$ is the point $\mathbf{x} = \mathbf{l} \times \mathbf{l}'$. The line through two points $\mathbf{x}$ and $\mathbf{x}'$ is $\mathbf{l} = \mathbf{x} \cdot \mathbf{x}'$. The point $(x_1, x_2, 0)^T$ with last coordinate $x_3 = 0$ are known as *ideal point* , or point and infinity. *Line at infinity* is denoted by the vector $\mathbf{l}_\infty = (0, 0, 1)^T$.

**Theorem 6.1** (Duality Principle). *Duality principle states that to any theorem of 2-D projective geometry, there corresponds a dual problem, which may be derived by interchanging the roles of points and lines in the original theorem.*

A model for the projective plane is given in Fig. 6.2. To find the projective line representations, suppose $\mathbf{x}_1 = (x_1, y_1)$ and $\mathbf{x}_2 = (x_2, y_2)$ are two points on line $\mathbf{l} = (a, b, c)$, and $c$ can be assigned as any arbitrary value to make $[a, b]^T$ in a reasonable magnitude.

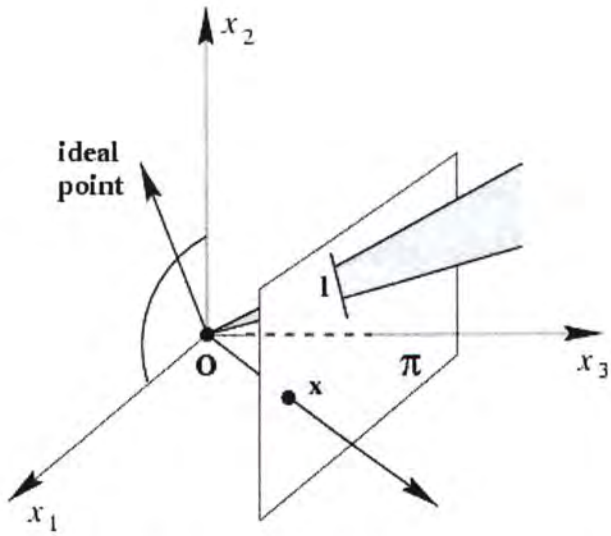



Figure 6.2: A model for the projective plane.

Then, Line is expressed as projective point on image plane in pinhole camera model as in Fig. 6.3(a). It can be found that,

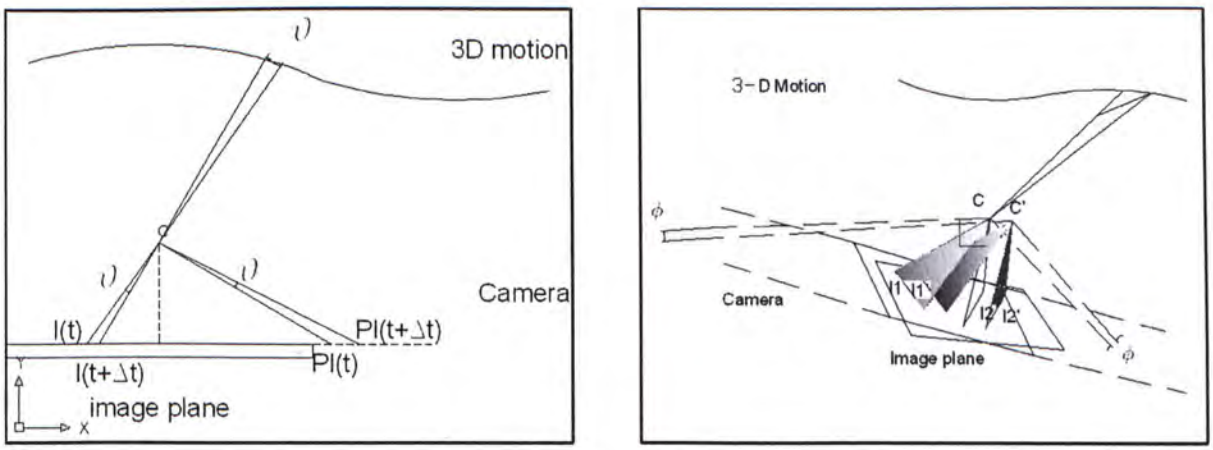
$$\begin{aligned} \mathbf{x}_1 \cdot \mathbf{l} &= 0 \\ \mathbf{x}_2 \cdot \mathbf{l} &= 0 \end{aligned} \tag{6.1}$$

which can be expanded as,

$$\begin{bmatrix} a \\ b \end{bmatrix} = - \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \end{bmatrix}^{-1} \begin{bmatrix} c \\ c \end{bmatrix} \tag{6.2}$$

Lines across multiple views can be tracked, which is the same as tracking their projective point representation. However, the remaining problems are the following, first, it is not geometrically meaningful to assume any motion model on projective points in inhomogeneous representation; second, if we want to track projective points in homogeneous representation, it is not linear with respect to its three coordinates $x_1$, $x_2$ and $x_3$. Thus, we have to look back to some more geometric originality of the problem. Kalman filter is used for tracking $\theta$ and $\phi$ as shown in Fig. 6.3(a) and (b) respectively. $\theta$ is the angle resulted from vertical rotation and translation of the camera; $\phi$ is the angle resulted from horizontal rotation and translation of the camera.

In our model, the state vector denoted by $\mathbf{X}_k$ consists of two angle parameters and is




(a) (b)

Figure 6.3: Projective line representation, motion parameter $\theta$ and $\phi$ in 2-D and 3-D.

defined as

$$\mathbf{X}_k = \begin{bmatrix} \theta & \phi & \dot{\theta} & \dot{\phi} \end{bmatrix}^T \tag{6.3}$$

where $\dot{\theta}$, $\dot{\phi}$ are the velocities of the angle parameters $\theta$ and $\phi$ respectively. From the assumption that the scene is far away from the camera, these two angles, as shown in Fig. 6.3, are capable in describing the position of the camera relative to the scene. Initial guess can be computed from nearest neighbor method or RANSAC robust estimation method.

The linear dynamic model in Kalman Filtering is considered as state transition and measurement equations respectively

$$\begin{aligned} \mathbf{X}_k &= \mathbf{A}_{k-1}\mathbf{X}_{k-1} + \mathbf{q}_{k-1} \\ \mathbf{Y}_k &= \mathbf{H}_k\mathbf{X}_k + \mathbf{r}_k \end{aligned} \tag{6.4}$$

where $\mathbf{q}_{k-1} \sim N(0, \mathbf{Q}_{k-1})$, $\mathbf{r}_k \sim N(0, \mathbf{R}_k)$ are process noise and measurement noise respectively. As there are $N$ linear features extracted from each frame, the number of KF is also $N$.

$\mathbf{A}$ is the $4 \times 4$ diagonal state transition matrix, and $\mathbf{H}$ is the measurement matrix. $\mathbf{A}$

and $\mathbf{H}$ are respectively defined as

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{6.5}$$

Discrete Kalman filter time update equations and measurement update equations will be used to estimate the projective point representation of lines.

$$\begin{aligned} \hat{\mathbf{x}}_k^- &= \mathbf{A}\hat{\mathbf{x}}_{k-1} \\ \mathbf{P}_k^- &= \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{Q} \end{aligned} \tag{6.6}$$

$$\begin{aligned} \mathbf{K}_k &= \mathbf{P}_k^-\mathbf{H}^T(\mathbf{H}\mathbf{P}_k^-\mathbf{H}^T + \mathbf{R})^{-1} \\ \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k^-) \\ \mathbf{P}_k &= (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_k^- \end{aligned} \tag{6.7}$$

$\mathbf{Q}$ and $\mathbf{R}$ are the covariance matrix for process noise and measurement noise respectively,

$$\mathbf{Q} = 0.1 \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{6.8}$$

Kalman filter starts from the third view. After the kalman prediction, a local search is applied to find a unique match. If no line is found in the local search, then we assume that this line feature is no longer available.

As the coordinate frame for lines is known, then we can compute the angle-adjusted shortest distance $d$ between each pair of line segments (reminded that length of line segment is finite). If $d$ is smaller than a given threshold, the parameter of line detection system (*LINEAR*) can be tuned to extract less line segments from each frame to avoid such case. We do not need as many line correspondences as point for tracking.

To summarize, a geometric meaningful line tracking algorithm is proposed by merging kalman filter and 3-view geometry. The proposed tracking algorithm is simple and fast, and it does not use any 3-D information of the scene. We also plan to include the depth $d$ and the velocity of $d$ as two additional variables in the state vector. In future, it can be extended to a Bayesian Network framework concerning the relationship of points in 2-D, or incorporated with 3-D geometry of lines. However, there are several limitations that we should consider. The sample rate can not be either too slow or too fast, simply because in over-dense image sequences, the measurement noise will become dominant. The image sequence that is currently used only contains 16 images. Later, we will use a longer image sequence to test our method. Besides, other soft features may be applied, such as the edge gradient.

### 6.2.2 Experimental results

Real images experiment is performed. Image sequence with resolution $480 \times 512$ comes from $C$MU VASC Image Database taken by SONY XC57 camera with NIKON $35-105$mm zoom lens, fixed 50mm, focus infinity. The distances are as follows: the camera to the nearest parking meter: $39'6"$; the camera to the 2nd parking meter: $60'7"$; the camera to the 3rd parking meter: $82'4"$; the camera to the farthest corner of the building : $112'2"$. It contains 15 frames.

The images come in PNG format, and they are converted to BMP format (A=imread ('pmseq0.png'); imwrite(A,'pmseq0.bmp');). Then, *LINEAR* AnyToRaw utility converts the image format from BMP to DAT. *LINEAR* is used for line detection (Linear -img pmseq0.dat). The detection results are controllable by adjusting the parameters in *Filter Options*, such as minimum segment length and minimum segment strength. Sample result from *LINEAR* is shown in Table 6.2. ".aseg" files record the information of all line segments, but it is less convenient. The results from the command line are more straightforward.

Every two points (Begins, Ends) represent one line segment. Line representation can be transformed to its corresponding projective point representation, which is $(a, b)$ in inhomogeneous form. The point-form lines can be tracked using Kalman Filter: $\mathbf{X}_k = (\theta_k, \phi_k, \dot{\theta}_k, \dot{\phi}_k)$, in which $\theta$ and $\phi$ are the only two observables.

Table 6.2: Results from line extraction using LINEAR - Part of .aseg file.

| Segment | Fam | Pred | Succ | Begins | Ends | Length | Theta | Strength |
|---|---|---|---|---|---|---|---|---|
| 6 | 5 | 0 | 7 | (3, 86) | (191, 86) | 188.00 | 0.00 | 40.54 |
| 48 | 33 | 0 | 0 | (3, 492) | (116, 488) | 113.07 | 357.97 | 86.23 |
| 49 | 34 | 0 | 0 | (3, 503) | (78, 501) | 75.03 | 358.47 | 31.27 |
| 50 | 35 | 0 | 0 | (3, 510) | (41, 510) | 38.00 | 0.00 | 46.55 |
| 596 | 414 | 0 | 597 | (45, 510) | (163, 510) | 118.00 | 0.00 | 73.89 |
| 715 | 494 | 714 | 716 | (65, 373) | (249, 370) | 184.02 | 359.07 | 38.21 |
| 717 | 495 | 0 | 718 | (54, 411) | (3, 273) | 147.12 | 249.72 | 34.04 |
| 720 | 495 | 719 | 0 | (3, 225) | (197, 222) | 194.02 | 359.11 | 51.13 |
| 800 | 540 | 799 | 0 | (54, 414) | (3, 414) | 51.00 | 180.00 | 39.63 |

Nearest neighborhood method is applied to establish the initial matching. The initial value of $\mathbf{X}$ is computed as $\mathbf{X}_1 = [-116.27907, 0, 1, 0]^T$. The initial value of $\mathbf{Q}$ is

$$\mathbf{Q}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.9}$$

Kalman Filtering predicts the position of the point-form lines ($kalman\_filter(x, A, H, Q, R, initX, initQ)$). In addition, to reduce the error caused by measurement noise, we also apply Kalman smoothing technique to achieve the smooth motion of the points ($kalman\_smoother(x, A, H, Q, R, initx, initQ)$). Both "kalman_filter" and "kalman_smoother" results are shown in Fig. 6.4.

Then, we use local search within a circle whose radius is determined by the norm of the covariance matrix of the estimation to find the actual position of lines. The correspondences are then established iteratively. The result shows that most lines that exist in all frames can be tracked correctly, and it is fast.

**Integrating trifocal tensor.** Trifocal tensor can be integrated into the line tracking. Line transfer from two images to the third one revealed in [61] can be applied. Line correspondences and trifocal tensor can be integrated to the state vector of the Kalman filter system such that the estimate of trifocal tensor can be updated from data in new image frame. As trifocal tensor tells the camera motion up to projectivity, this treatment coincides with our primary motion estimation objective and leads to the camera pose

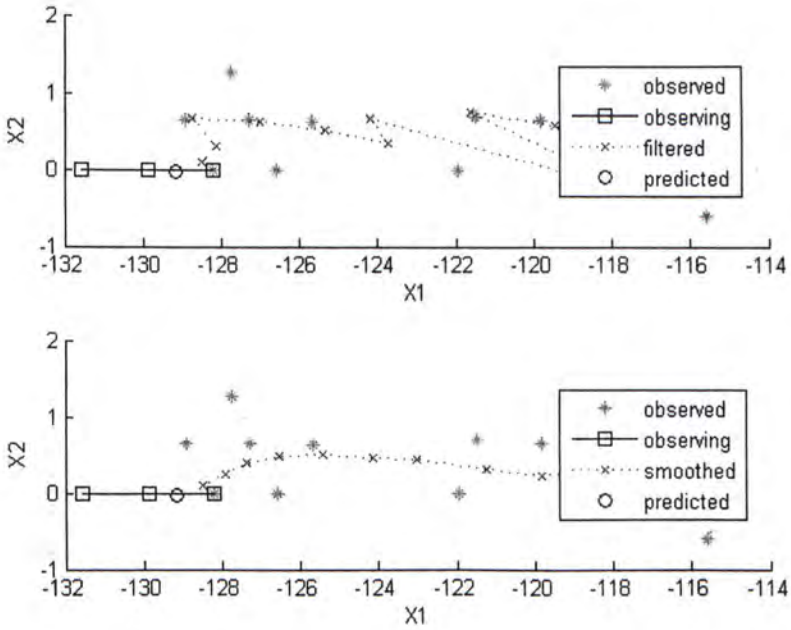



Figure 6.4: Prediction from Kalman filter and Kalman smoother.

estimation framework as illustrated in the next section.

Finally, The line tracking over frames is shown in Fig. 6.5.

## 6.3 Camera motion estimation framework using EKF

**NOTE:** *This part is only about the framework of camera motion estimation. The system have not been fully implemented yet.*

**Remark 6.2** (Camera motion estimation). *Find* $(\mathbf{R}_{12}, \mathbf{t}_{12}), \ldots, (\mathbf{R}_{1f}, \mathbf{t}_{1f}), \ldots, (\mathbf{R}_{1F}, \mathbf{t}_{1F})$, *such that for all* $l$, *its correspondences in the image sequence* $(\mathbf{l}_{1l}, \ldots, \mathbf{l}_{fl}, \ldots, \mathbf{l}_{Fl})$ *intersect in* 3*-D at the same* 3*-D line segment under transformation* $\{(R_{1f}, t_{1f}), f = 2, \ldots, F\}$, *where* $F$ *is the total number of frames, and* $l$ *is the index of lines.*

In motion estimation, several choices should be made before designing the algorithm, such as optimal flow or geometry, point feature or line feature, recursive or non-recursive method, two-view or multiple-view. We choose to use line feature, recursive, three-view geometry-based camera motion estimation with trifocal tensor as in Fig. 6.6.

The way to create a motion model depends on how to parameterize camera pose. In Euler angles, a rotation matrix $\mathbf{R}$ can always be factorized as the product of three

(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 6.5: Line tracking over 12 frames.

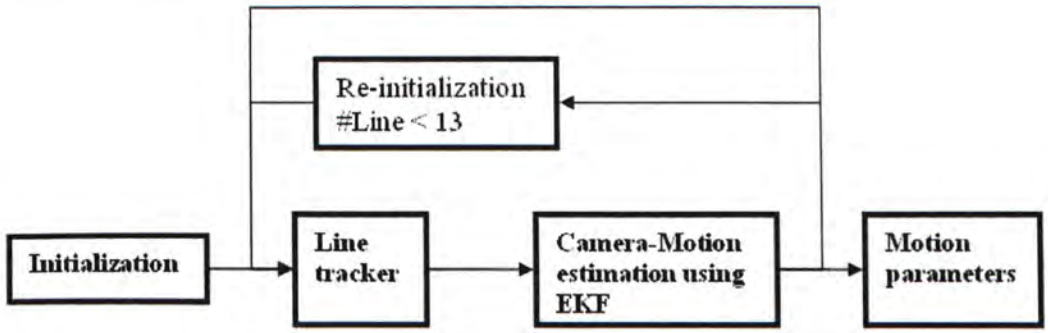



Figure 6.6: Camera motion estimation framework using line correspondences with the trifocal tensor.

matrices representing rotations around the $X$, $Y$, and $Z$ axes as in Eqn. (3.7).

Constant velocity model is assumed for camera motion. The state vector includes translation vector $(t_x, t_y, t_z)$ rotation matrix parameters $(\alpha, \beta, \gamma)$ as

$$\mathbf{x}_k = [t_x, t_y, t_z, \alpha, \beta, \gamma, \dot{t}_x, \dot{t}_y, \dot{t}_z, \dot{\alpha}, \dot{\beta}, \dot{\gamma}, x_2, y_2, z_2, \alpha_2, \beta_2, \gamma_2]^T \tag{6.10}$$

where $\dot{t}_x, \dot{t}_y, \dot{t}_z, \dot{\alpha}, \dot{\beta}, \dot{\gamma}$ are the velocities of the camera pose parameters. $x_2, y_2, z_2, \alpha_2, \beta_2, \gamma_2$ are the pose parameters with respect to the second camera. As initial estimate obtained by nearest neighborhood method or RANSAC method is not accurate, and also because the baseline between frame 1 and 2 is short, it is necessary to include the pose information of frame 2 into the state vector.

$$\begin{aligned} \mathbf{x}_k &= f(\mathbf{x}_{k-1}) + \mathbf{w}_{k-1}) \\ \mathbf{z}_k &= h(\mathbf{x}_k) + \mathbf{v}_k \end{aligned} \tag{6.11}$$

where $\mathbf{w}_{k-1}$, $\mathbf{v}_k$ are the system noise and measurement noise respectively. $\mathbf{z}_k$ is a $2N \times 1$ vector. $h(\cdot)$ is the $2N \times 1$ function determined by line transfer using trifocal tensor. Next, EKF prediction and update equations can be used to obtain optimal estimation.

To summarize, we have illustrated recursive camera motion estimation framework based on tracking lines across views integrating kalman filter and trifocal tensor. As mentioned before, the three-view geometry plays an important role in structure and motion. First, it makes consistent of geometric parameters across three views, which is unfor-

tunately not directly satisfied for fundamental matrix in two views; Secondly, the time complexity for algorithms based on three-view is allowable, in contrast to four or more views, where it is not only hard to analyze but also time-consuming.

In future, our framework can be implemented, so that camera motion will be computed with respect to the coordinate frame of the first camera. Next, the sparse lines under tracking with known camera motion will be matched with the known 3-D models of buildings to obtain the camera position and motion with respect to an arbitrary world coordinate frame in Euclidean space.

# Chapter 7

# Experimental Results

Here we present results on both simulated and real image data to illustrate the validity of the above findings. To present the rank property of the estimation matrix $\mathbf{A}$ in each experiment, we display the magnitudes of the various singular values of $\mathbf{A}$ extracted by the Singular Value Decomposition (SVD) mechanism. Simple visual examination of the relative magnitudes of the singular values would already allow the true rank of the matrix to be revealed despite the presence of influence from resolution error and image noise (in the case of real image experiments).

## 7.1 Simulated data experiments

We have conducted massive empirical experiments to examine the proposed rank properties of the tensor estimation matrix. In such experiments image noise was absent, making the rank property more apparent, though resolution error was allowed to be present and have effect.

The way we generated the simulated data is summarized in Table 7.1.

**Linear line space.** Line pencil, point star, and ruled plane were generated in 3-D, and three views of 50 lines of them were created and they are shown in Fig. 7.1(a), 7.2(a), 7.3(a) respectively. Fig. 7.1(b), 7.2(b), 7.3(b) show the logarithmic values of all singular values for these three line structures. Simple visual check already shows that the rank of the estimation matrix was $7, 11, 15$ respectively.

**Linear ruled surface.** Linear ruled surface was generated in 3-D, and three views

Table 7.1: Simulation process.

| |
|---|
| 1.Generate 3-D lines and form the matrix **L**. |
| 2.For each of the three cameras, choose its position in space and the Euler angles for its orientation; then generate three camera matrices using the camera model described in Chapter 3. |
| 3.Calculate the line projection matrices using Eqn. (3.9). |
| 4.Calculate the image lines using Eqn. (5.4). |
| 5.Determine tensor estimation matrix **A** using Eqn. (5.6). |
| 6.Apply SVD to **A** to let its rank be visually examined. |

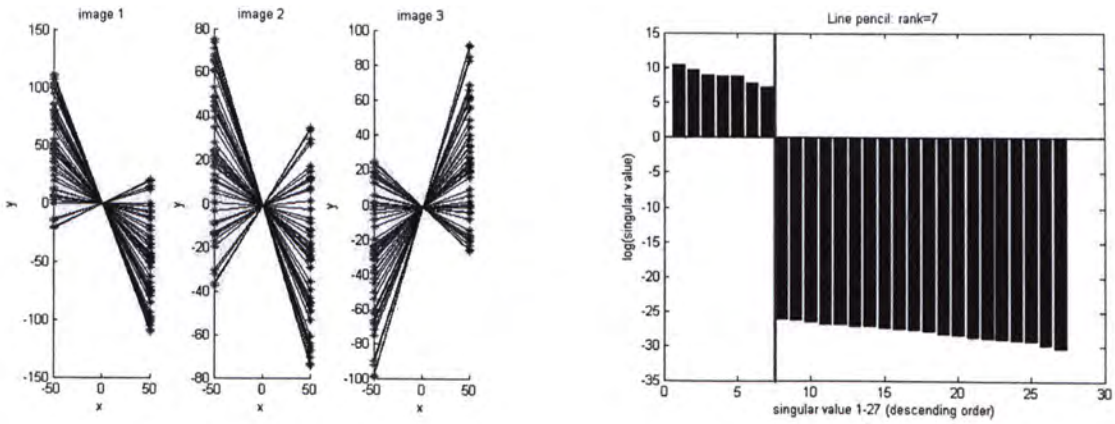
of 50 lines of it were created and they are shown in Fig. 7.4(a). Fig. 7.4(b) shows the logarithmic values of all singular values. It is clear visually that the rank of the estimation matrix was 12.

**Linear line congruence.** Linear line congruence was generated in 3-D, and three views of 50 lines of it were created and they are shown in Fig. 7.6(a). Fig. 7.6(b) shows the logarithmic values of all singular values. Visual check shows that the rank of the estimation matrix was 19. Helicoid is a special line congruence shown in Fig. 7.5. Rank of estimation matrix is 19.

**Linear line complex.** Linear line complex was generated in 3-D, and three views of 50 lines of it were created and they are shown in Fig. 7.7(a). Fig. 7.7(b) shows the logarithmic values of all singular values. Visually it is obvious that the rank of the estimation matrix was 23.

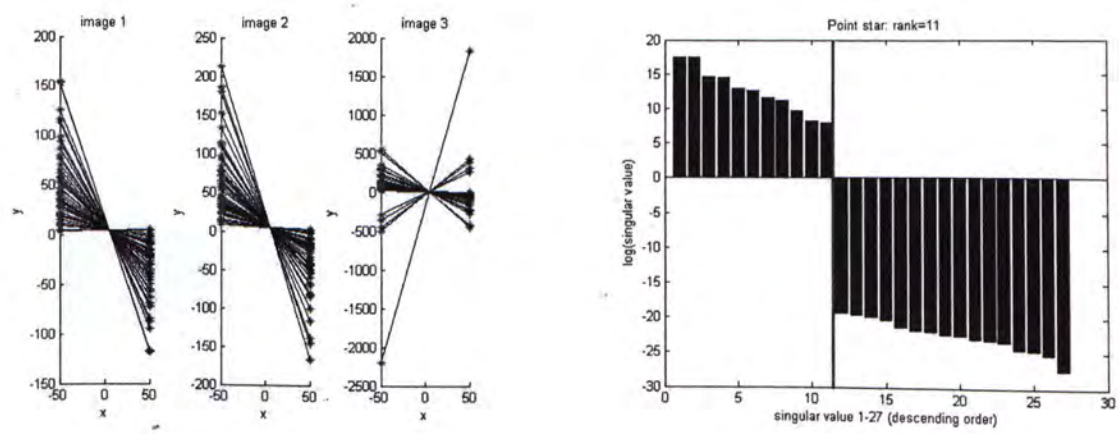
## 7.2 Real data experiments

The scene was composed of a cube and square board, with square grids covering every surface of the scene. The images were captured by a consumer-grade camera (Fuji FinePix S602 Zoom camera) with image resolution $2048 \times 1536$. Image lines were precisely located by having two widely separated grid-corner points identified on each of them, and line correspondences were manually established carefully. The grid-corner points came from four different planes: three faces of the cube and one square board behind the cube. Lines and subsequently line correspondences could be established from points within a plane



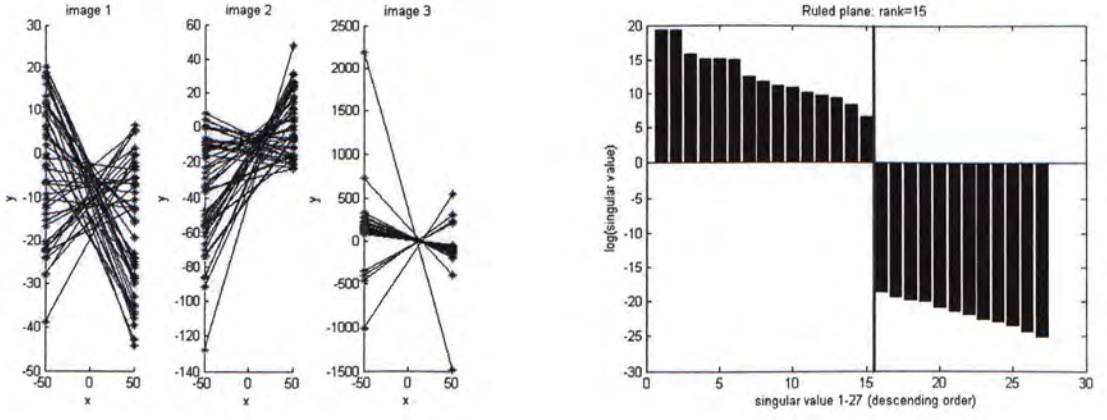

(a) (b)

Figure 7.1: Three views of 50 lines that belong to a line pencil: $rank(\mathbf{A}) = 7$; (a) the three views; (b) magnitudes of all singular values of the tensor estimation matrix.



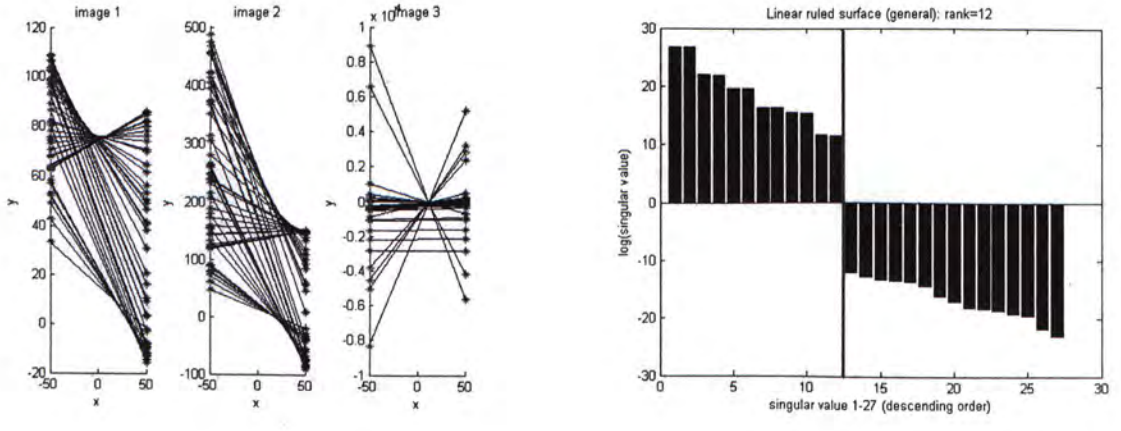

(a) (b)

Figure 7.2: Three views of 50 lines that belong to a point star: $rank(\mathbf{A}) = 11$; (a) the three views; (b) magnitudes of all singular values of the tensor estimation matrix.



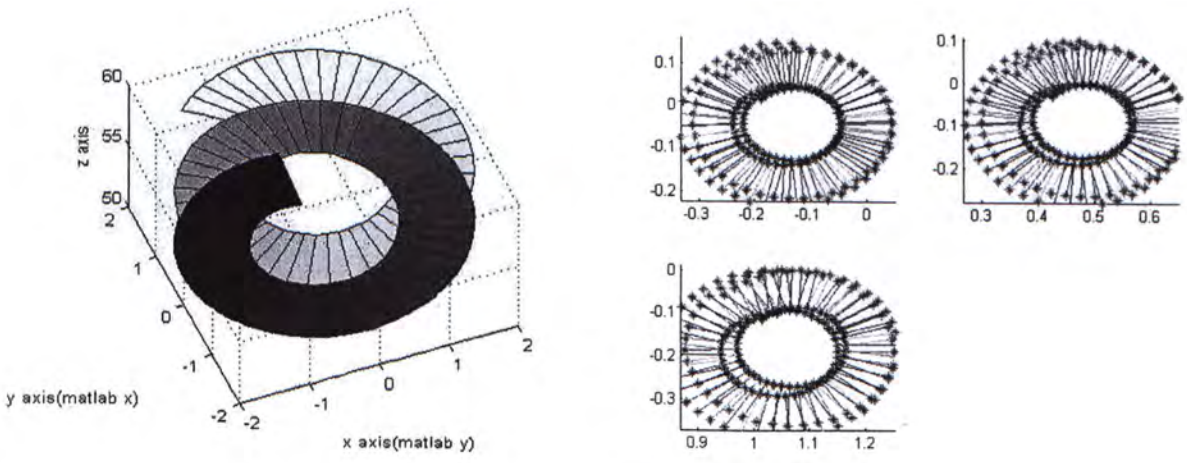

(a) (b)

Figure 7.3: Three views of 50 lines that belong to a ruled plane: $rank(\mathbf{A}) = 15$; (a) the three views; (b) magnitudes of all singular values of the tensor estimation matrix.



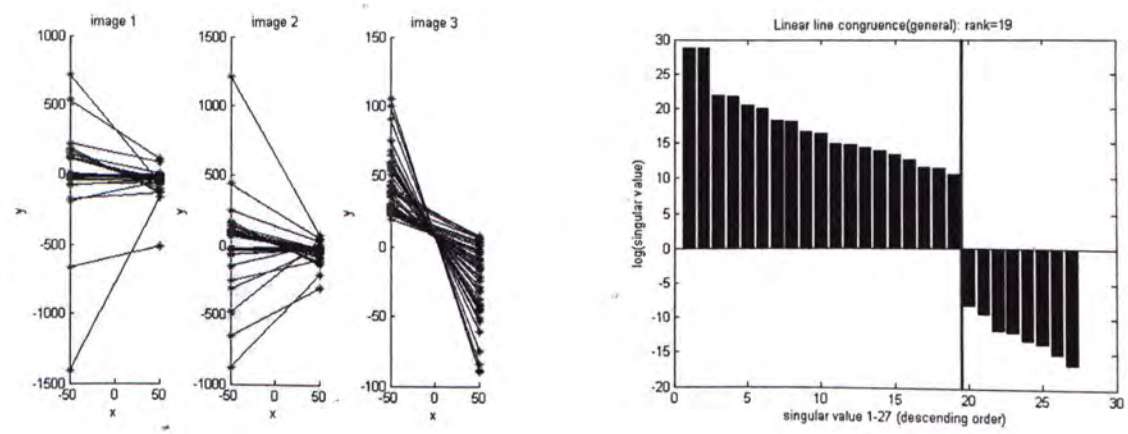

(a) (b)

Figure 7.4: Three views of 50 lines that belong to a linear ruled surface: $rank(\mathbf{A}) = 12$; (a) the three views; (b) magnitudes of all singular values of the tensor estimation matrix.



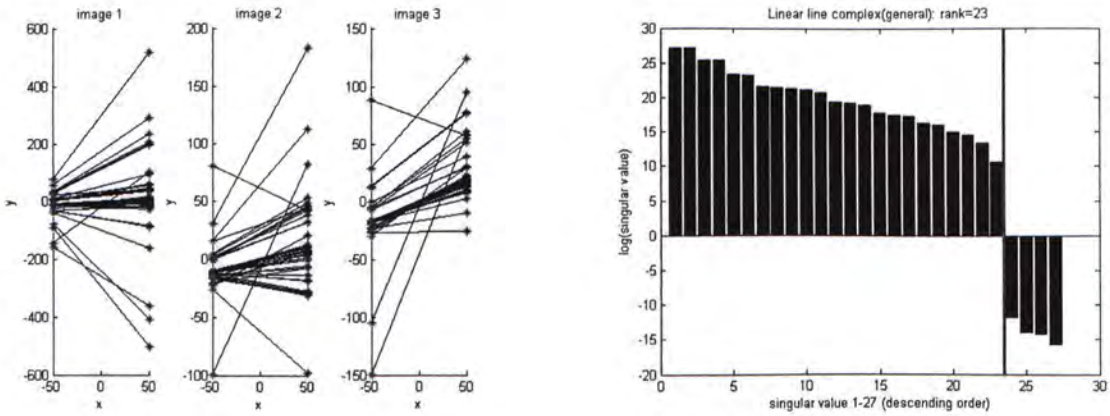

(a) (b)

Figure 7.5: Helicoid and its images in three-view




(a) (b)

Figure 7.6: Three views of 50 lines that belong to a linear congruence: $rank(\mathbf{A}) = 19$; (a) the three views; (b) magnitudes of all singular values of the tensor estimation matrix.




(a) (b)

Figure 7.7: Three views of 50 lines that belong to a linear complex: $rank(\mathbf{A}) = 23$; (a) the three views; (b) magnitudes of all singular values of the tensor estimation matrix.

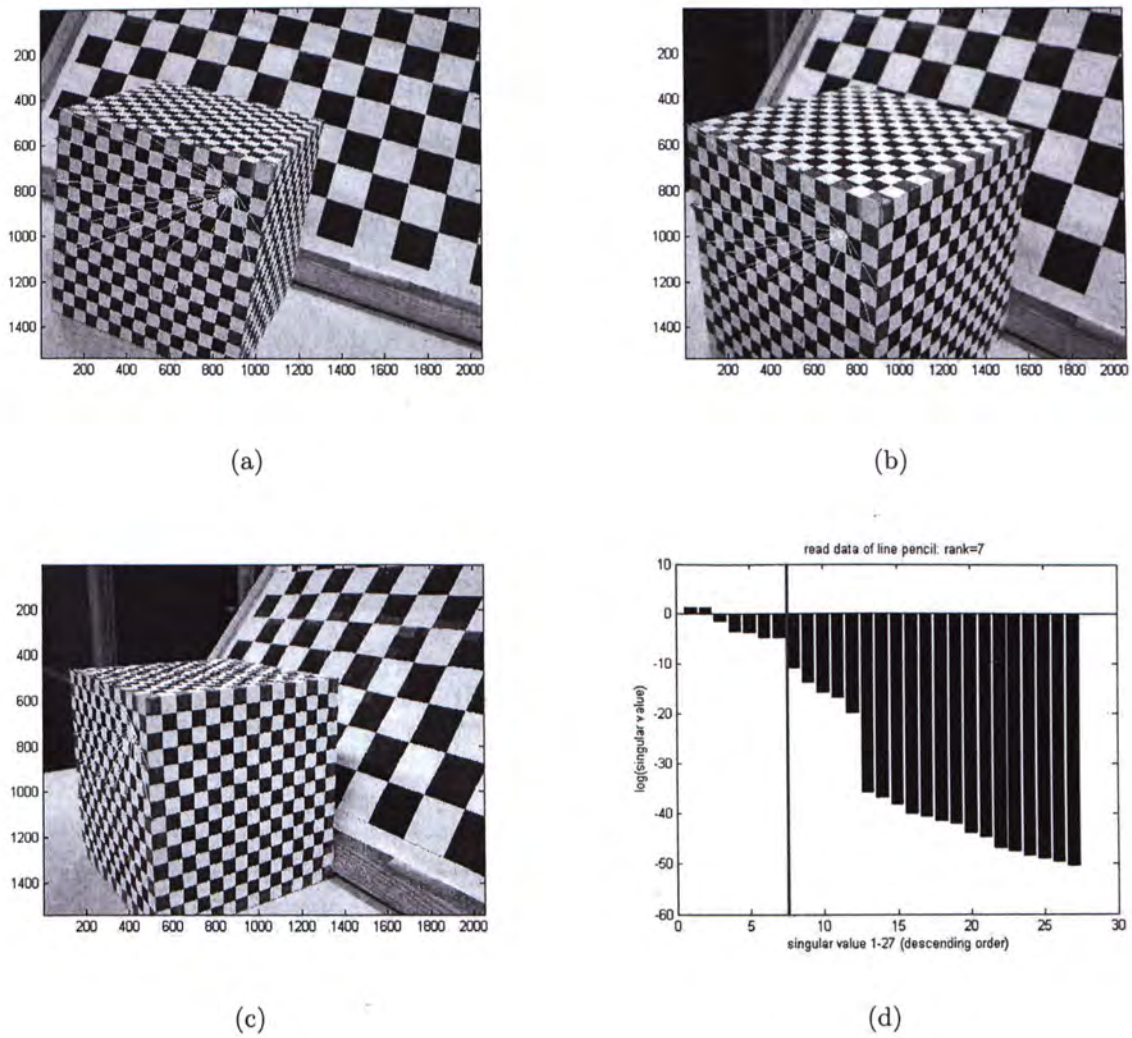
as well as across different planes. This way, lines could form a number of different line structures. Line pencil was formed by taking all points(lines) that co-intersected in one plane. Point-star was the family of lines joining points from one plane and points off the plane. Ruled plane was formed by lines that joined only coplanar points. Besides, for any particular family of lines, new lines that were outside the family could also be arbitrarily generated. Thus the scene setting, though simple, did allow all linear line structures we consider in this work to be established.

In all experiments, once the image lines and the line correspondences were established, the normalized linear algorithm was used to estimate the trifocal tensor.

**NOTE:** *While any family of lines we consider in this work are not bounded by any particular sector of the 3-D space, any line set visible in the image data were (they were all inside the sector of 3-D space where the visual fields of the three cameras overlapped). Thus it could be expected that the tensor estimation matrix* $\mathbf{A}$ *of the image lines could be under-ranked, i.e., having a rank less than the upper bounds presented in Chapter 5.*

### 7.2.1 Linear line space

**Line pencil.** Three views of 15 lines of a line pencil were captured in Fig. 7.8(a),(b),(c). The singular values of the tensor estimation matrix as determined from the real image




(a) (b)

(c) (d)

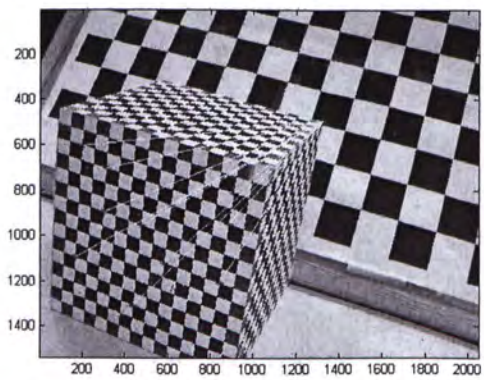
Figure 7.8: Three views of 15 lines of a line pencil: $rank(\mathbf{A})$ should be 7: (a),(b),(c): the three views, (d) the singular values of the tensor estimation matrix.

data is displayed in Table 7.2. Fig. 7.8(d) shows the logarithmic values of singular values.

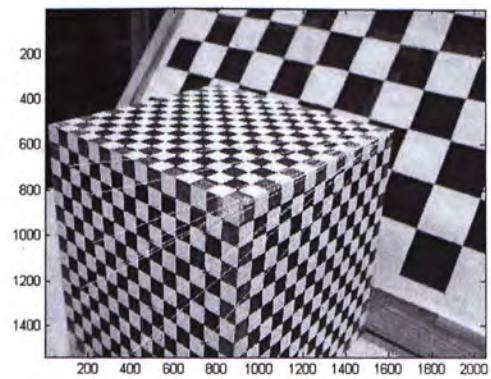
Notice that there was a sharp decrease from the $7th$ largest singular value to the $8th$ largest singular value, and the ratio between their magnitudes was only 0.002. It is consistent with the proposed result that the rank of the estimation matrix is no more than 7.

**Point star.** Three views of 15 lines of a point star were captured in Fig. 7.9(a),(b),(c). The singular values of the tensor estimation matrix as determined from the real image data is given in Table 7.3. Fig. 7.9(d) shows the logarithmic values of singular values.

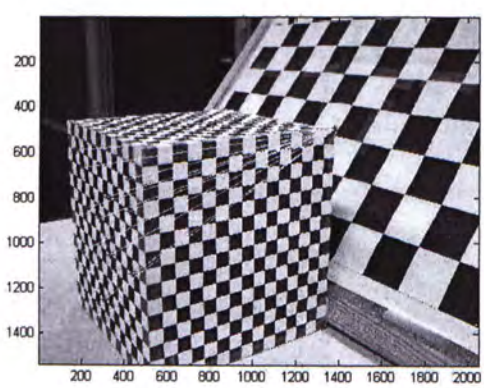
Notice that there was a sharp decrease from the $11th$ largest singular value to the $12th$



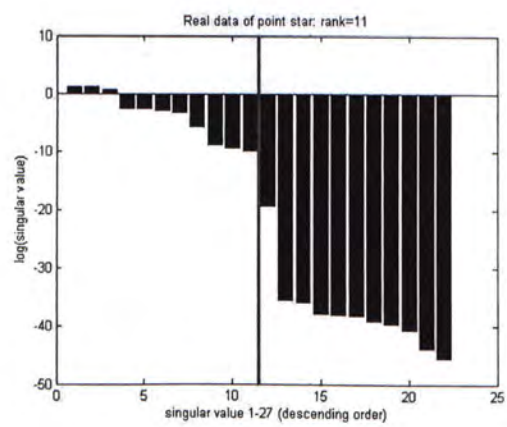

(a)




(b)




(c)




(d)

Figure 7.9: Three views of 11 lines of point star: $rank(\mathbf{A})$ should be 11: (a),(b),(c): the three views, (d) the singular values of the tensor estimation matrix.

Table 7.2: Line pencil: The first 9 largest singular values of the tensor estimation matrix in descending order.

| 1 | 2 | 3 |
|---|---|---|
| 3.87391415456492 | 3.87300652270457 | 0.25289777951591 |
| **4** | **5** | **6** |
| 0.03040938703019 | 0.02976519813995 | 0.01067095306172 |
| **7** | **8** | **9** |
| 0.01060214514503 | 0.00002074615175 | 0.00000118752174 |

Table 7.3: Point-star: The first 12 largest singular values of the tensor estimation matrix in descending order.

| 1 | 2 | 3 |
|---|---|---|
| 3.37856000778844 | 3.31704866741561 | 1.76964758669184 |
| **4** | **5** | **6** |
| 0.07929898313645 | 0.07925389410443 | 0.05977953639001 |
| **7** | **8** | **9** |
| 0.03721143608304 | 0.00361908958561 | 0.00013670897335 |
| **10** | **11** | **12** |
| 0.00008674288015 | 0.00005252513361 | 0.00000000400253 |

largest singular value, and the ratio between their magnitudes was only 0.0000762. It is consistent with the proposed result that the rank of the estimation matrix is no more than 11.

**Ruled plane.** Due to measurement noise and the relatively small scope of space the line set was contained in, unlike the above, the actual rank of the tensor estimation matrix in this case was not as obvious from the magnitudes of the singular values. We verified the proposed upper bound of the rank in a different way. We used a line transfer mechanism as summarized in Table 7.4. Fig. 7.11 shows the line transfer result. A trifocal tensor was determined from merely the first 15 largest singular values of the tensor estimation matrix. Line transfer using this trifocal tensor could still re-project lines from the second and third views to the first view correctly, meaning that the rank of the tensor estimation matrix was indeed no more than 15, as predicted by the described analysis. However, upon having the estimation matrix's rank reduced further to less than 15, the line transfer results (as

Table 7.4: Checking the rank property of a linear line structure using line transfer.

1. Choose a line set of the line structure from the image data.
2. Use the image lines in the three views to construct the tensor estimation matrix $\mathbf{A}$, and find the singular values of this matrix $\mathbf{A}$.
4. Suppose that the described rank analysis says the rank of $\mathbf{A}$ for the particular line structure is no more than $r$. Using SVD, keep only the largest $r$ singular values of $\mathbf{A}$, thereby reducing matrix $\mathbf{A}$ to matrix $\mathbf{A}'$ that is of rank only $r$.
5. Determine the trifocal tensor of the three views using the reduced matrix $\mathbf{A}'$ as the tensor estimation matrix.
6. Using the trifocal tensor so obtained, by means of Eqn. (7.1), transfer the above image lines $\mathbf{l}'$, $\mathbf{l}''$ in view 2 and 3 respectively to view 1.
7. How close the transferred image lines are to the actual image lines in view 1 indicate whether the proposed rank property is correct or not.

shown in Fig. 7.11(b-d) deteriorated quite dramatically from (b) to (d), showing that for a ruled plane the rank of the tensor estimation $\mathbf{A}$ was no more than 15 but close to 15.

The experiments for linear ruled surface, linear line congruence, and linear line complex are conducted in the same way.

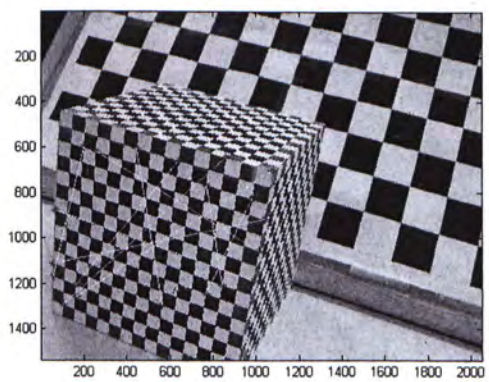
### 7.2.2 Linear ruled surface

**Linear ruled surface.** As shown in Fig. 7.13, the line transfer error using tensor approximated from respectively 12, 11 and 10 largest singular values of the estimation matrix were small, indicating that the rank of the estimation matrix $\mathbf{A}$ was indeed less than 12, as predicted.

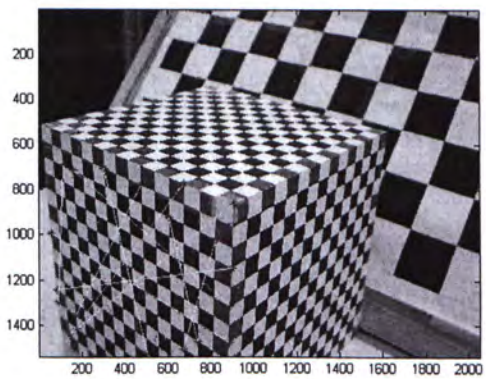
**Nonlinear ruled surface.** Rank of estimation matrix is up to 26.

### 7.2.3 Linear line congruence

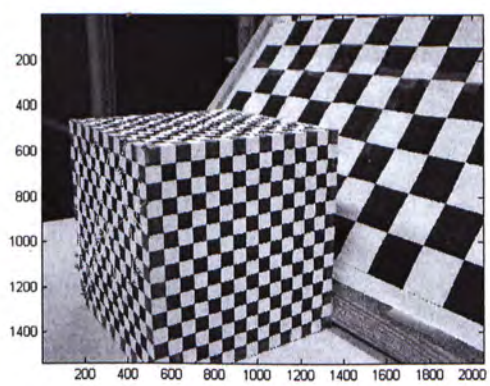
**Linear line congruence.** As shown in Fig. 7.15, line transfers using tensor approximated from 19 largest singular values of the estimation matrix $\mathbf{A}$ were correct, but results from 18 singular values contained an incorrect line, indicating that the rank of $\mathbf{A}$ was about 19. This was consistent with the described rank analysis.



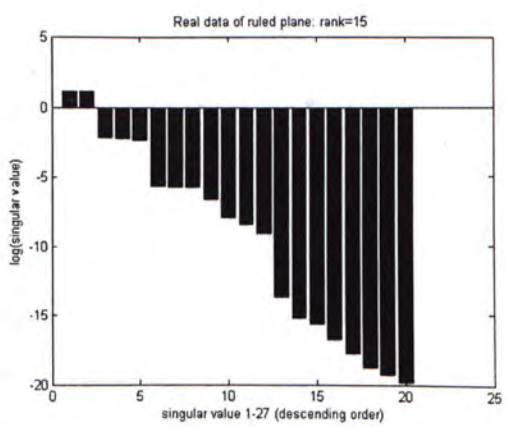

(a)




(b)




(c)




(d)

Figure 7.10: Three views of 10 lines of ruled plane: $rank(\mathbf{A})$ should be 15: (a),(b),(c): the three views, (d) the singular values of the tensor estimation matrix.

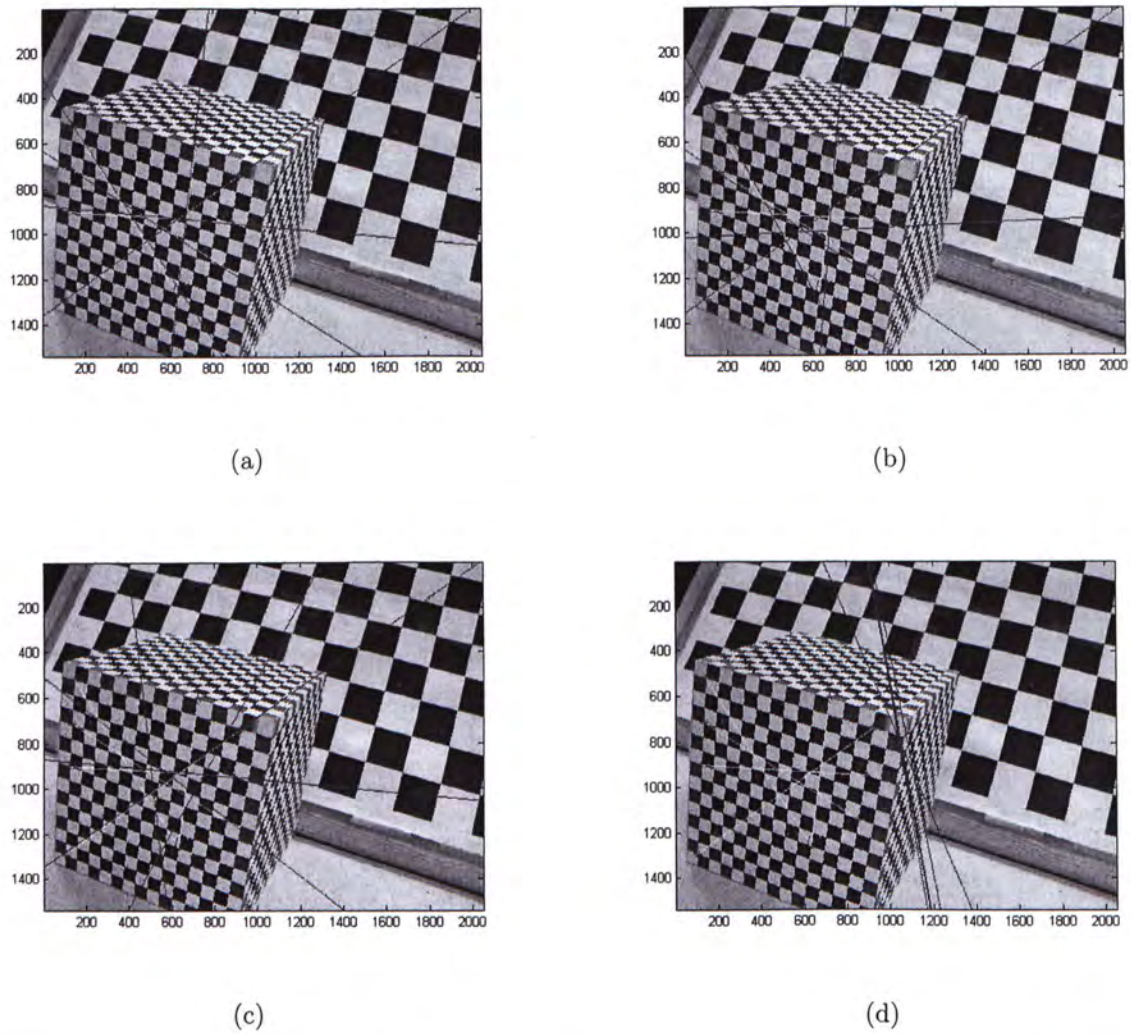
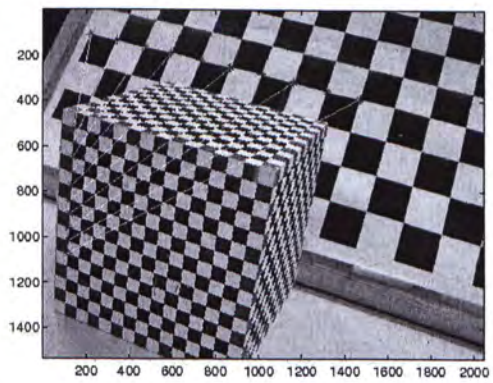
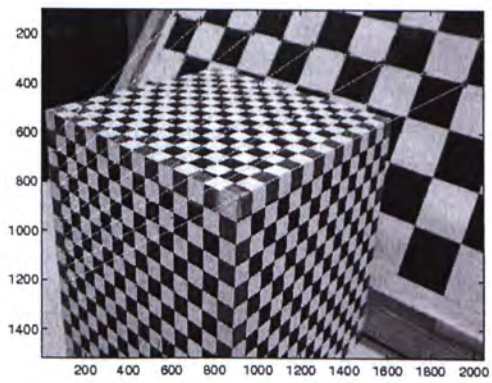


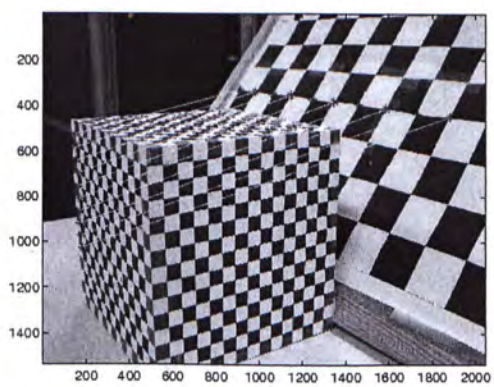

Figure 7.11: Line transfer from the second and third views of a ruled plane (the frontal surface of the box) ($rank(\mathbf{A}) = 15$) to the first view by trifocal tensor that was determined respectively from (a) 15, (b) 14, (c) 13, and (d) 5 largest singular values of the estimation matrix.



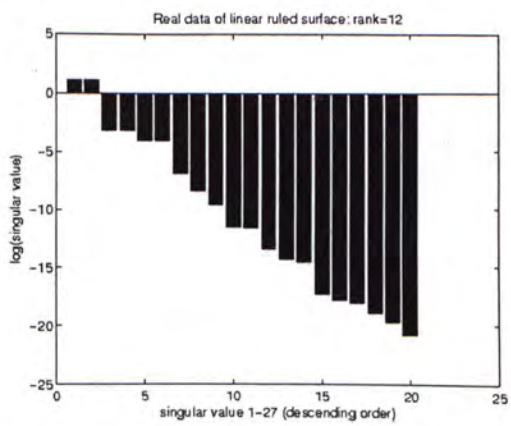

(a)




(b)

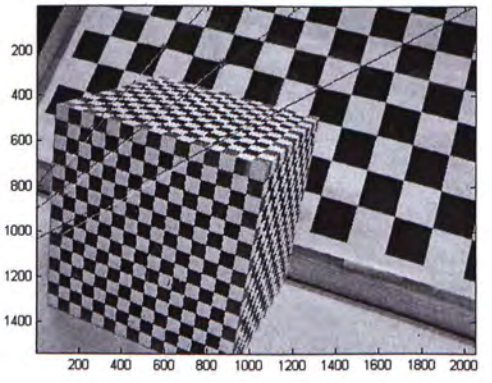



(c)

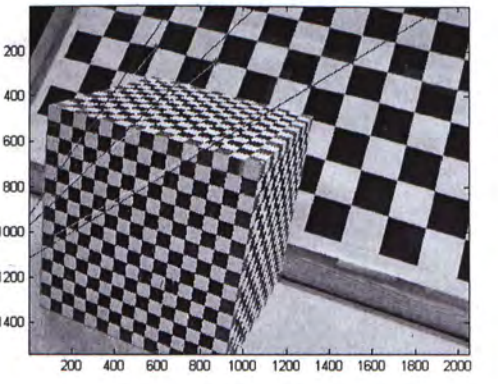



(d)

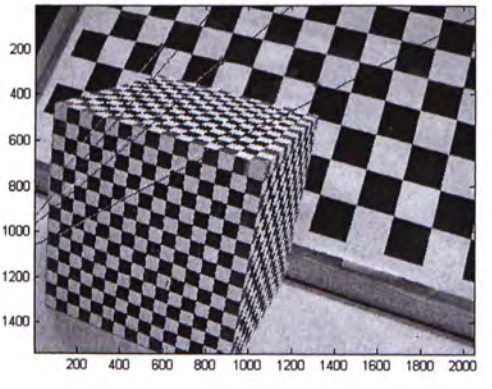
Figure 7.12: Three views of 10 lines of linear ruled plane: $rank(\mathbf{A})$ should be 12: (a),(b),(c): the three views, (d) the singular values of the tensor estimation matrix.



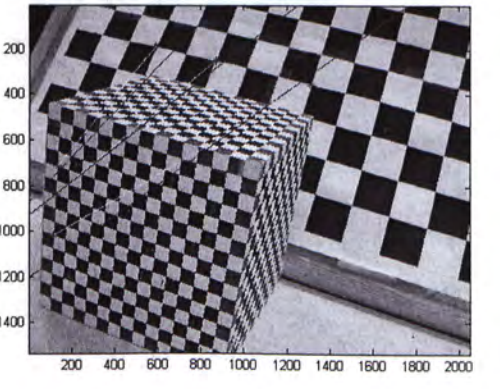

(a)




(b)

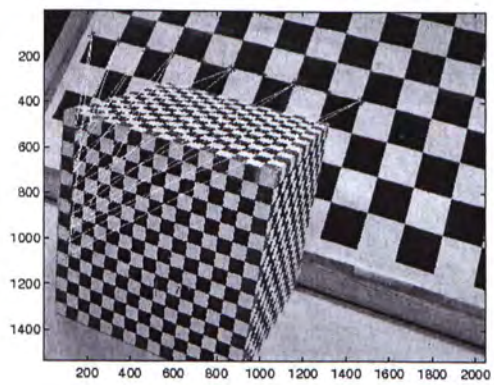



(c)

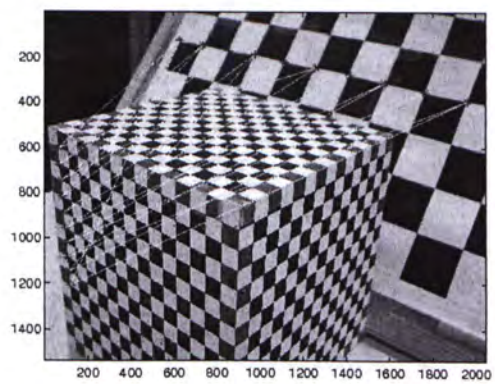



(d)

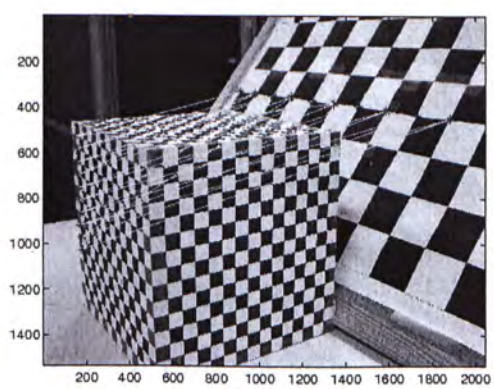
Figure 7.13: Line transfer from the second and third views of a linear ruled surface ($rank(\mathbf{A}) = 12$) to the first view by trifocal tensor that was determined respectively from (a) 12, (b) 11, (c) 10, and (d) 9 largest singular values of the estimation matrix.



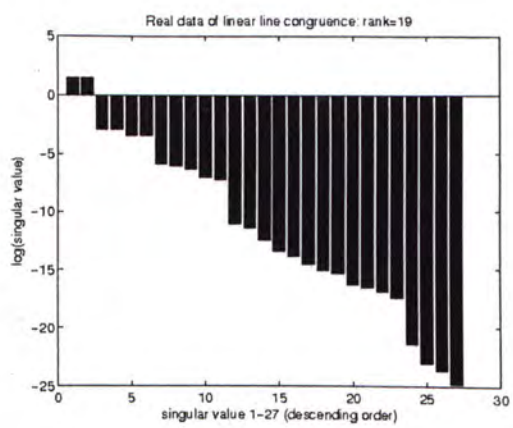

(a)




(b)




(c)




(d)

Figure 7.14: Three views of 20 lines of linear line congruence: $rank(\mathbf{A})$ should be 19: (a),(b),(c): the three views, (d) the singular values of the tensor estimation matrix.

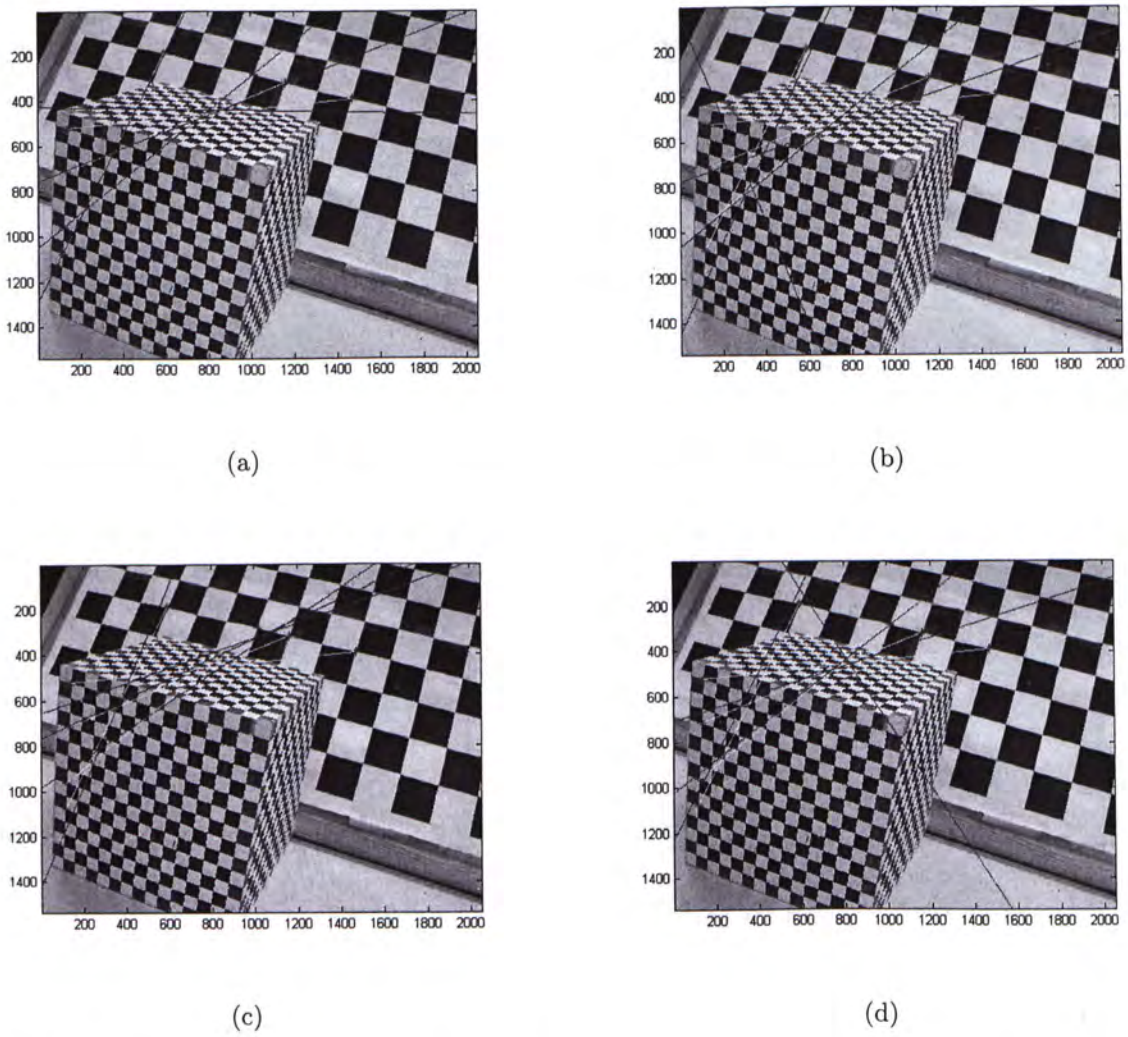
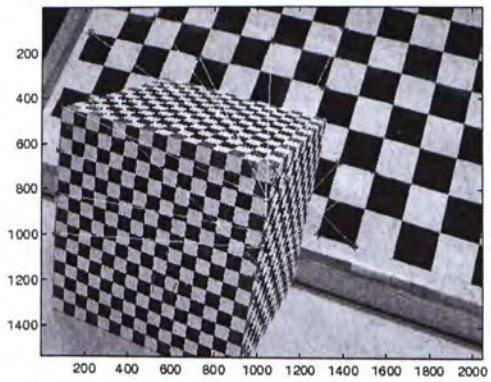
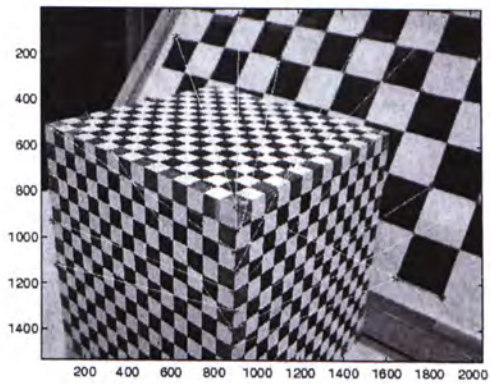


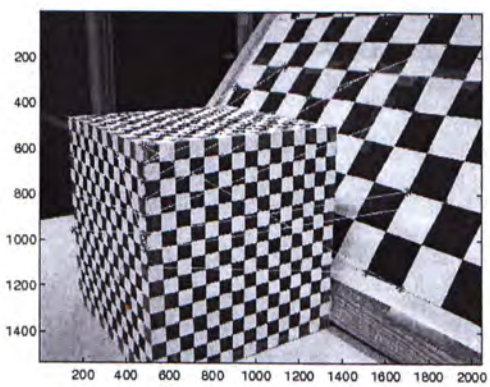

Figure 7.15: Line transfer from the second and third views of a linear line congruence ($rank(\mathbf{A}) = 19$) to the first view by trifocal tensor that was determined respectively from (a) 19, (b) 18, (c) 17, and (d) 16 largest singular values of the estimation matrix.



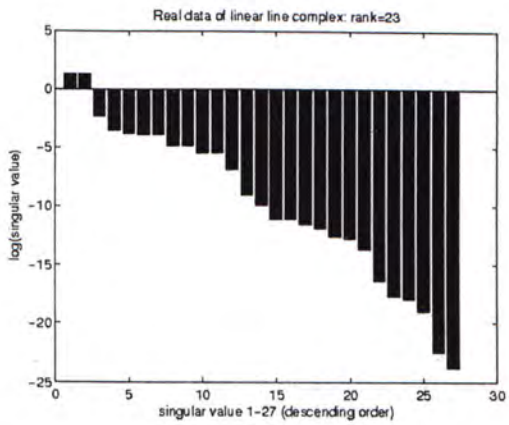

(a)




(b)




(c)




(d)

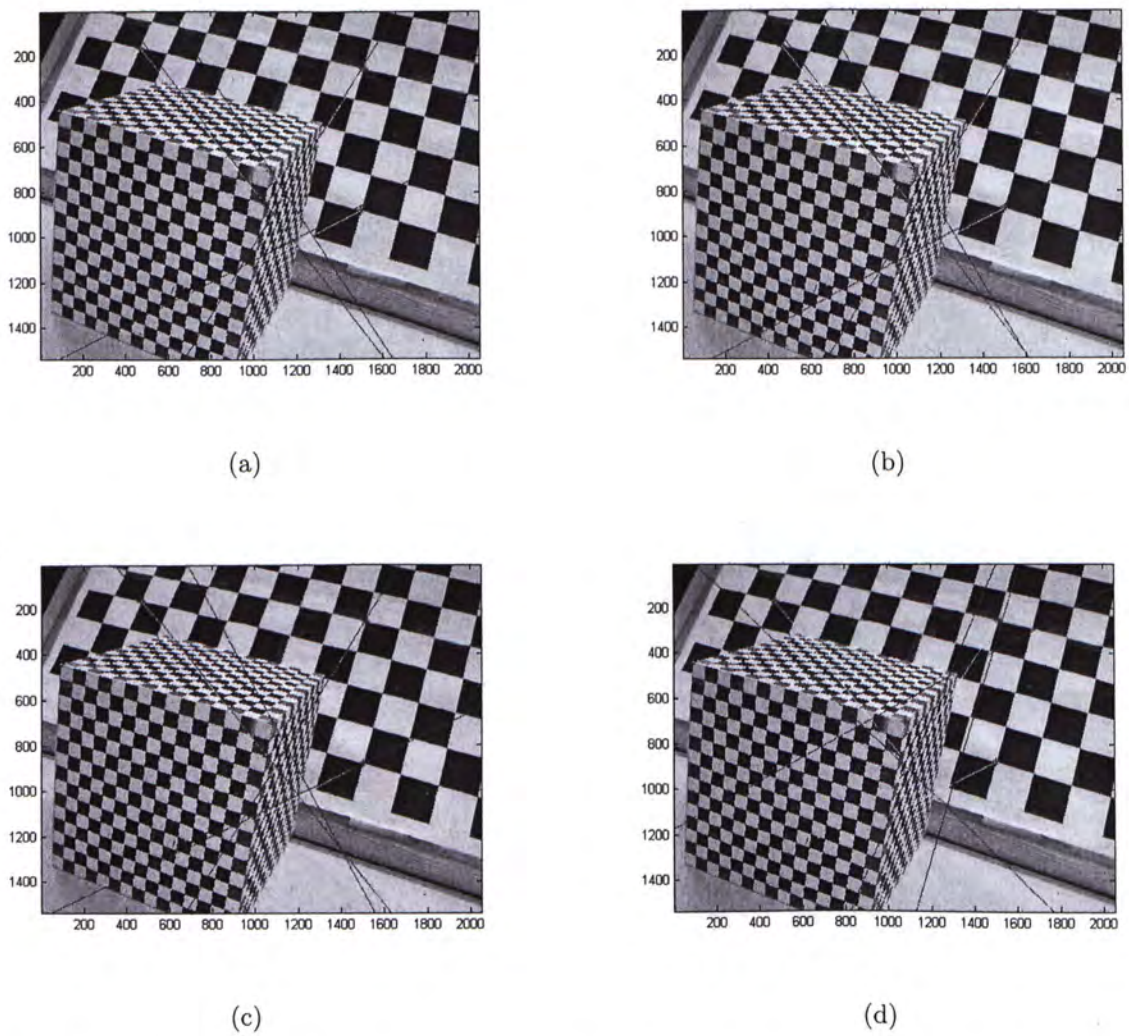
Figure 7.16: Three views of 15 lines of linear line complex: $rank(\mathbf{A})$ should be 23: (a),(b),(c): the three views, (d) the singular values of the tensor estimation matrix.

**Nonlinear line congruence.** Rank of estimation matrix is up to 26.

### 7.2.4 Linear line complex

**Linear line complex.** As shown in Fig. 7.17, line transfers using tensor approximated from 23, 22 and 21 largest singular values of the estimation matrix $\mathbf{A}$ were small, indicating that the rank of $\mathbf{A}$ was less than 23. This was in line with the prediction.

**Nonlinear line complex.** Rank of estimation matrix is up to 26.




(a) (b)

(c) (d)

Figure 7.17: Line transfer from the second and third views of a linear line complex ($rank(\mathbf{A}) = 23$) to the first view by trifocal tensor that was determined respectively from (a) 23, (b) 22, (c) 21, and (d) 20 largest singular values of the estimation matrix.

## 7.3 Empirical observation: ruled plane for line transfer

In this section, we observe that trifocal tensor can still be partially recovered from degenerate line structures. The line transfer result is correct for lines that belong to the degenerate line structures that is used for tensor estimation, but not correct for lines that do not belong to this set.

Consider image line correspondences from a ruled plane. Although the correspondences should be under-determining the trifocal tensor (since the rank of the estimation matrix in the case of ruled plane is no more than 15), owing to biasing influence of image noise and resolution error to the data, any linear estimation process would most definitely reach a particular solution for the trifocal tensor. Such a solution is only a particular one among the multiple trifocal tensor solutions constrained by the image line correspondences. If one is ignorant of the nature of the input data and blindly uses such a result for the initialization of any nonlinear algorithm in the literature for trifocal tensor determination, the quality of the final solution would be compromised.

We have experimented to examine how this particular solution of the trifocal tensor performed in transferring the positions of image lines from two views to the third view using the following equation:

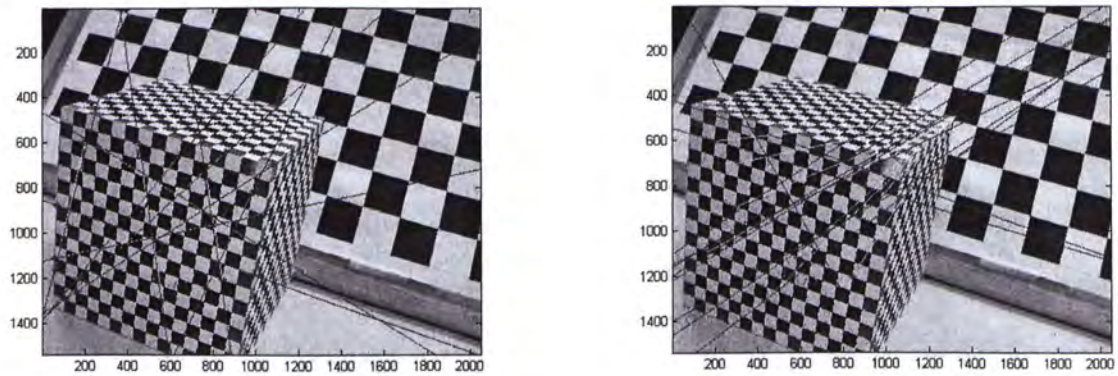
$$l_i = l'_j l''_k \mathcal{T}_i^{jk} \tag{7.1}$$

where $\mathbf{l}, \mathbf{l}', \mathbf{l}''$ are image lines in the three views respectively and $\mathcal{T}$ is the trifocal tensor.

For illustration, the three views of the scene shown in Fig. 7.11 were used, and the frontal face of the box in the scene was chosen as the ruled plane for determining the trifocal tensor. Extensive experiments show that while image lines coming from that particular "ruled plane" could be transferred correctly, those coming from beyond that plane generally could not, as displayed by Fig. 7.18.

**NOTE:** *The above properties hold not only for ruled plane but for all linear line structures described in this thesis.*

The fact that the trifocal tensor determined by image data of under-determining line structure could still successfully transfer image lines that are on the line structure itself




(a) (b)

Figure 7.18: Line transfer from the second and third views (shown in previous figure) to the first view by trifocal tensor that was determined from a ruled plane (the frontal surface of the box) ($rank(\mathbf{A}) = 15$): (a) line transfer to the first view for lines on the above ruled plane; (b) line transfer for lines beyond the above ruled plane.

(but generally not lines that are beyond the line structure) is actually exploited in our real image experimentation; it is used in the line transfer mechanism described in Table 7.4 for confirming the rank property of the various linear line structures.

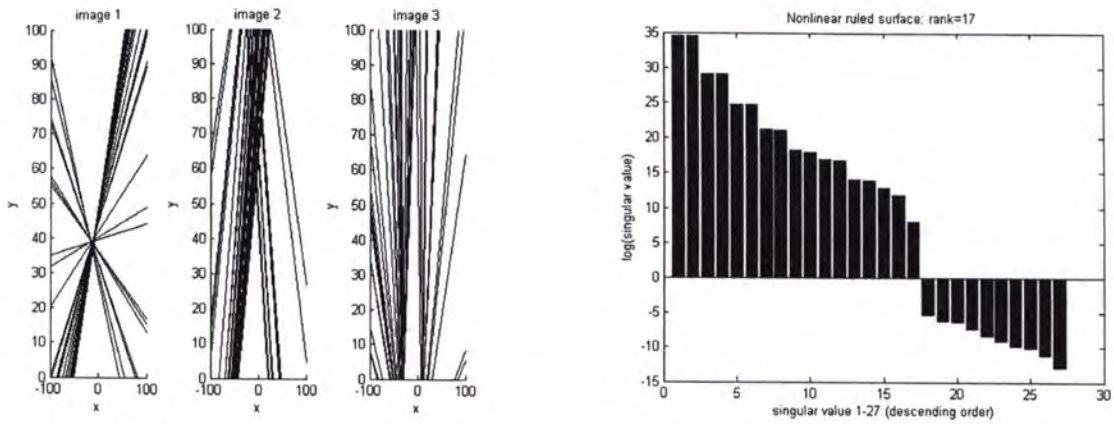
## 7.4 Simulation for non-linear line structures

As from Chapter 5, we use massive simulations to find out rank classifications of non-linear line structure.

**Nonlinear ruled surface.** Nonlinear ruled surface of order 2 was generated in 3-D, and three views of 50 lines of it were created and they are shown in Fig. 7.19(a). Fig. 7.19(b) shows the logarithmic values of all singular values. Visually it is obvious that the rank of the estimation matrix was 17.

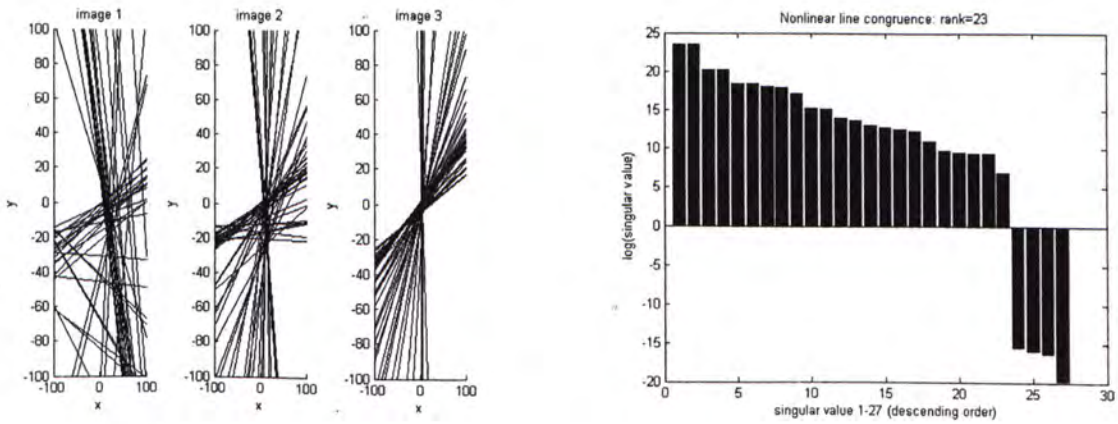
**Nonlinear line congruence.** Nonlinear line congruence of order 2 was generated in 3-D, and three views of 50 lines of it were created and they are shown in Fig. 7.20(a). Fig. 7.20(b) shows the logarithmic values of all singular values. Visually it is obvious that the rank of the estimation matrix was 23.

**Nonlinear line complex.** Nonlinear line complex of order 2 was generated in 3-D, and three views of 50 lines of it were created and they are shown in Fig. 7.21(a).



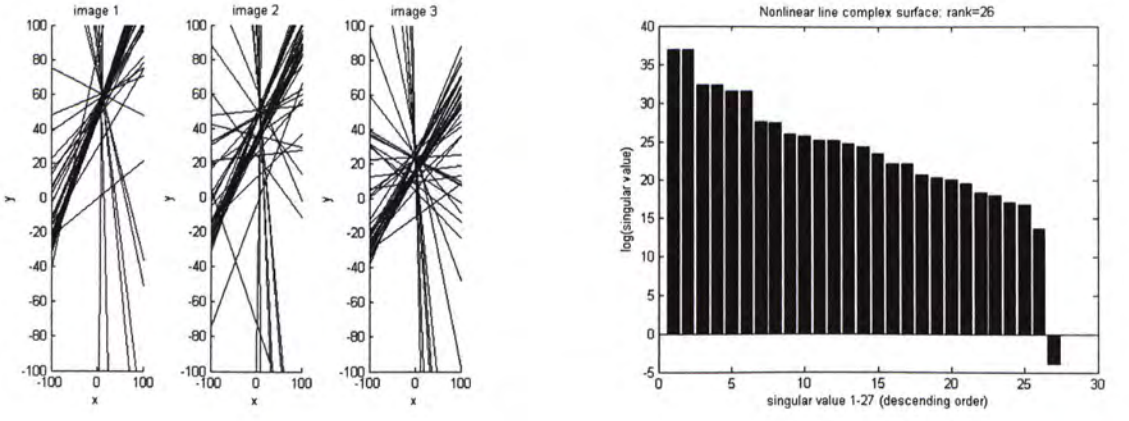

(a) (b)

Figure 7.19: Three views of 50 lines that belong to a 2nd order ruled surface: $rank(\mathbf{A}) = 17$; (a) the three views; (b) magnitudes of all singular values of the tensor estimation matrix.




(a) (b)

Figure 7.20: Three views of 50 lines that belong to a 2nd order line congruence: $rank(\mathbf{A}) = 23$; (a) the three views; (b) magnitudes of all singular values of the tensor estimation matrix.




(a) (b)

Figure 7.21: Three views of 50 lines that belong to a 2nd order line complex: $rank(\mathbf{A}) = 26$; (a) the three views; (b) magnitudes of all singular values of the tensor estimation matrix.

Fig. 7.21(b) shows the logarithmic values of all singular values. Visually it is obvious that the rank of the estimation matrix was 26.

# Chapter 8

# Conclusions and Future Work

This chapter concludes the thesis. We first summarize the content of the thesis, including rank classification of trifocal tensor estimation and recursive camera motion estimation framework. Then, we propose possible extensions to this work.

## 8.1 Summary

**Degeneracy of trifocal tensor estimation and empirical studies.**

The topic of the thesis is rank classification of trifocal tensor estimation for linear line structures, which describes how far is it from determining a unique trifocal tensor for different linear line structures. Trifocal tensor is an important quantity in multiple view geometry because it can be determined uniquely and linearly from image observables, and it captures all geometric constraints for among three-view. Thus, the topic of study in this thesis should be well-investigated before the application of trifocal tensor using line correspondences.

We have supplied a classification of the geometry of a collection of lines in space in regard to the determination of trifocal tensor from three views of the lines. In particular, we have presented what rank the estimation matrix would have if the line set is a sub-class (linear ruled surface, linear line congruence, or linear line complex) of linear line structure, or more specifically a sub-structure (point-pencil, point-star, or ruled plane) of linear line space. Trifocal tensor determination is crucial to motion estimation and projective reconstruction, and ambiguity would be present if a less than general set of

lines are unfortunately all that can be observed. Notice that not only is the presence of ambiguity pointed out, the extent of the ambiguity is also pinpointed in this work. Such a study over a list of major line structures is unprecedented.

The whole concept can also be looked at from the opposite angle. By examining the rank property of the tensor estimation matrix, one can have a glimpse of what possible line structure the cameras are indeed observing even if it is an under-ranked case. And this can be done without even attempting motion estimation, 3-D structure recovery, or trifocal tensor determination.

**Preliminary camera motion estimation framework.**

A geometric meaningful line tracking algorithm has been proposed in this research by merging geometry and kalman filter. It also illustrates recursive camera motion estimation framework based on tracking lines across views integrating kalman filter and trifocal tensor. It is only the preliminary investigation beyond the core work-rank classification of tensor estimation matrix, to show the possible applications. In the author's view, three-view geometry is the heart of motion analysis accompanied with the latest development on computing tools. First, it makes consistent of the geometric parameters across three views, which is unfortunately not directly satisfied for fundamental matrix across two views; Second, the time complexity for algorithms based on three-view is allowable, in contrast to four-view or more, where it is not only hard to analyze but also time-consuming.

Briefly, our research work has the following contribution - rank classification of trifocal tensor estimation using line correspondences with proof of algebraic lemmas and theorems in two papers:

1. Ming Zhao and Ronald Chung. Rank classification of linear line structure in determining trifocal tensor. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2008. to be published.

2. Ming Zhao and Ronald Chung. Trifocal tensor under linear line structures and its use in line transfer. In *Proceedings of International Conference on Pattern Recognition (ICPR)*, 2008. to be published.

## 8.2 Future work

**Degenerate line structures.**

Rank property analysis is meant for linear structures. A possible future work is to explore how the equivalence of rank property can be extracted for nonlinear line structures, and even different camera motions. Slight deviation of the observed lines from the line structures mentioned in this work, like having a set of lines that are supposedly from a ruled plane but are not exactly all co-planar, would affect the singular values of the tensor estimation matrix. Another possible future work would be to investigate how tolerant is the rank property toward such deviations.

There are to additional issues to investigate. 1. **Completeness.** It aims to prove that whether aforementioned rank classification can represent all line structures in the sense of trifocal tensor estimation. 2. **Distortion.** It involves two problems: the first is to study how lens distortion and image errors would affect the rank properties of line structures; the second is to study how minor changes in line structure in 3-D would affect the rank properties and therefore the accuracy of trifocal tensor estimation.

**Camera motion estimation.**

**Implementation.** There is a need to implement the whole camera motion estimation framework. **Calibrated images.** Just as essential matrix and fundamental matrix are in Euclidean space and projective space respectively, trifocal tensor in projective space, trifocal tensor can also have a similar quantity in Euclidean space. Therefore, there is a room to investigate the way to determine camera motion using this "essential trifocal tensor".

# Appendix A

# Notations

In this chapter the notations used in this thesis are summarized.

Table A.1: Notations.

| Notion | Interpretation |
|---|---|
| $\mathbf{A}$ | Tensor estimation matrix |
| $\mathbf{B}$ | Sub-matrix of tensor estimation matrix |
| $\mathbf{T}$ | Tensor-equivalent column vector |
| $\mathbf{X}$ | 3-D point |
| $\mathbf{\Omega}$ | 3-D plane |
| $\mathbf{H}, \mathbf{P}_{proj}$ | Projective transformation matrix |
| $\mathbf{P}_{aff}$ | Affine transformation matrix |
| $\mathbf{P}_{sim}$ | Similarity transformation matrix |
| $\mathbf{P}_{Euclidean}$ | Euclidean transformation matrix |
| $\mathbf{R}$ | Rotation matrix, or 27 tensor-equivalent vector |
| $\mathbf{t}$ | Translation matrix |
| $\mathbf{P}$ | Camera projection matrix |
| $\mathbf{K}$ | Intrinsic parameters matrix |
| $\mathbf{M}$ | Extrinsic parameters matrix |
| $f$ | Focal length |
| $p_x, p_y$ | coordinates of the principle point on image plane |
| $\alpha, \beta, \gamma$ | Euler angles |
| $\mathbf{C}$ | Camera center in camera coordinate frame |

Table A.2: Notations (cont.).

| | |
|---|---|
| $\tilde{\mathbf{C}}$ | Camera center in the world coordinate frame |
| $\tilde{\mathbf{P}}$ | Camera line projection matrix |
| $\mathbf{e}, \mathbf{e}'$ | Epipoles |
| $\mathbf{F}$ | Fundamental matrix |
| $\mathbf{H}_\pi$ | Homography matrix |
| $\mathbf{P}^+$ | Pseudo-inverse of $\mathbf{P}$ |
| $\mathbf{T}_i$ | Trifocal tensor in matrix form |
| $\mathbf{E}_\delta, \mathbf{W}_\delta$ | Homography matrices |
| $\mathbf{G}_\delta$ | Correlation matrix |
| $\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3$ | Standard homography slices |
| $\mathbf{G}_1, \mathbf{G}_2, \mathbf{G}_3$ | Standard correlation slices |
| $\mathcal{T}_i^{jk}$ | Trifocal tensor with indices i,j,k |
| $\mathbf{L}$ | Plücker coordinate of lines |
| $\mathbf{l}, \mathbf{l}', \mathbf{l}''$ | Image lines in three-view |
| $[\mathbf{l}]_\times$ | Skew-symmetric matrices related to $\mathbf{l}$ |
| $\bar{\mathbf{L}}$ | Plücker matrix |
| $\mathbf{L}^*$ | Dual Plücker representation |
| $\mathbf{K}$ | Any $9 \times N$ matrix |
| $[*]$ | A vector with * as its column entry |
| $\mathbf{A}_{*,i}$ | The $i$th column of $\mathbf{A}$ |
| $\mathbf{X}_k$ | State vector in Kalman filter |
| $\mathbf{q}_{k-1}$ | Process noise |
| $\mathbf{r}_k$ | Measurement noise |
| $\mathbf{A}$ | State transition matrix |
| $\mathbf{H}$ | Measurement matrix |
| $\mathbf{Q}$ | Covariance matrix for process noise |
| $\mathbf{R}$ | Covariance matrix for measurement noise |

# Appendix B

# Tensor

Tensor notation is frequently used in this thesis. For more details, the reader can refer to mathematics textbook about tensor or [62].

For vectors, indices are written as subscripts and coordinates are written as upper index, such as $x^i\mathbf{e}_i$, $\mathbf{x} = (x^1, x^2, x^3)^T$. In standard tensor notation, *covariant* indices are written as subscripts, and *contravariant* indices are written as superscripts. The sum over an index, such as $\mathbf{H}^i_j x^j$ is referred as a *contraction* , which can be simplified to $x^i$.

The trifocal tensor $\mathcal{T}_i^{jk}$ has one covariant $i$ and two contravariant indices $j, k$. Points and lines are indexed as contravariant $x^i$ and covariant $l_i$ respectively.

According to tensor slices, the 27-element trifocal tensor can be represented in a 3×3×3 cube as shown in Fig. B.1. It visually illustrates how line transfer is applied.

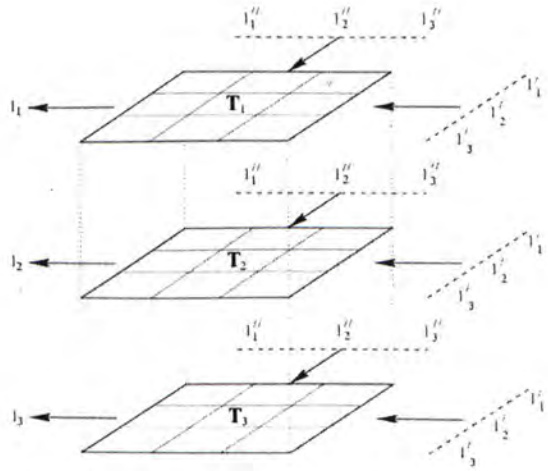



Figure B.1: Inside trifocal tensor.

# Appendix C

# Matrix Decomposition and Estimation Techniques

**Least-squares Minimization.** The problem can be formulated using simultaneous linear equations, $\mathbf{Y} = \beta\mathbf{X} + \varepsilon$. In practice, we assume that the dimension of column of $\mathbf{X}$ is much larger than the dimension of row of $\mathbf{X}$, and $\mathbf{X}$ is full column rank. Then $\beta$ can be estimated as $\beta = \mathbf{X}^{+}\mathbf{Y}$, where

$$\mathbf{X}^{+} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}' \tag{C.1}$$

is the pseudo-inverse of $\mathbf{X}$. If noise is normally distributed, then maximum likelihood estimation (MLE) is least squares estimator (LSE).

**Robust Estimation.** Two main robust estimators (M-Estimator, RANSAC) can be used to solving the sensitivity of LSE to outliers.

For M-Estimator, it minimizes $\sum \rho(r_i)$ instead of $\sum {r_i}^2$, where $\rho$ is an M-estimator. Huber estimator and Tukey estimator are respectively given by

$$\rho_{Hub}(x) = \begin{cases} x^2/2 & if|x| \leq c \\ c(|x| - c/2) & otherwise. \end{cases}$$

$$\rho_{Tuk}(x) = \begin{cases} \frac{c^2}{6}[1 - (1 - (\frac{x}{c})^2)^3)] & if|x| \leq c \\ c^2/6 & otherwise. \end{cases}$$

RANSAC can be referred to the classical text [63]. M-Estimator requires an initial estimate to converge; on the contrary, RANSAC does not require such initial estimate. M-Estimator utilizes data better (more accurate); RANSAC only utilizes partial data. Therefore, they can be combined as RANSAC for initial estimate and M-estimator for better results.

**Kalman filter.** Kalman Filtering [64] considers a linear dynamic model

$$\begin{aligned} \mathbf{x}_k &= \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{q}_{k-1} \\ \mathbf{y}_k &= \mathbf{H}_k\mathbf{x}_k + \mathbf{r}_k \end{aligned} \tag{C.2}$$

where

$$\begin{aligned} \mathbf{q}_{k-1} &\sim N(\mathbf{0}, \mathbf{Q}_{k-1}) \\ \mathbf{r}_k &\sim N(\mathbf{0}, \mathbf{R}_k) \end{aligned} \tag{C.3}$$

$\mathbf{x}_k$ is the state at time $t_k$, $\mathbf{y}_k$ is the direct measurement containing noise, $\mathbf{A}_{k-1}$ is the transition matrix, $\mathbf{q}_{k-1}$ denotes the uncertainty during the state transition. The $\mathbf{A}_{k-1}, \mathbf{H}_k, \mathbf{Q}_{k-1}, \mathbf{R}_k$ are known for all $k > 0$. The initial estimate is assumed to be known. Let $\mathbf{m}_k$, $\mathbf{P}_k$ denote the state mean and state covariance respectively, then we have the prediction result,

$$\begin{aligned} \mathbf{m}_k^- &= \mathbf{A}_{k-1}\mathbf{m}_{k-1} \\ \mathbf{P}_k^- &= \mathbf{A}_{k-1}\mathbf{P}_{k-1}\mathbf{A}_{k-1}^T + \mathbf{Q}_{k-1} \end{aligned} \tag{C.4}$$

Finally, the state mean and covariance can be updated by

$$\begin{aligned} \mathbf{K}_k &= \mathbf{P}_k^-\mathbf{H}_k\mathbf{m}_k^- \\ \mathbf{m}_k &= \mathbf{m}_k^- + \mathbf{K}_k\mathbf{v}_k \\ \mathbf{P}_k &= \mathbf{P}_k^- - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^T \end{aligned} \tag{C.5}$$

**Extended Kalman Filter (EKF).** EKF [64] considers a nonlinear dynamic model

governed by the stochastic difference equation

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}) \tag{C.6}$$

with a measurement

$$\mathbf{z}_k = h(\mathbf{x}_k, \mathbf{v}_k) \tag{C.7}$$

where $\mathbf{w}_k$, $\mathbf{v}_k$ represent the process and measurement noise, which are not known at each time step. Therefore, the state and measurement can be approximated as

$$\begin{aligned} \tilde{\mathbf{x}}_k &= f(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}, \mathbf{0}) \\ \tilde{\mathbf{z}}_k &= h(\tilde{\mathbf{x}}_k, \mathbf{0}) \end{aligned} \tag{C.8}$$

By linearization, the state vector can be approximated according to the optimality of Bayes' rule. EKF time update ("Predict") equations are given by

$$\begin{aligned} \hat{\mathbf{x}}_k^- &= f(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}, \mathbf{0}) \\ \mathbf{P}_k^- &= \mathbf{A}_k \mathbf{P}_{k-1} \mathbf{A}_k^T + \mathbf{W}_k \mathbf{Q}_{k-1} \mathbf{W}_k^T \end{aligned} \tag{C.9}$$

and the measurement update ("Correct") equations are

$$\begin{aligned} \mathbf{K}_k &= \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{V}_k \mathbf{R}_k \mathbf{V}_k^T)^{-1} \\ \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - h(\hat{\mathbf{x}}_k^-, 0)) \\ \mathbf{P}_k &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- \end{aligned} \tag{C.10}$$

# Appendix D

# MATLAB Files

Here encloses partial matlab code for trifocal tensor estimation, line transfer and simulation.

## D.1 Estimation matrix

```
% Tensor estimation matrix
A = zeros(2*nlines,27);
% equation 1
r = 1 : nlines;
A(r, 10) =( l1(3,:).*l2(1,:).*l3(1,:) )';
A(r, 11) =( l1(3,:).*l2(1,:).*l3(2,:) )';
A(r, 12) =( l1(3,:).*l2(1,:).*l3(3,:) )';
A(r, 13) =( l1(3,:).*l2(2,:).*l3(1,:) )';
A(r, 14) =( l1(3,:).*l2(2,:).*l3(2,:) )';
A(r, 15) =( l1(3,:).*l2(2,:).*l3(3,:) )';
A(r, 16) =( l1(3,:).*l2(3,:).*l3(1,:) )';
A(r, 17) =( l1(3,:).*l2(3,:).*l3(2,:) )';
A(r, 18) =( l1(3,:).*l2(3,:).*l3(3,:) )';

A(r, 19) =( -l1(2,:).*l2(1,:).*l3(1,:) )';
A(r, 20) =( -l1(2,:).*l2(1,:).*l3(2,:) )';
```

```
A(r, 21) =( -l1(2,:).*l2(1,:).*l3(3,:) )';
A(r, 22) =( -l1(2,:).*l2(2,:).*l3(1,:) )';
A(r, 23) =( -l1(2,:).*l2(2,:).*l3(2,:) )';
A(r, 24) =( -l1(2,:).*l2(2,:).*l3(3,:) )';
A(r, 25) =( -l1(2,:).*l2(3,:).*l3(1,:) )';
A(r, 26) =( -l1(2,:).*l2(3,:).*l3(2,:) )';
A(r, 27) =( -l1(2,:).*l2(3,:).*l3(3,:) )';

% equation 2
r = (nlines+1) : nlines*2;
A(r, 1) =( -l1(3,:).*l2(1,:).*l3(1,:) )';
A(r, 2) =( -l1(3,:).*l2(1,:).*l3(2,:) )';
A(r, 3) =( -l1(3,:).*l2(1,:).*l3(3,:) )';
A(r, 4) =( -l1(3,:).*l2(2,:).*l3(1,:) )';
A(r, 5) =( -l1(3,:).*l2(2,:).*l3(2,:) )';
A(r, 6) =( -l1(3,:).*l2(2,:).*l3(3,:) )';
A(r, 7) =( -l1(3,:).*l2(3,:).*l3(1,:) )';
A(r, 8) =( -l1(3,:).*l2(3,:).*l3(2,:) )';
A(r, 9) =( -l1(3,:).*l2(3,:).*l3(3,:) )';

A(r, 19) =( l1(1,:).*l2(1,:).*l3(1,:) )';
A(r, 20) =( l1(1,:).*l2(1,:).*l3(2,:) )';
A(r, 21) =( l1(1,:).*l2(1,:).*l3(3,:) )';
A(r, 22) =( l1(1,:).*l2(2,:).*l3(1,:) )';
A(r, 23) =( l1(1,:).*l2(2,:).*l3(2,:) )';
A(r, 24) =( l1(1,:).*l2(2,:).*l3(3,:) )';
A(r, 25) =( l1(1,:).*l2(3,:).*l3(1,:) )';
A(r, 26) =( l1(1,:).*l2(3,:).*l3(2,:) )';
A(r, 27) =( l1(1,:).*l2(3,:).*l3(3,:) )';

rankA=rank(A);
```

$[U, D, V] = svd(A, 0)$;

t0=V(:,27);

T0=permute( reshape(t0,3,3,3) , [2 1 3] );

## D.2 Line transfer

% Line transfer from corresponding lines in the second and third views

% to the first using trifocal tensor, refer to [7] p.367

% Arguments: $l1$, $l2$, $l3$: lines in view 1,2,3 respectively

function [l1]=line_transfer(l2,l3, T)

if size(l2) =size(l3) error('size of l2, l3 is different'); end

Size=size(l2); Size=Size(2);

for i=1:Size

l1(:,i)=[l2(:,i)'*T(:,:,1)*l3(:,i); l2(:,i)'*T(:,:,2)*l3(:,i); l2(:,i)'*T(:,:,3)*l3(:,i)];

l1(:,i)=l1(:,i)/l1(3,i);

end

return

## D.3 Simulation

% **Camera matrix.**

K=[5 0 0; 0 5 0; 0 0 1];

alpha=pi/29;

beta=pi/30;

gamma=pi/31;

R1=[1 0 0; 0 cos(alpha) -sin(alpha); 0 sin(alpha) cos(alpha)]* [cos(beta) 0 sin(beta); 0 1 0; -sin(beta) 0 cos(beta)]* [cos(gamma) -sin(gamma) 0; sin(gamma) cos(gamma) 0; 0 0 1];

```
t1=[-1.5;-0.5;-1];
P1=K*[R1 t1];

% Line projection matrix from camera matrix.
P1_lin=[det( P1(:,1:3) )*inv(P1(:,1:3)'), -vgg_contreps(P1(:,4))*P1(:,1:3)];
P2_lin=[det( P2(:,1:3) )*inv(P2(:,1:3)'), -vgg_contreps(P2(:,4))*P2(:,1:3)];
P3_lin=[det( P3(:,1:3) )*inv(P3(:,1:3)'), -vgg_contreps(P3(:,4))*P3(:,1:3)];

% LLC.
nlines=50;
L(:,6)=ones(nlines, 1);
L(:,1)=rand(nlines, 1);
L(:,4)=rand(nlines, 1)*100;
L(:,2)=rand(nlines, 1)*50;
L(:,5)=L(:,1)-4*L(:,2)+3*L(:,4)+5*L(:,6);
L(:,3)=-L(:,1).*L(:,4)-L(:,2).*L(:,5);

% Line projection.
l1_lin=P1_lin*L';
l1_lin(1,:)=l1_lin(1,:)./l1_lin(3,:);
l1_lin(2,:)=l1_lin(2,:)./l1_lin(3,:);
l1_lin(3,:)=l1_lin(3,:)./l1_lin(3,:);
```

# Bibliography


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

[2] A.E. Bartoli and P.F. Sturm. The 3d line motion matrix and alignment of line reconstructions. *Int. J. Comput. Vision*, 57(3):159–178, May 2004.

[3] Adrien Bartoli, Richard I. Hartley, and Fredrik Kahl. Motion from 3d line correspondences: Linear and non-linear solutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 01, page 477, Los Alamitos, CA, USA, 2003. IEEE Computer Society.

[4] Adrien Bartoli and Peter Sturm. Structure-from-motion using lines: representation, triangulation, and bundle adjustment. *Comput. Vis. Image Underst.*, 100(3):416–441, 2005.

[5] Stephen John Maybank. The critical line congruence for reconstruction from three images. *Applicable Algebra in Engineering, Communication and Computing*, 6:89–113, 1993.

[6] Motilal Agrawal and Kurt Konolige. Real-time localization in outdoor environments using stereo vision and inexpensive gps. In *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*, pages 1063–1068, Washington, DC, USA, 2006. IEEE Computer Society.

[7] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[8] N. Navab and O.D. Faugeras. Monocular pose determination from lines: Critical sets and maximum number of solutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 254–260, 1993.

[9] M.K. Leung, Yuncai Liu, and T.S. Huang. Estimating 3d vehicle motion in an outdoor scene from monocular and stereo image sequences. *IEEE Workshop on Visual Motion*, 91:62–68, 7-9 Oct 1991.

[10] Y. Liu. Rigid object motion estimation from intensity images using straight line correspondences. In *Ph.D. Thesis*, 1990.

[11] Ming Zhao and Ronald Chung. Rank classification of linear line structure in determining trifocal tensor. In *Proceedings of the European Conference on Computer Vision*, 2008. to be published.

[12] Ming Zhao and Ronald Chung. Trifocal tensor under linear line structures and its use in line transfer. In *Proceedings of International Conference on Pattern Recognition*, 2008. to be published.

[13] O. Faugeras and B. Mourrain. On the geometry and algebra of the point and line correspondences between n images. In *International Conference on Computer Vision*, page 951, Washington, DC, USA, 1995. IEEE Computer Society.

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

[15] Richard Hartley and Fredrik Kahl. Critical configurations for projective reconstruction from multiple views. *Int. J. Comput. Vision*, 71(1):5–47, 2007.

[16] O.D. Faugeras and S.J. Maybank. Motion from point matches: Multiplicity of solutions. 4(3):225–246, 1990.

[17] A. Shashua and S. Maybank. Degenerate n point configurations of three views: Do critical surfaces exist. Technical Report 96-19, Hebrew Univ. of Jerusalem, Nov. 1996.

[18] Robert J. Holt and Arun N. Netravali. Uniqueness of solutions to three perspective views of four points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(3):303–307, 1995.

[19] Long Quan. Invariants of 6 points from 3 uncalibrated images. In *ECCV '94: Proceedings of the Third European Conference-Volume II on Computer Vision*, pages 459–470, London, UK, 1994. Springer-Verlag.

[20] Stephen J. Maybank and Amnon Shashua. Ambiguity in reconstruction from images of six points. In *ICCV*, pages 703–708, 1998.

[21] Richard I. Hartley. Ambiguous configurations for 3-view projective reconstruction. In *Proceedings of the 6th European Conference on Computer Vision-Part I*, pages 922–935, London, UK, 2000. Springer-Verlag.

[22] Fredrik Kahl, Richard Hartley, and Kalle Astrom. Critical configurations for n-view projective reconstruction. In *Proceedings of the Computer Vision and Pattern Recognition*, volume 02, page 158, Los Alamitos, CA, USA, 2001. IEEE Computer Society.

[23] Richard Hartley and Fredrik Kahl. A critical configuration for reconstruction from rectilinear motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 01, page 511, Los Alamitos, CA, USA, 2003. IEEE Computer Society.

[24] Thomas Buchanan. Critical sets for 3d reconstruction using lines. In *ECCV '92: Proceedings of the Second European Conference on Computer Vision*, pages 730–738, London, UK, 1992. Springer-Verlag.

[25] Thomas Buchanan. On the critical set for photogrammetric reconstruction using line tokens in p3(c). *Geometriae Dedicata*, 44:223–232, 1992.

[26] Y. Liu and T.S. Huang. A linear algorithm for motion estimation using straight line correspondences. In *Proceedings of the International Conference on Pattern Recognition*, pages I: 213–219, 1988.

[27] N. Navab and O.D. Faugeras. The critical sets of lines for camera displacement estimation: A mixed euclidean-projective and constructive approach. *Int. J. Comput. Vision*, 23:17–44, 1997.

[28] Gideon P. Stein and Amnon Shashua. On degeneracy of linear reconstruction from three views: Linear line complex and applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1407:862–878, 1998.

[29] Carlo Tomasi and Takeo Kanade. Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University, April 1991.

[30] J. Weng, Y. Liu, T.S. Huang, and N. Ahuja. Estimating motion/structure from line correspondences: A robust linear algorithm and uniqueness theorems. *CVPR*, 88:387–392, 1988.

[31] Cordelia Schmid and Andrew Zisserman. Automatic line matching across views. In *International Conference on Computer Vision & Pattern Recognition*, pages 666–671, 1997.

[32] Vincent Lepetit and Pascal Fua. Monocular model-based 3d tracking of rigid objects. *Found. Trends. Comput. Graph. Vis.*, 1(1):1–89, 2005.

[33] Tom Drummond and Roberto Cipolla. Real-time visual tracking of complex structures. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7):932–946, 2002.

[34] David G. Lowe. Robust model-based motion tracking through the integration of search and estimation. *Int. J. Comput. Vision*, 8(2):113–122, 1992.

[35] E. Marchand, P. Bouthemy, and F. Chaumette. A 2d-3d model-based approach to real-time visual tracking. 19(13):941–955, November 2001.

[36] A. Fitzgibbon and A. Zisserman. Automatic camera tracking. In M. Shah and R. Kumar, editors, *Video Registration*, pages 18–35. Kluwer, 2003.

[37] J.C. Clarke, S. Carlsson, and A. Zisserman. Detecting and tracking linear features efficiently. page Tracking, 1996.

[38] Edward Rosten and Tom Drummond. Fusing points and lines for high performance tracking. In *IEEE International Conference on Computer Vision*, volume 2, pages 1508–1511, October 2005.

[39] D. Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, volume 20, pages 91–110, 2003.

[40] C. Schmid and A. Zisserman. The geometry and matching of lines and curves over multiple views. 40(3):199–233, December 2000.

[41] J.P. Gambotto. Tracking points and line segments in image sequences. pages 38–46, 1989.

[42] J. B. Park Y. Yoon, A. Kosaka and A. C. Kak. A new approach to the use of edge extremities for model-based object tracking. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1883–1889, April 2005.

[43] Xinquan Shen and Phil Palmer. Uncertainty propagation and the matching of junctions as feature groupings. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(12):1381–1395, 2000.

[44] Maria Petrou and Josef Kittler. Optimal edge detectors for ramp edges. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(5):483–491, 1991.

[45] M.E. Spetsakis and J. Aloimonos. Structure and motion using line correspondences. *Int'l J. Computer Vision*, 4(3):171–184, 1990.

[46] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: a factorization method. *Int. J. Comput. Vision*, 9(2):137–154, 1992.

[47] Conrad J. Poelman and Takeo Kanade. A paraperspective factorization method for shape and motion recovery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3):206–218, 1997.

[48] Michael Ming Yuen Chang and Kin Hong Wong. Model reconstruction and pose acquisition using extended lowe's method. *IEEE Transactions on Multimedia*, 7, April 2005.

[49] A.W. Fitzgibbon and A. Zisserman. Automatic camera tracking. page Chapter 2, 2003.

[50] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *ICCV '99: Proceedings of the International Workshop on Vision Algorithms*, pages 298–372, London, UK, 2000. Springer-Verlag.

[51] Y.K. Yu, K.H. Wong, M.M.Y. Chang, and S.H. Or. Recursive camera-motion estimation with the trifocal tensor. 36(5):1081–1090, October 2006.

[52] T. J. Broida, S. Chandrashekhar, and R. Chellappa. Recursive 3-D motion estimation from a monocular image sequence. *IEEE Transactions on Aerospace and Electronic Systems*, 26(4):639–656, 1990.

[53] F. Lustaman O.D. Faugeras and G. Toscani. Motion and structure from point and line matches. In *Proc. Int'l Conf. Computer Vision*, pages 25–33, June 1987.

[54] James L. Crowley and Patrick Stelmaszyk. Measurement and integration of 3-d structures by tracking edge lines. In *European Conference on Computer Vision*, pages 269–280, 1990.

[55] J.L. Jezouin and N.J. Ayache. 3d structure from a monocular sequence of images. pages 441–444, 1990.

[56] Gideon P. Stein and Amnon Shashua. On degeneracy of linear reconstruction from three views: Linear line complex and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(3):244–251, 1999.

[57] E. J. Wilczynski. *Projective differential geometry of curves and ruled surfaces.* New York : Chelsea Pub. Co., 1905.

[58] Johannes Wallner Helmut Pottmann. *Computational line geometry.* Mathematics and visualization. Berlin ; New York : Springer, 2001.

[59] J.G. Semple and G.T. Kneebone. *Algebraic projective geometry.* Oxford : Clarendon Press ; New York : Oxford University Press, 1952.

[60] R. Nevatia and K. R. Babu. Linear feature extraction and description. In *Computer Graphics and Image Processing*, volume 13, pages 257–269, July 1980.

[61] Olivier Faugeras and Luc Robert. What can two images tell us about a third one? *Int. J. Comput. Vision*, 18(1):5–19, 1996.

[62] C. Ressl. *Geometry, Constraints and Computation of the Trifocal Tensor.* PhD thesis, Vienna University of Technology, June 2003.

[63] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[64] G. Welch and G. Bishop. An introduction to the kalman filter. Technical report, University of North Carolina at Chapel Hill, 1995.

# Index