TOWARDS A SELF-CALIBRATING VIDEO CAMERA NETWORK FOR CONTENT ANALYSIS AND FORENSICS

by

IMRAN NAZIR JUNEJO
B.S. University of Arizona, Dec. 2001
M.S. University of Central Florida, Dec. 2005

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the School of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2007

Major Professor: Hassan Foroosh

ABSTRACT

Due to growing security concerns, video surveillance and monitoring has received an immense attention from both federal agencies and private firms. The main concern is that a single camera, even if allowed to rotate or translate, is not sufficient to cover a large area for video surveillance. A more general solution with wide range of applications is to allow the deployed cameras to have a non-overlapping field of view (FoV) and to, if possible, allow these cameras to move freely in 3D space. This thesis addresses the issue of how cameras in such a network can be calibrated and how the network as a whole can be calibrated, such that each camera as a unit in the network is aware of its orientation with respect to all the other cameras in the network.

Different types of cameras might be present in a multiple camera network and novel techniques are presented for efficient calibration of these cameras. Specifically: (i) For a stationary camera, we derive new constraints on the Image of the Absolute Conic (IAC). These new constraints are shown to be intrinsic to IAC; (ii) For a scene where object shadows are cast on a ground plane, we track the shadows on the ground plane cast by at least two unknown stationary points, and utilize the tracked shadow positions to compute the horizon line and hence compute the camera intrinsic and extrinsic parameters; (iii) A novel solution to a scenario where a camera is observing pedestrians is presented. The uniqueness of formulation lies in recognizing two harmonic homologies present in the geometry obtained by observing pedestrians; (iv) For a freely moving camera, a novel practical method is proposed for its self-calibration which even allows it to change its internal parameters by zooming; and (v) due to the increased application of the pan-tilt-zoom (PTZ) cameras, a technique is presented that uses only two images to estimate five camera parameters.

For an automatically configurable multi-camera network, having non-overlapping field of view and possibly containing moving cameras, a practical framework is proposed that determines the geometry of such a dynamic camera network. It is shown that only one automatically computed vanishing point and a line lying on any plane orthogonal to the vertical direction is sufficient to infer the geometry of a dynamic network. Our method generalizes previous work which considers restricted camera motions. Using minimal assumptions, we are able to successfully demonstrate promising results on synthetic as well as on real data. Applications to path modeling, GPS coordinate estimation, and configuring mixed-reality environment are explored.

To my parents and my wonderful wife

with all my love.

ACKNOWLEDGEMENTS

After thanking God Almighty, I would like to thank my father *Nazir* and my mother *Shamshad* for their love, support and encouragement. I would like to thank my wonderful wife *Naheed* for her patience and support. In my memory are always my brothers *Asif* and *Khurram*, and my sisters *Shabana*, *Munazza*, and *Anum*. I would like to acknowledge my dearest uncles *Dr. Deedar* and *Dr. Tameez* for instilling the love of learning and achievement.

I would like to thank my thesis advisor *Dr. Hassan Foroosh* for his valuable guidance and encouragement, and for having incredible patience with me. I have learned more than just computer vision from him.

I would like to thank *Dr. Charles Hughes, Dr. Marshall Tappen*, and *Dr. Olusegun Illegbusi* for serving as my committee members. I acknowledge their valuable comments and suggestions.

I thank the amazing people of the Computational Imaging Lab and my coauthors *Nazim Ashraf, Xiaochun Cao, Omar Javed, Alexei Gritai, Yaser Sheikh,* and *Yuping Shen*. I would like to also thank *Dr. Alper Yilmaz* for his valuable help and support.

I would also like to thank National Science Foundation (NSF) for supporting my research at UCF.

Last but not the least, my friends who have made my stay at Orlando an unforgettable experience of my life. I have to be careful to name them all: *Muzzamil, Muazzam, Arsalan, Adeel, Shahab, Zeeshan, Mumtaz, Muhammad Rami, Zain, Ashar, Jamal, Mais, Alex, Chris Millward, Majid, Sajjad, Saad, Wisam, Shaadi, Murat, Feras, Sohail, Farzan,* and *Ashraf Riad*.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

In recent years, there has been a growing interest in both federal agencies and private firms to employ video cameras for monitoring and surveillance. These employed video cameras can have an overlapping or non-overlapping field of view (FoV). It is the aim of this thesis to allow these networked video cameras to self-configure. That is, each camera should automatically determine its relative orientation with respect to every other camera in the network.

Most of the deployed camera systems share one common feature; a human operator must monitor them. The effectiveness and the responsiveness of such a network is determined not by the technological capabilities, but by the vigilance of the person monitoring these cameras. Moreover, employing many people to monitor video cameras can be quite expensive. Therefore, due to increased interest in the field of video surveillance, automatic object detection and tracking is one of the primary areas of research in the field of computer vision. Using automatic object detection and tracking not only minimizes the cost of employing many humans to monitor surveillance cameras (or surveillance videos), but also maximizes the chances of successful event detection. Some example of such systems include [JRA03, KCM03, ZAK05, BA03].

However, most of the existing methods for configuring camera network employ *stationary*, or *overlapping FoV* cameras; or cameras whose *intrinsic* and *extrinsic* parameters are assumed to be fixed or known (for e.g. [CT99, TDG05, HHZ06, SSK05]). For example, Kang et al. [KCM03] use an affine transformation between each consecutive pair of images to stabilize moving camera sequences. A planar homography computed by point correspondences is used to register stationary and moving cameras. Zhao et al. [ZAK05] formulate tracking in a unified mixture model

framework. Ground-based space-time cues are used to match trajectories of objects moving from one camera to another. Javed et al. [JRS03] track objects across multiple stationary cameras by exploiting redundancy in paths that objects tend to follow. The system learns the camera topology and path probabilities of objects using Parzen windows in the training phase. The correct correspondences in the testing phase are assigned using the maximum a posteriori (MAP) estimation framework.

If the camera is moved for some reason, or the lighting conditions are changed (due to a cloudy weather), the methods mentioned above generally depict undesirable behavior. And they need to recalculated the probabilities associated with object behavior (i.e. motion characteristics). Similarly, if the camera intrinsic paramors are changes, for example a change of zoom, the methods fail to cater for this changed condition (e.g. [KCM03]).

Our goal in this thesis is to overcome the above restrictions, and when possible employ non-overlapping FoV cameras that are able to move freely in the environment. The main motivation for deploying networked cameras is that a single camera, even if allowed to rotate or translate, is not sufficient to cover a large area. By employing multiple cameras with non-overlapping or disjoint FoV, we would like to maximize the monitoring area in addition to inferring the network configuration. By network configuration we mean the location and orientation of cameras in the network with respect to each other, also known as the network geometry. A more general case with a wide range of applications is when the deployed disjoint FoV cameras may be allowed to move freely in 3D space, e.g. on roaming security vehicles. This configuration induces a *dynamic network geometry*. We propose a framework for self-calibration of such a dynamic network, thereby obtaining the dynamic geometry of the network along with self-calibrating each camera in the network.

1.1 Motivation

By configuring a camera network, where the cameras are able to freely move in space and the camera FoV is non-overlapping (or disjoint), we can perform tasks which might not be possible on existing systems that use stationary cameras or cameras with fixed parameters. Some motivating factors for configuring such a camera network can be to:

- direct cameras to follow a particular object [DDZ01],
- calibrate cameras so that the observations are more coordinated and perform measurements (with known scale) and possibly construct a Euclidean model of the 3-D world model [MK04, CT04],
- solve the camera hand-over problem i.e. establish correspondence between tracked objects in different cameras
- generate image/video scene mosaic
- infer network topology [ME05],
- build terrain model [CT98] or do spatial learning for navigation [YB96, Tan96], and
- estimate relative orientation and location between cameras in the network.

An overview of the key components of the system are shown in Figure 1.1. Each of these components has a long history in computer vision. The components which are addressed in this thesis are camera calibration and relative camera orientation (i.e. network configuration) (rectangles with red outline cf. Figure 1.1).

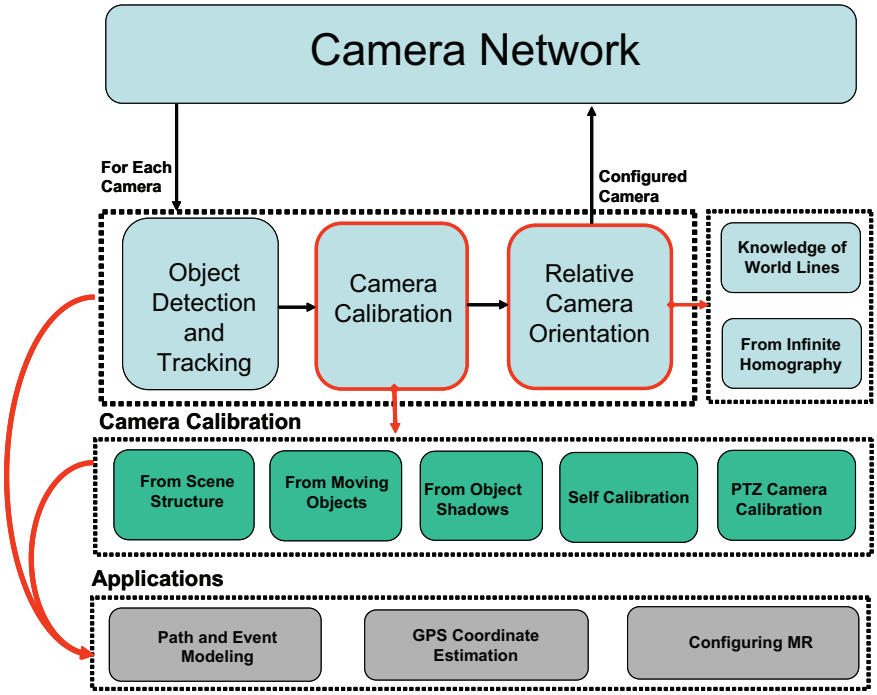


Figure 1.1: Overview of the key components of a multi-camera network.

1.2 Contributions

This thesis improves on the state of the art on various aspects of computer vision.

Camera Calibration: We present camera calibration techniques for different real world scenarios. We propose five different calibration techniques, based on the characteristics of the scene:

I. We revisit the role of image of the absolute conic (IAC) in determining the camera geometry, and propose new constraints that are intrinsic to it, reflecting its invariant features. We investigate the application of these new constraints on camera calibration.

II. We focus on the scenes where there is a reference plane and some shadows are cast on it. In such scenes, we track the shadows on the reference plane (e.g. the ground plane)

cast by at least two unknown stationary points, and utilize the tracked shadow positions to compute the horizon line and hence compute the camera intrinsic and extrinsic parameters.

III. We propose a robust and a general linear solution to the problem of camera calibration by observing moving objects by adopting a formulation different from the existing methods. The uniqueness of formulation lies in recognizing two harmonic homologies present in the geometry associated with walking pedestrians, and then using properties of these homologies to obtain linear constraints on the unknown camera parameters.

IV. We present a novel practical method for self-calibrating a camera which may move freely in space while changing its internal parameters by zooming. We show that point correspondences between a pair of images, and the fundamental matrix computed from these point correspondences, are sufficient to recover the internal parameters of a camera. No calibration object with known 3-D shape is required and no limitations are imposed on the unknown camera motion, as long as the camera is projective.

V. A novel solution for a pan-tilt-zoom (PTZ) camera is proposed. Using only two images, we are able to solve for 5 camera parameters by trading off linearity with polynomial equations. Our solution is based on using a sequence of Givens rotations, whereby we decompose the infinite homography into a pair of projectively equivalent upper-triangular matrices that provide up to 5 constraints directly on the camera parameters.

Self-Configuring Camera Network: In order to monitor sufficiently large areas of interest for surveillance or any event detection, we need to look beyond stationary cameras and employ

an automatically configurable network of non-overlapping cameras. Moreover, features like zooming in/out, readily available in security cameras these days, should be exploited in order to focus on any particular area of interest if needed. A practical framework is presented that determines the geometry of such a dynamic camera network. It is shown that only one automatically computed vanishing point and a line lying on any plane orthogonal to the vertical direction is sufficient to infer the dynamic network configuration. Our method generalizes previous work which considers restricted camera motions [AHR01]. Using minimal assumptions, we are able to successfully demonstrate promising results on synthetic as well as on real data.

1.3 Applications

The theory presented here can be applied to solving many of the other problems in the field of computer vision and photogrammetry. This section analyzes four of the many possible uses, which will be described later in this thesis.

1.3.1 Path Modeling

We address the issue of path surveillance in a single uncalibrated and calibrated camera. We propose a novel solution for detecting unusual behaviors of objects as they pass through a scene. The method consists of a path building training phase and a testing phase. During the unsupervised training phase, a weighted graph is constructed with trajectories represented by the nodes and weights determined by a similarity measure. Normalized-cuts are used recursively to partition the graph into prototype paths. Each partition represents a group of trajectories, which in turn

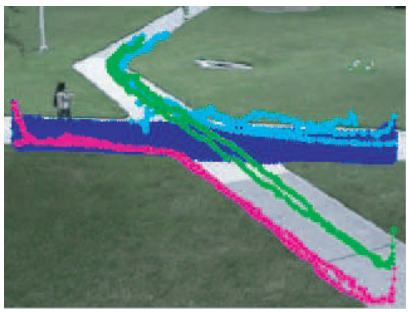


Figure 1.2: An example of different paths followed by objects in a scene. Different colors indicate different paths.

is represented by a path envelope and an average trajectory. During the testing phase we seek a relation between the input trajectories derived from a sequence and the prototype path models using our similarity measures. The proposed method is used to generate a topology of a scene and calculate probabilities for predicting object behavior. Real-world pedestrian sequences are used to demonstrate the practicality of our method. Figure 1.2 shows an example of multiple paths extracted from a video sequence.

1.3.2 Registration To Satellite Imagery

Registration to the satellite imagery gives a global view of the scene being observed. Using the calibration techniques presented in this thesis, the images can be rectified to one that would have been obtained from a fronto-parallel view of the plane for a good registration to the aerial imagery. To make this process automatic (i.e. without having to manually specify the Euclidean world coordinates of points), the estimated affine and the projective components of the transformation can be combined together to efficiently metric rectify the video-sequence such that the only unknown transformation is a similarity transformation.

1.3.3 GPS Coordinate Estimation

We introduce a novel application to the field of vision-based video forensics. By using only computer vision techniques, we are able to estimate the GPS coordinates of the camera location. Once we have a calibrated camera, we make some measurements on shadow trajectories to obtain the geo-latitude of the camera. This step only requires three shadow trajectory points. We also obtain the day (up to sign ambiguity) on which these images were taken and the declination angle of the earth when these pictures were taken. This is possible by integrating techniques from the field of astronomy and computer vision. We also discuss how the longitude can be obtained if more information is available.

1.3.4 Mixed-Reality

To demonstrate the broader applicability of our proposed work, we present a practical framework for registering a Mixed Reality (MR) environment of an arbitrary number of participants. Each participants wears a head mounted display, which consists of a pair of stereo cameras. Participants are assumed to be moving freely in 3D space and multiple HMDs need not have a common Field of View (FoV). We show that the plane at infinity and a common vertical vanishing point can be use to determine the exact orientation of all HMDs with respect to each other, and establish a common reference frame up to translation.

1.4 Outline Of The Thesis

This thesis is divided into four parts:

Part I: Introduction and Background

A brief history of the projective geometry is presented in Chapter 2 followed by some basic concepts in the projective geometry of 2-space and 3-space. The pinhole camera model is described and its various parameters are introduced. The absolute conic, lying on the plane at infinity, at its use in camera calibration is highlighted. The epipolar geometry, arising between different views of the camera or between multiple cameras, is elaborated. These concepts serve as a foundation for the rest of the thesis.

Part II: Camera Calibration

Camera calibration is the process of extracting *intrinsic* and *extrinsic* camera parameters. Calibration is an obligatory process in computer vision in order to obtain a Euclidean structure of the scene (up to a global scale), and to determine rigid camera motion.

This part presents novel solutions to calibrate any camera present in a network. Therefore, this part applies to any single camera, not the network as a whole. Camera calibration techniques can be broadly classified into two categories:

1. **Scene Based Calibration:** Calibration by observing a calibration object whose geometry in the 3-D space is known. The original work in this category is that of Tasi [Tsa87], where the calibration object consists of two or more planes set orthogonal to each other.

2. **Self-Calibration:** The metric properties of the cameras are determined directly from constraints on the internal and/or external parameters [Tri97, FLM92, PKG99, HB06, AHR01, Stu97b]. No calibration objects are required in these techniques. Simply by moving a camera in a static scene the rigidity of the scene provides constraints that are used to calibrate the camera.

Another *intermediate* technique for camera calibration is based on *scene constraints*. The knowledge of scene geometry, e.g. *vanishing points* or *vanishing lines*, is used to impose constraints on the camera parameters [SH04, LZ99]. Due to their ease of use and wide applicability, the camera calibration methods presented in this work are all self-calibration or scene constraints based.

Chapter 3 revisits the role of the image of the absolute conic (IAC) in recovering the camera geometry [JF06a]. New constraints on IAC are derived that advance our understanding of its underlying building blocks. These new constraints are shown to be intrinsic to IAC, rather than exploiting the scene geometry or the prior knowledge on the camera. We provide geometric interpretations for these new intrinsic constraints, and show their relations to the invariant properties of the IAC. This in turn provides a better insight into the role that IAC plays in determining the camera internal geometry.

Chapter 4 shows that a set of six or more photographs of shadow trajectories of stationary objects in a scene are sufficient to accurately calibrate the camera [JF07d]. Calibration is possible after the line at infinity has been recovered. The chapter provides two methods to recover this line which is used with the concepts presented in Chapter 3 to perform calibration.

Chapter 5 addresses a practical situation where a stationary camera is observing pedestrians. We present a robust linear solution to the problem of camera calibration from observing pedestrians by adopting a formulation that is more general than existing methods [LZN02, KM05]. The uniqueness of formulation lies in recognizing two harmonic homologies present in the geometry induced by walking pedestrians, and then using properties of these homologies to obtain linear constraints on the unknown camera parameters for arbitrarily walking pedestrians. This work has

been published in various conferences [JFa, JF06b, JAS07]

Chapter 6 describes a camera calibration method when the camera is freely moving [JCF06a, JCF07]. We show that point correspondences between a pair of images, and the fundamental matrix computed from these point correspondences, are sufficient to recover the internal parameters of a camera. The main contribution of this chapter is the development of a global linear solution which is based on the well-known Kruppa equations. We introduce a formulation different from the Huang-Faugeras constraints.

Chapter 7 describes a novel method for calibrating a pan-tilt-zoom (PTZ) camera from only two images by trading off linearity with polynomial equations [JFb, JF07a]. Our solution is based on using a sequence of Givens rotations, whereby we decompose the infinite homography into a pair of projectively equivalent upper-triangular matrices that provide up to 5 constraints directly on the camera parameters.

Part III: Network Calibration

This part focuses on a network of multiple cameras. In order to monitor sufficiently large areas of interest for surveillance or any event detection, we need to look beyond stationary cameras and employ an automatically configurable network of non-overlapping cameras. These cameras need not have an overlapping *Field of View* (FoV) and should be allowed to move freely in space if desired. Moreover, features like zooming in/out, readily available in security cameras these days, should be exploited in order to focus on any particular area of interest if needed.

Chapter 8 presents a practical framework to use calibrated (possibly moving and zooming) cameras and determine their absolute and relative orientations, assuming that their relative position is known (using either survey points, GPS, or by initialization). It is shown that only one

automatically computed vanishing point and a line lying on any plane orthogonal to the vertical direction is sufficient to infer the dynamic network configuration. The method generalizes previous work which considers restricted camera motions. Using minimal assumptions, we are able to successfully demonstrate promising results on synthetic as well as on real data. This chapter is the result of several publications [JCF07, JCF06c].

Part IV: Applications

Previous chapters described camera calibration methods for various scenarios and a network configuration method. These novel methods can be applied to solve different problems in video content analysis and video forensics. This chapter aims to describe some of the applications of our proposed work that we have investigated.

Chapter 9 describes application to **Euclidean path modeling** for video surveillance. We present a novel yet simple method to model the behavior of pedestrians in a scene. Using pedestrians for camera calibration, the trajectories of the tracked pedestrians are metric rectified to remove projective distortion from the trajectories. These metric rectified trajectories represent a *truer* picture of the data. This chapter is the result of several publications [JFa, JF07c, JF07b, JJS04]

We also described how a modeled scene can be registered to satellite imagery for a global view of the scene. Results are presented for single and multiple camera systems.

Chapter 10 presents a technique for **GPS coordinate estimation**. We show that once a camera is calibrated from observing shadow trajectories (Chapter 4), we can recover the GPS coordinates of the camera location. Determining the GPS coordinates and the date of the year from shadows in images is a new video forensic concept that we introduce in our work. This is possible by incorporating techniques from the field of astronomy and computer vision. This chapter is submitted for

publication [JF07d]

Chapter 11 describes application to a Mixed Reality (MR) environment [JCF06b]. We show that the method described in Chapter 8 can be used to configure a MR environment, where multiple agents are using head mounted display (HMD) units.

1.5 Notations

Although this thesis adopts the standard notations used in computer vision literature, for example [HZ04], we briefly highlight the most important notations:

- sets are denoted by symbols in "caligraph" or "script" font (e.g. $\mathcal{S}$).
- matrices by using bold upper case symbols (e.g. $\mathbf{K}, \mathbf{P}$).
- scalars by normal face symbols (e.g f, λ).
- vectors, points, and lines are presented in homogeneous coordinates using lower case bold symbols (e.g. $\mathbf{x}, \boldsymbol{\omega}$). At locations, the homogeneous coordinates are also denoted by a tilde ($\tilde{}$).
- 3D elements, like points and lines by upper case bold symbols (e.g. $\mathbf{\Pi}, X$).
- equality up to a multiplication by a non-zero scalar factor in a homogeneous coordinate system as $\sim$.
- skew symmetric matrix is denoted as $[\mathbf{e}]_\times$ for a vector $\mathbf{e}$[1]

[1]If $\mathbf{e} = (\mathbf{e_1}, \mathbf{e_2}, \mathbf{e_3})^\mathbf{T}$ is a 3-vector, then we can define a corresponding skew-symmetric matrix as: $[\mathbf{e}]_\times = \begin{bmatrix} 0 & -e_3 & e_2 \\ e_3 & 0 & -e_1 \\ -e_2 & e_1 & 0 \end{bmatrix}$.

The terms multi-camera, multiple cameras and networked cameras are used interchangeably. Similarly, calibrating and configuring a camera network shall be used interchangeably as well. More notations shall be introduced at appropriate places when necessary.

CHAPTER 2
BACKGROUND: PROJECTIVE GEOMETRY

Projective geometry deals with the geometry of straight lines. We no longer deal with a right-angled triangle or a circle, but with triangles and conics.

2.1 A Bit Of History

Projective Geometry: While Euclid's geometry may be defined as the geometry of lines and circles, the projective geometry can be defined as geometry of the *straight lines* alone. All the propositions for projective geometry are in fact old and may be traced back to Euclid (285 B.C.), to Apollonius of Perga (247 B.C), to Pappus of Alexandria (4^{th} century C.E.); to Desargues of Lyons ($1593-1662$); to Pascal (1623-1662); to de la Hire (1640-1718); to Newton ($1642-1727$); to Maclaurin ($1698-1746$); and to J.H. Lambert ($1728-1777$). The theories and methods derived from these propositions are called *modern* because they have been discovered or perfected by mathematicians of an age nearer to ours, such as Carnot, Brianchon, Poncelet, Möbius, Steiner, Chasles, Staudt, etc. [Cre85].

Plane projective geometry deals with the projection of a 3-Dimensional world onto a 2-Dimensional plane. The projective geometry deals with triangles, quadrangles and so on, but not with right-angled triangles or parallelograms, and so on. This is due to our focus concern with geometrical properties only that remain unchanged by the *central projection*. The motivation for this kind of geometry came from fine arts. In 1425 Italian architect Brunelleschi began to discuss the geometrical theory of perspective, which later was consolidated by Alberti [FL01], see Figure 2.1 for an illustration of Alberti's grid.

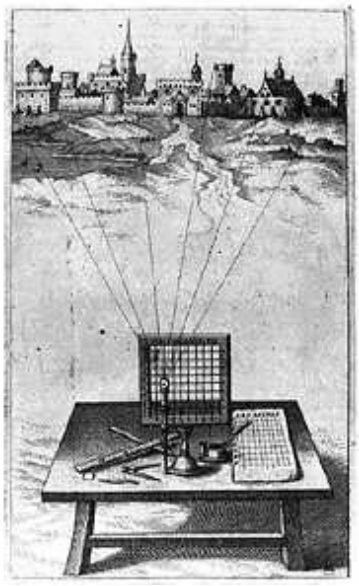
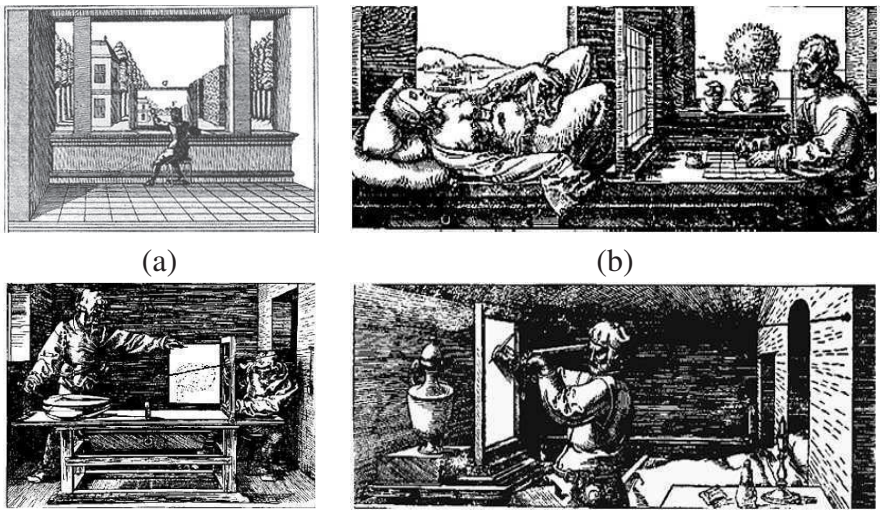


Figure 2.1: Alberti's Grid - c.1450 (also known as Alberti's Veil).

Similarly, an ellipse or a parabola are simply *conics* in a projective geometry. Although conics were studied by Manaechmus, Euclid, Archimedes and Apollonius in the early $4^{th} - 5^{th}$ century B.C., it was Pappus of Alexandria in third century C.E. who truly discovered the projective theorems [Cox74]. J.V. Poncelet was the first to prove these theorems by purely projective reasoning.

Almost two hundred years before Poncelet, the concept of *point at infinity* occurred independently to two scientists Johann Kepler and Girard Desargues. Desargues declared that, "parallel lines have a common end at an infinite distance". And, "when no point of a line is at a finite distance, the line itself is at an infinite distance". This work laid out the foundation for the concept of *line at infinity*, discovered later by Poncelet. This concept justifies our assumption that if coplanar lines have no point in common, they intersect at a point at infinity.

The last traces of dependencies on Euclidean geometry were removed when Felix Klein, in 1871, provided an algebraic foundation for the projective geometry by introducing *homogeneous*



(a) (b)

(c) (d)

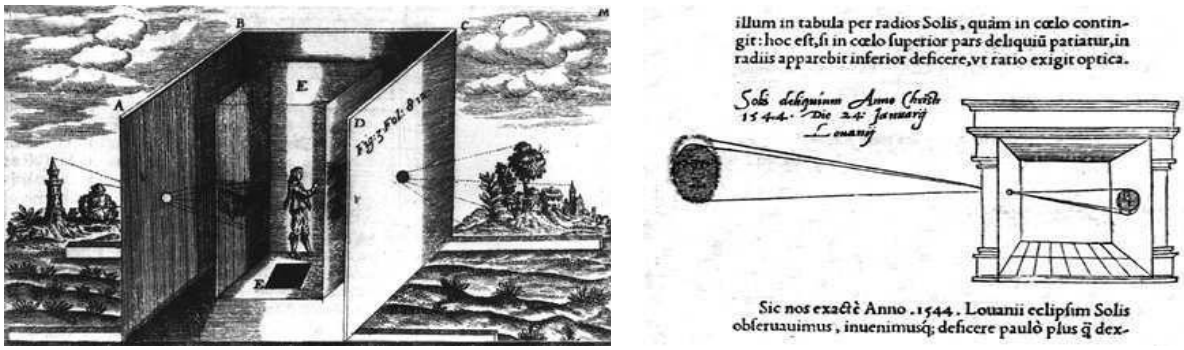
Figure 2.2: **Perspective Frames:** (a) A painter incorporating perspective effect into his painting. (b) Albrecht Dürer's interpretation of "The Draftsman's Net". (c) Albrecht Dürer's Perspective Machine of 1525 demonstrates the principle of ray tracing.(d) Albrecht Dürer's interpretation of "Jacob de Keyser's Invention".

coordinates.

The *principle of duality* - every statement about points and lines (in a plane) can be replaced by a dual statements about lines and points - was introduced by Poncelet, later elaborated by J.D. Gergonne ($1771 - 1859$).

Pinhole Camera: Along the time when the theory of projective geometry was being developed, *perspective machines* were being developed to help painters accurately produce life like image of the real world [FL01]. In this kind of machines, the eye of the painter was generally fixed and a device was used to materialize the visual ray with the image plane, illustrating the geometry of central projection. Figure 2.2 depicts some of the devices invented for painters to add linear perspective effects to their work.

The *camera obscura* (Latin for 'dark room') was the ancestor of the modern camera. We find



(a) (b)

Figure 2.3: **Earlier Pinhole Cameras:** (a) Camera Obscura, Athanasius Kircher, 1646. (b) Camera Obscura, Reinerus Gemma-Frisius, 1544.

casual references by Aristotle (Problems, ca 330 B.C.), and Euclid. Abu Ali Al-hasen Ibn-Alhasen is the first to show how an image is formed on the eye, using the camera obscura as an analog. (1038), printed in *Opticae Thesaurus Alhazani* in 1572. The camera obscura would be a dark room where the user would enter. The light entering through a small hole would produce the inverted image on the opposite wall. Two examples of different camera obscura invented are shown in Figure 2.3.

2.2 Camera Model

In computer vision and other related fields, there are numerous different camera models which model the imaging process by mapping points from the 3D world to 2D points on an image plane. This process of image formation must be modeled in a rigorous mathematical fashion. The choice of an appropriate camera model depends on several factors including the accuracy required in the mapping, the actual camera used, and the relationship between the camera and the scene being viewed.

Our work focuses mainly on the pinhole camera (or *central projection*), described below in Section 2.2.2, and is the most commonly used camera model in the computer vision community (e.g. [WMC03, CRZ00, CF04b, Zha02, CBP05, AZH96]). However, as the theory of camera calibration is based on Projective Geometry, the important concept of homogeneous coordinates is described first.

2.2.1 Homogeneous Coordinates

Suppose we have a point (x, y) on a Euclidean plane. In order to represent that point in a projective space, we add a third coordinate: $(x, y, 1)$. The overall scaling is unimportant i.e. $(x, y, 1)$ is same as $\lambda(x, y, 1)$ for any non-zero λ.

More formally, the *homogeneous coordinate* set for a point $\mathbf{X}$ in *n*-dimensional space with Euclidean coordinates given by the n-tuple $(X_1, X_2, ..., X_n) \in \mathbb{R}^n$ is a $(n + 1)$-tuples $\{w(X_1, X_2, ..., X_n, X_{n+1}) \in \mathbb{R}^{n+1} \setminus \{0, 0, ..., 0\}, \forall w \neq 0\}$. Conversely, given the homogeneous coordinates $\{w(X_1, X_2, ..., X_n, X_{n+1}) \in \mathbb{R}^{n+1} \setminus \{0, 0, ..., 0\}, \forall w \neq 0\}$ of a point X in n dimensional space, Euclidean coordinates are derived as: $(X_1, X_2, ..., X_n)/X_{n+1}$, if $X_{n+1} \neq 0$. The special case when $X_{n+1} = 0$ happens when the point is at infinity; this can not be represented in Euclidean space.

Two $n + 1$ vectors $\mathbf{x} = [\mathbf{x_1}, ...\mathbf{x_{n+1}}]^\mathbf{T}$ and $\mathbf{x'} = [\mathbf{x'_1}, ...\mathbf{x'_{n+1}}]^\mathbf{T}$ represent the same point in projective space if and only if $\exists\lambda \neq 0$ such that $x_i = \lambda x'_i$ for $1 \leq i \leq n + 1$.

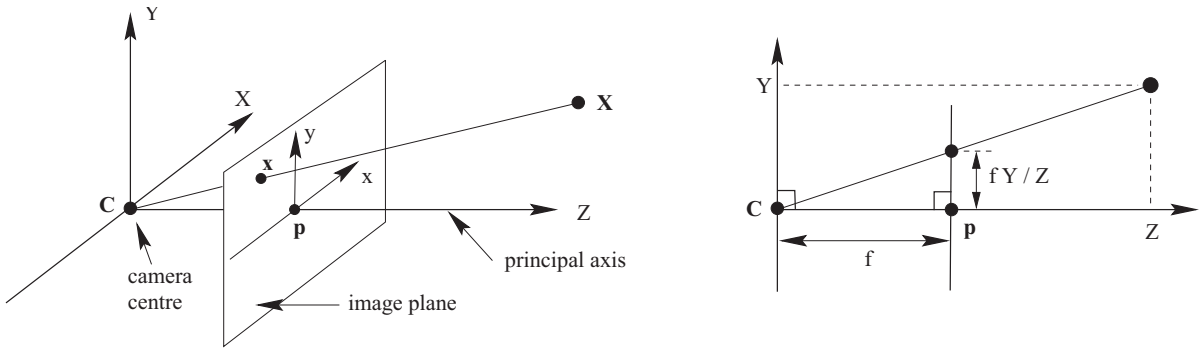


Figure 2.4: **Pinhole Camera Model:** The camera center is denoted by $\mathbf{C}$. The 3D point $\mathbf{X}$ is projected onto a points $\mathbf{x}$ on the image plane. The image plane is placed in front of the the camera center. The camera in the figure is placed at the origin of the coordinate system (Figure courtesy of [Har]).

2.2.2 Pinhole Camera Model

The most general linear camera model is the pinhole camera model. This model is a perspective projection of the world to the image plane. The pinhole camera model does not model the non-linear distortions introduced by the camera. A 3D point in projective space $\mathbb{P}^3$ is projected onto a plane in $\mathbb{P}^2$ by means of straight visual rays (cf. Figure2.4). The corresponding point is the intersection of the image plane with the visual ray connecting the 3D point to the optical center.

Formally, represented in homogeneous coordinates, the projection of a 3**D** scene point $\mathbf{X} \sim \begin{bmatrix} X & Y & Z & 1 \end{bmatrix}^{\mathbf{T}}$ onto a point in the image plane $\mathbf{x} \sim \begin{bmatrix} x & y & 1 \end{bmatrix}^{\mathbf{T}}$, for a perspective camera can be modeled by the central projection equation:

$$\mathbf{x} \sim \underbrace{\mathbf{K}\begin{bmatrix} \mathbf{R} & |-\mathbf{RC} \end{bmatrix}}_{\mathbf{P}} \mathbf{X}, \mathbf{K} = \begin{bmatrix} f & \gamma & u_o \\ 0 & f\lambda & v_o \\ 0 & 0 & 1 \end{bmatrix} \tag{2.1}$$

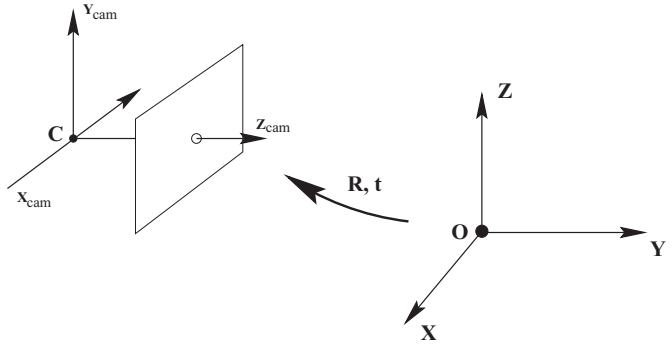


Figure 2.5: Euclidean transformation of the camera coordinate frame w.r.t. to the world coordinate frame.(Figure courtesy of [Har]).

where $\sim$ indicates equality up to a non-zero scale factor and $\mathbf{C} = \begin{bmatrix} C_x & C_y & C_z \end{bmatrix}^{\mathbf{T}}$ represents the Euclidean coordinates of the camera center. Here $\mathbf{R} = \mathbf{R_x R_y R_z} = \begin{bmatrix} \mathbf{r_1} & \mathbf{r_2} & \mathbf{r_3} \end{bmatrix}$ is the rotation matrix and $-\mathbf{RC}$ is the relative translation between the world origin and the camera center (cf. Figure2.5). The upper triangular 3×3 matrix $\mathbf{K}$ encodes the five intrinsic camera parameters: focal length f, aspect ratio λ, skew γ and the principal point at (u_o, v_o) [WS94].

The matrix $\mathbf{P}$ is denoted as the projective camera matrix, and the matrix $\mathbf{K}$ corresponds to the matrix of *intrinsic* parameters. The matrix $\mathbf{R}$ and the vector $-\mathbf{RC}$ are jointly called the *extrinsic* or *external* parameters. If the matrix $\mathbf{K}$ is known, the camera is said to be calibrated. Hereafter, the expressions "the camera $\mathbf{P}$" and "the intrinsic parameters $\mathbf{K}$" should be read as "the camera with projective camera matrix given by $\mathbf{P}$" and "the intrinsic parameters represented by the matrix $\mathbf{K}$", respectively.

Once $\mathbf{P}$ is obtained, the camera model is said to be completely determined. The matrix can be computed from the relative positioning of the world points and camera center, and from the camera internal parameters; however, it can also be computed directly from image-to-world point correspondences [AK71].

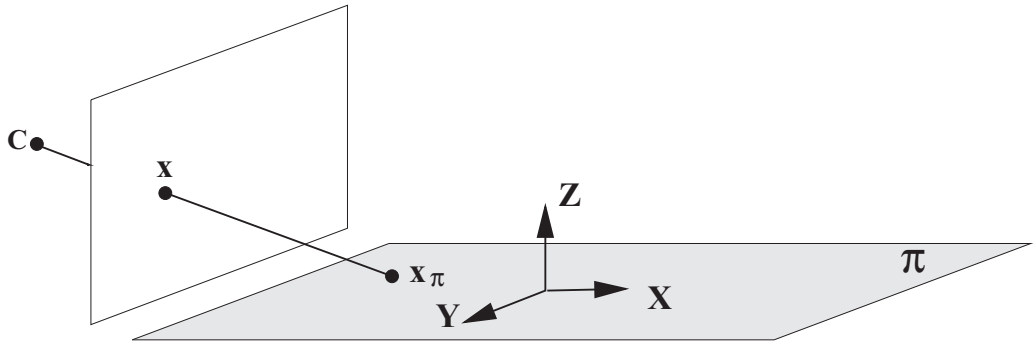


Figure 2.6: **Plane-to-plane homography**: Points on one plane are projected by a plane-to-plane homography to points on another plane. The camera center is denoted by $\mathbf{C}$ (Figure courtesy of [Har]).

The projective camera describes a pinhole camera model by introducing the internal camera parameters to account for the real camera characteristics. Physical lenses, however, introduce non-linear distortions in the image, often modeled by radial distortion [SGN03]. Distortion will be ignored in the current work - it is insignificant in the example images used and will be removed when necessary [DF95].

2.2.3 Planar Homography

An interesting specialization of the perspective projection is the *plane-to-plane* projection (cf. Figure 2.6). Points on one plane are projected by a plane-to-plane homography to points on another plane [SK79]. This homography, also known as the plane projective transformation or collineation, is a bijective mapping. Planar homography arises generally when the camera is projecting a planar scene, for e.g. side of a building or looking at the ground plane.

Formally, a 3D point $\mathbf{X_i}$ lying on a plane is projected to a point $\mathbf{x_i}$ on the image plane as:

$$\mathbf{x_i} = \mathbf{H}\mathbf{X_i} \tag{2.2}$$

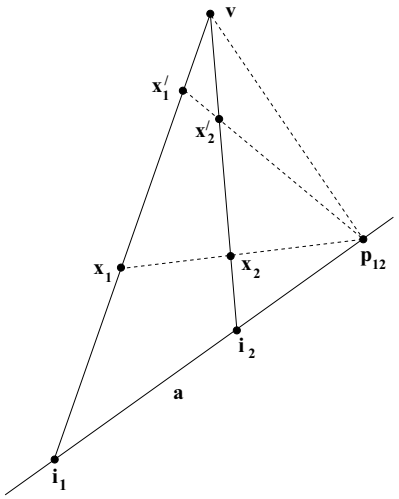


Figure 2.7: **Planar Homology**: A planar homology is defined by a vertex $\mathbf{v}$ and an axis $\mathbf{a}$. μ, the characteristic ratio, can be determined by the cross ratio $< \mathbf{v}, \mathbf{x}'_1, \mathbf{x}_1, \mathbf{i}_1 >$ of the four aligned points. The point $\mathbf{x}'_1$ is projected on to the point $\mathbf{x}_1$, and similarly $\mathbf{x}'_2$ on to $\mathbf{x}_2$. (Figure courtesy of [Har]).

where $\mathbf{H}$ is a 3×3 homogeneous planar projection matrix describing the homography. The world points are represented in homogeneous coordinates $\mathbf{X} = (\mathbf{X}, \mathbf{Y}, \mathbf{W})^{\mathbf{T}}$ (omitting the Z-component) and the 2D image points are denoted as $\mathbf{x} = (\mathbf{x}, \mathbf{y}, \mathbf{w})^{\mathbf{T}}$, respectively.

Since the homography is bijective, it follows that $\mathbf{X_i} = \mathbf{H}'\mathbf{x_i}$ is also valid, where $\mathbf{H}' = \mathbf{H}^{-1}$. Computation of $\mathbf{H}$ is similar to that of $\mathbf{P}$. In particular, $\mathbf{H}$ has eight degrees of freedom (nine parameters minus an overall scale), hence it can be shown that at least four world-to-image feature points suffice to define the homography [HZ04].

2.2.4 Planar Homology

Planar homology is a plane projective transformation and an specialization of the homography. It is characterized by a line of fixed points, the *axis*, and a distinct fixed point not on the line, the *vertex*. Planar homology arises in many situation, for instance, when different light sources cast

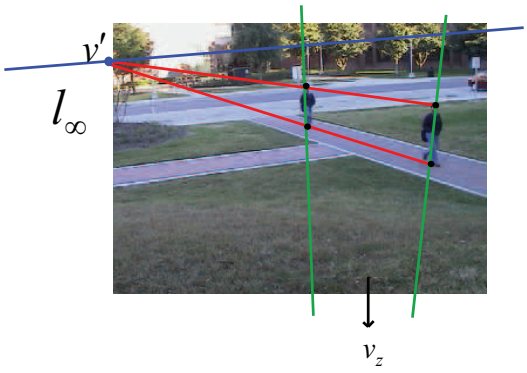


Figure 2.8: **Calibration Geometry**: A pedestrian, detected at two different time instances, provides vertical vanishing point ($\mathbf{v_z}$) and another vanishing point ($\mathbf{v}'$) lying on the horizon line of the ground plane. As a result, two harmonic homologies exist in the scenario: one, having $\mathbf{v}'$ as its vertex, and the other with $\mathbf{v_z}$ as it vertex.

shadows of an object onto the same plane.

The planar homology is defined by a 5 d.o.f. 3×3, matrix $\mathbf{H}$, and can be parameterized as:

$$\mathbf{H} = \mathbb{I} - (\mu - \mathbf{1})\frac{\mathbf{v}\mathbf{a}^\mathbf{T}}{\mathbf{v}^\mathbf{T}\mathbf{a}} \tag{2.3}$$

where μ is the characteristic ratio that can be computed as the cross ratio of the four aligned points as shown in Figure 2.7, and $\mathbf{v}$ and $\mathbf{a}$ represent the vertex and the axes of the homology, respectively. Planar homology contains one distinct and two repeated eigenvalues i.e. eigenvalues are $\{\lambda_1 = \mu, \lambda_2 = 1, \lambda_3 = 1\}$ and the eigenvectors are $\{\mathbf{e_1} = \mathbf{v}, \mathbf{e_2} = \mathbf{a_1^\perp}, \mathbf{e_3} = \mathbf{a_2^\perp}\}$, such that $\mathbf{a} = \mathbf{a}_1^\perp \times \mathbf{a}_2^\perp$.

Harmonic Homology: A specialization of the planar homology is the case when the cross ratio is harmonic i.e. $\mu = -1$. This planar homology is called the planar harmonic homology and has 4 degrees of freedom (one less due to the known μ). This special case has the parametrization:

$$\mathbf{H} = \mathbf{H^{-1}} = \mathbb{I} - \mathbf{2}\frac{\mathbf{v}\mathbf{a^T}}{\mathbf{v^T}\mathbf{a}} \tag{2.4}$$

In perspective images of a planar object with bilateral symmetry, corresponding points in the images are related by a harmonic homology. Figure 2.8 shows an example of harmonic homology related to our work. A pedestrian, detected at two different time instances, provides vertical vanishing point ($\mathbf{v_z}$) and another vanishing point ($\mathbf{v}'$) lying on the horizon line of the ground plane.

2.3 Image Of The Absolute Conic

Consider the equation of a conic C:

$$ax_1^2 + 2bx_1x_2 + 2cx_1 + dx_2^2 + 2ex_2 + f = 0$$

In homogeneous coordinates this becomes $\mathbf{x^T C x} = \mathbf{0}$, where $\mathbf{C} = \begin{bmatrix} a & b & c \\ b & d & e \\ c & e & f \end{bmatrix}$.

The matrix $\mathbf{C}$ is the homogeneous representation of the conic C. The equation of an n-dimensional quadric, in general, is given as:

$$\mathbf{X^T Q X} = \mathbf{0}$$

where $\mathbf{Q}$ is a $(n+1) \times (n+1)$ symmetric matrix.

The *Absolute Conic* (AC) Ω_∞ is a point conic on the plane at infinity $\mathbf{\Pi}_\infty$. The *Image of the Absolute Conic* (**IAC**), denoted by $\boldsymbol{\omega}$, is the conic $\boldsymbol{\omega} = \left(\mathbf{K K^T}\right)^{-\mathbf{1}} = \mathbf{K^{-T} K^{-1}}$, where $\mathbf{K}$ is the

camera parameter matrix. Thus $\boldsymbol{\omega}$ only depends on the internal parameters $\mathbf{K}$ of the matrix $\mathbf{P}$. $\boldsymbol{\omega}$ can be expanded up to a non-zero scale as:

$$\boldsymbol{\omega} = \begin{bmatrix} 1 & -\frac{\gamma}{\lambda f} & -\frac{-v_o\,\gamma+u_o\,\lambda\,f}{\lambda f} \\ -\frac{\gamma}{\lambda f} & \frac{\gamma^2+f^2}{\lambda^2 f^2} & -\frac{v_o\,f^2+v_o\,\gamma^2-u_o\,\lambda\,f\gamma}{\lambda^2 f^2} \\ -\frac{-v_o\,\gamma+u_o\,\lambda\,f}{\lambda f} & -\frac{v_o\,f^2+v_o\,\gamma^2-u_o\,\lambda\,f\gamma}{\lambda^2 f^2} & \frac{\lambda^2 f^4+f^2 {v_o}^2+\gamma^2 {v_o}^2+\lambda^2 f^2 {u_o}^2-2\,u_o\,v_o\,\lambda\,f\gamma}{\lambda^2 f^2} \end{bmatrix} \tag{2.5}$$

The dual image of the absolute conic (the DIAC) may be defined as:

$$\boldsymbol{\omega}^* = \boldsymbol{\omega}^{-1} = \mathbf{K}\mathbf{K}^{\mathbf{T}}$$

The conic $\boldsymbol{\omega}^*$ is a dual (line) conic, whereas $\boldsymbol{\omega}$ is a point conic.

The aim of camera calibration is to determine the calibration matrix $\mathbf{K}$. Instead of directly determining $\mathbf{K}$, it is common practice [AHR01] to compute the symmetric matrix $\mathbf{K}^{-\mathbf{T}}\mathbf{K}^{-\mathbf{1}}$ or its inverse (the dual image of the absolute conic). The obtained matrix, $\boldsymbol{\omega}$ or $\boldsymbol{\omega}^*$, can then be decomposed uniquely using the Cholesky Decomposition [PFT88] to obtain the calibration matrix $\mathbf{K}$. The matrix $\mathbf{K}$ can also be obtained uniquely from $\boldsymbol{\omega}$, as shown by [Zha00, CSS05]:

$$\lambda = \sqrt{1/(\omega_{22} - \omega_{12}^2)}$$

$$v_o = (\omega_{12}\omega_{13} - \omega_{23})/(\omega_{22} - \omega_{12}^2)$$

$$u_o = -(v_o\omega_{12} + \omega_{13})$$

$$f = \sqrt{\omega_{32} - \omega_{13}^2 - v_o(\omega_{12}\omega_{13} - \omega_{23})}$$

$$\gamma = -f\lambda\omega_{12} \tag{2.6}$$

where the subscripts of ω_{ij} denote an element's row i and column j in matrix $\boldsymbol{\omega}$.

2.4 Vanishing Points And Vanishing Lines

Vanishing points and vanishing lines are extremely powerful geometric cues. These entities convey a lot of information about the scene. These points and lines can be estimated directly from the images with no explicit knowledge required about the relative geometry between the camera and the viewed scene [MK95, LZ98, Shu99].

As shown in Figure 2.9, image of parallel lines in the world intersect at a common points called the vanishing point. Similarly, vanishing points of a set of coplanar parallel lines in different directions meet at a common line, called the vanishing line of their common plane.

In $\mathbb{P}^3$, the plane at infinity $\mathbf{\Pi}_\infty$ is the plane of directions - i.e. all parallel lines meet on $\mathbf{\Pi}_\infty$ at one common point. A vanishing point is simply the projection of this point on the image plane. Thus a vanishing point depends only on the direction of a line, not on its position. Thus, if a line has a direction $\mathbf{d}$, then it intersect $\mathbf{\Pi}_\infty$ at a point $\mathbf{X}_\infty = (\mathbf{d}^T, 0)^T$. Then the vanishing point, $\mathbf{v}$, is

Figure 2.9: **Vanishing Point**: (left) Image of parallel lines in the world intersect at a common point called the vanishing point. (right) Set of more than one parallel lines in different direction meet at a common line, the vanishing line.

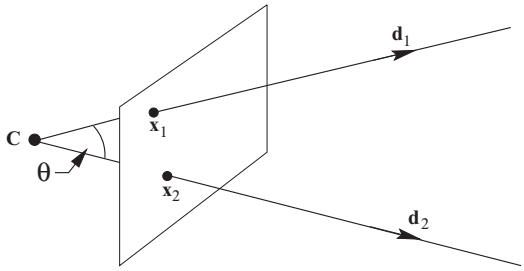


Figure 2.10: The angle between two rays.

given as:

$$\mathbf{v} \sim \mathbf{P}\mathbf{X}_\infty = \mathbf{K}\,[\mathbf{I}|\mathbf{0}] \begin{pmatrix} \mathbf{d} \\ \\ 0 \end{pmatrix} = \mathbf{K}\mathbf{d}$$

Thus, the vanishing point of a line with direction $\mathbf{d}$ in $\mathbb{P}^3$ is the intersection of the ray with the image plane at a point $\mathbf{v} = \mathbf{K}\mathbf{d}$. Conversely, the direction $\mathbf{d}$ is obtained from the vanishing points as $\mathbf{d} = \mathbf{K}^{-1}\mathbf{d}$ up to a scale.

The angle between two rays $\mathbf{d_1}$ and $\mathbf{d_2}$ corresponding to image points $\mathbf{x_1}$ and $\mathbf{x_2}$ respectively, may thus be obtained from the cosine formula for the angle between the two vectors:

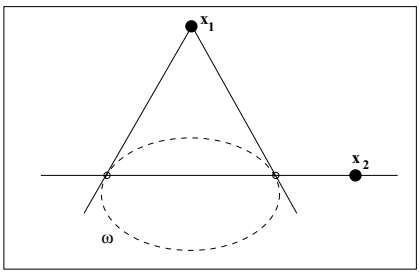


Figure 2.11: Two vanishing points, $\mathbf{x_1}$ and $\mathbf{x_2}$, of mutually orthogonal directions are said to be conjugate w.r.t. the conic $\boldsymbol{\omega}$.

$$\cos\theta = \frac{\mathbf{d_1^T d_2}}{\sqrt{\mathbf{d_1^T d_1}}\sqrt{\mathbf{d_2^T d_2}}} = \frac{(\mathbf{K^{-1}x_1})^{\mathbf{T}}(\mathbf{K^{-1}x_2})}{\sqrt{(\mathbf{K^{-1}x_1})^{\mathbf{T}}(\mathbf{K^{-1}x_1})}\sqrt{(\mathbf{K^{-1}x_2})^{\mathbf{T}}(\mathbf{K^{-1}x_2})}} = \frac{\mathbf{x_1^T(K^{-T}K^{-1})x_2}}{\sqrt{\mathbf{x_1^T(K^{-T}K^{-1})x_1}}\sqrt{\mathbf{x_2^T(K^{-T}K^{-1})x_2}}} \tag{2.7}$$

This equation shows that if $\boldsymbol{\omega} = (\mathbf{K^{-T}K^{-1}})$ is known, then the angle between rays can be measured from their corresponding image points. In other words, a calibrated camera is a direction tensor, acting as a 2D protractor. In the case when two vanishing points $\mathbf{v_1}$ and $\mathbf{v_2}$ represent mutually orthogonal directions, i.e. $\cos\theta = 0$, Eq. 2.7 reduces to $\mathbf{v_1^T \boldsymbol{\omega} v_2} = \mathbf{0}$. Geometrically, the two vanishing points are said to be conjugate w.r.t. the conic $\boldsymbol{\omega}$, as shown in the Figure 2.11. This orthogonality relation puts a constraint on $\boldsymbol{\omega}$, and subsequently on $\mathbf{K}$, that are linear in elements of $\boldsymbol{\omega}$. Leibowitz and Zisserman [LZ99] were the first to formulate the calibration constraints provided by vanishing points of mutually orthogonal directions in terms of $\boldsymbol{\omega}$, [CT90] were the first to use vanishing points for camera calibration. Some of the methods proposed by other researcher using orthogonality condition include [Zha00, CBP05, Stu99, GS03, CDR99, WMC03, GP00, CS05].

2.5 Circular Points

Under any similarity transformation, two points, $\mathbf{I}$ and $\mathbf{J}$, on the line at infinity $\mathbf{l}_\infty$ are fixed. These points are called the *circular points*, with the canonical coordinates

$$\mathbf{I} = \begin{pmatrix} 1 \\ i \\ 0 \end{pmatrix} \qquad \mathbf{J} = \begin{pmatrix} 1 \\ -i \\ 0 \end{pmatrix} \tag{2.8}$$

The circular points are a pair of ideal complex conjugate points. Thus $\mathbf{l}_\infty$ intersects $\boldsymbol{\omega}$ at two points, $\mathbf{I}$ and $\mathbf{J}$, giving rise to two constraints on the elements of $\boldsymbol{\omega}$:

$$\mathbf{I}^\mathbf{T}\boldsymbol{\omega}\mathbf{I} = \mathbf{0} \qquad \mathbf{J}^\mathbf{T}\boldsymbol{\omega}\mathbf{J} = \mathbf{0} \tag{2.9}$$

In practice, all the circular point information is contained in one of the complex conjugate points. Writing out the real and imaginary parts of either $\mathbf{I}^\mathbf{T}\boldsymbol{\omega}\mathbf{I} = \mathbf{0}$ or $\mathbf{J}^\mathbf{T}\boldsymbol{\omega}\mathbf{J} = \mathbf{0}$ yields two linear expressions on the elements of $\boldsymbol{\omega}$.

2.6 Epipolar Geometry

A point $\mathbf{P}$ in a 3D space, viewed by a pair of cameras, makes a plane with the left and the right camera centers, i.e., O_l and O_r, respectively. This plane is called the *Epipolar Plane* (π), defined by the *Epipolar Geometry*. Figure 2.12 gives an example of the epipolar geometry. Let $\mathbf{P_l}$ and $\mathbf{P_r}$ be vectors in left and right camera reference frames respectively and let vectors $\mathbf{p_l}$ and $\mathbf{p_r}$ represent the projections of P onto the left and right image planes respectively. The vector $\mathbf{P_l}$ is

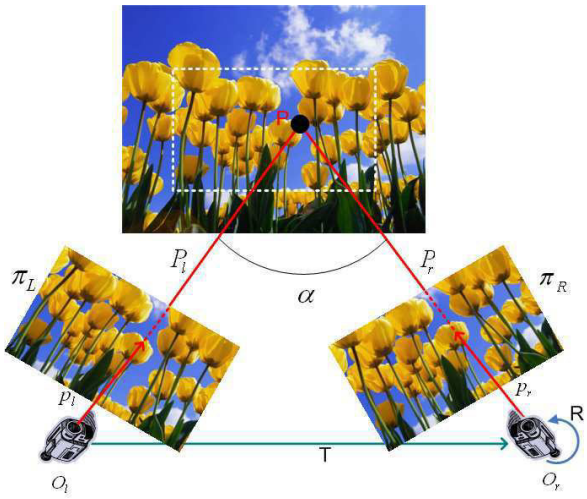


Figure 2.12: *Epipolar Geometry*: A point P in 3D Space as seen from two cameras with centers O_l and O_r. $\mathbf{P_l}$ and $\mathbf{P_r}$ are vectors in the left and right camera reference frames, respectively. The vectors p_l and p_r are the projections of P onto the left and right image planes, respectively.

related to $\mathbf{P_r}$ by the distance between the cameras ($\mathbf{T}$) and the angle α, given as $\mathbf{P_r} = \mathbf{R}(\mathbf{P_l} - \mathbf{T})$, where $\mathbf{R}$ is the rotation matrix defined by angle α. The coplanarity condition between vectors $\mathbf{P_l}, \mathbf{P_r}$ and $\mathbf{T}$ results in the relation $\mathbf{P_r}^{\mathbf{T}}\mathbf{E}\mathbf{P_l} = \mathbf{0}$, where $\mathbf{E} = \mathbf{R}[\mathbf{S}]_\times$ is the *essential matrix* and $[\mathbf{S}]_\times$ is the rank deficient matrix, obtained by factorizing $\mathbf{T} \times \mathbf{P_l}$. The essential matrix encodes information about the epipolar geometry and is defined in camera coordinates. Since we are dealing with image sequences, we need to know the transformation from the camera coordinates to the pixel coordinates. Therefore, we use the fundamental matrix($\mathbf{F}$) that encodes both the extrinsic parameters and the intrinsic parameters, along with the essential matrix. This relation is given as:

$$\mathbf{x}^{\mathbf{T}}\mathbf{F}\mathbf{x}' = \mathbf{0} \tag{2.10}$$

where $\mathbf{x}$ and $\mathbf{x}'$ are the points in left and right image planes, respectively. The projection of a point on the left image lies on a line in the right image defined by Eq. (2.10). This is called the *epipolar*

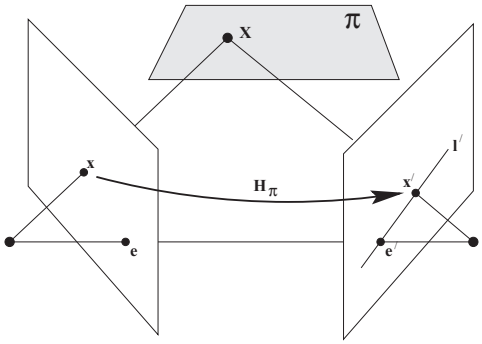


Figure 2.13: A point $\mathbf{x}$ in an image is transferred to a point $\mathbf{x}'$ in another image via the plane $\boldsymbol{\pi}$.

line [FLM92, HZ04].

Equivalently, consider a plane $\boldsymbol{\pi}$ in space not passing through either of the two camera centers (cf. Figure 2.13). The ray passing through the first camera center corresponding to the point $\mathbf{x}$ meets the plane π in a point $\mathbf{X}$. This point $\mathbf{X}$ is then project to a point $\mathbf{x}'$ in the second images. This procedure is known as transfer via the plane $\boldsymbol{\pi}$. The fundamental matrix can then be given as $\mathbf{F} \sim [\mathbf{e}']_\times \mathbf{H}_{\boldsymbol{\pi}}$, where $\mathbf{H}_{\boldsymbol{\pi}}$ is the transfer mapping from one image to another via any plane $\boldsymbol{\pi}$, and $[\mathbf{e}']_\times$ is the image of the camera center of the first camera as seen in the second camera.

A special case arises when the reference transfer plane is the plane at infinity i.e. $\boldsymbol{\pi} \sim \boldsymbol{\Pi}_\infty$. In this case, the transfer mapping i.e. the homography between the two images is given as

$$\mathbf{H}_\infty \sim \mathbf{K}'\mathbf{R}\mathbf{K}^{-1}, \tag{2.11}$$

where $\mathbf{R}$ is the relative rotation between the cameras and $\mathbf{H}^\infty_{\boldsymbol{\Pi}_\infty}$ is called the infinite homography. And the fundamental matrix is then given as [AHR01]:

$$\mathbf{F} \sim [\mathbf{e}']_\times \mathbf{H}_\infty \tag{2.12}$$

2.6.1 Kruppa's Equations

Originally, the auto-calibration method by Faugeras et al. [FLM92] involved the computation of the fundamental matrix, which encodes epipolar geometry between two images [Fau92, LF96]. Each fundamental matrix generates two quadratic constraints involving only the five elements of $\boldsymbol{\omega}^*$, where $\boldsymbol{\omega}^*$ is the dual image of the absolute conic (DIAC). From three views a system of polynomial equations is constructed called Kruppa's equations [Kru13].

Kruppa's equations are based on the relationship between the image of the absolute conic ($\boldsymbol{\omega}$) and the epipolar geometry. If an epipolar line (l) is tangent to $\boldsymbol{w}$, then the corresponding epipolar line (l) is also tangent to $\boldsymbol{\omega}$.

the infinite homography $\mathbf{H}_{\boldsymbol{\Pi}_\infty}^{\infty}$ gives constraints on $\boldsymbol{\omega}^*$ in the form of

$$\boldsymbol{\omega}^* \sim \mathbf{H}_\infty \boldsymbol{\omega}^* \mathbf{H}_\infty^T$$

Using the relation in Equation 2.12 and multiplying $\boldsymbol{\omega}^*$ on left and right by $[\mathrm{e}']_\times$, we obtain:

$$\begin{aligned} [\mathbf{e}']_\times \boldsymbol{\omega}^* [\mathbf{e}']_\times &\sim [\mathrm{e}']_\times \mathbf{H}_\infty \boldsymbol{\omega} * \mathbf{H}_\infty^{\mathbf{T}} [\mathrm{e}']_\times, \\ &\sim \mathbf{F} \boldsymbol{\omega}^* \mathbf{F}^{\mathbf{T}} \end{aligned} \tag{2.13}$$

Thus the fundamental matrix gives constraints on $\boldsymbol{\omega}^*$. However, $\mathbf{F}$ and $\boldsymbol{\omega}^*$ are only defined up to a non-zero scaling and cross multiplying to remove the unknown scale gives quadratic constraints on the elements of $\boldsymbol{\omega}^*$. Each pair of views gives two quadratic equations containing the

elements of $\boldsymbol{\omega}$, and, given three camera displacements (four independent pairs of views), they form an overdetermined set of simultaneous polynomial equations.

CHAPTER 3
DISSECTING THE IMAGE OF ABSOLUTE CONIC

The absolute conic, $\mathbf{\Omega}_\infty$, and its perspective projection $\boldsymbol{\omega}$, known as the Image of the Absolute Conic (IAC), are among the most important concepts in defining the camera geometry. The importance of $\mathbf{\Omega}_\infty$ arises from the fact that it lies on the plane at infinity, $\mathbf{\Pi}_\infty$, and hence is invariant under Euclidean transformations. This implies that the relative position of $\mathbf{\Omega}_\infty$ with respect to a moving camera is fixed. As a result its image, IAC, remains fixed if the camera internal parameters do not vary. Therefore IAC can be used as a calibration object, i.e. for recovering the intrinsic camera parameters. Knowing the IAC, the camera pose, and the Euclidean geometry of the scene [HZ04] can be recovered directly from image measurements up to a similarity.

In this chapter, we revisit the role of IAC in determining the camera geometry, and propose new constraints that are intrinsic to it, reflecting its invariant features. We investigate the application of these new constraints in camera calibration. We show that a more general camera model than the one proposed by [CT90] and formalized in [LZ99] can be recovered from a single view, given an input of three orthogonal vanishing points.

Next, we recall some preliminary notions on the relation between the camera geometry and the IAC. We then dissect the IAC into its constituent components, and provide their geometric meaning and importance. This is followed by an extensive set of experimentations and evaluation of the performance of calibration under noise, and experimental results on real data and comparison with [LZ99].

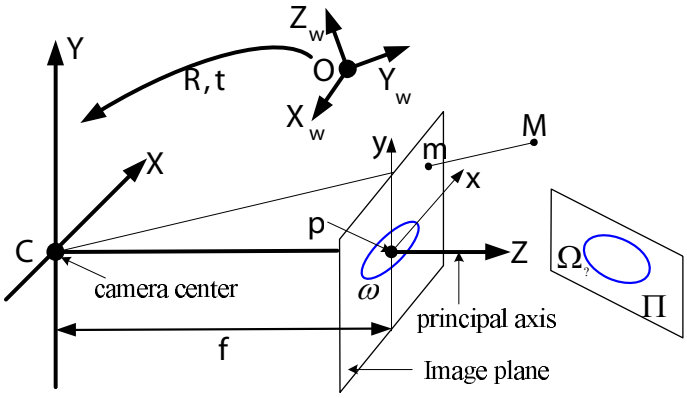


Figure 3.1: The geometry of a pinhole camera. The absolute conic $\mathbf{\Omega}$ is a conic on the plane at infinity that is projected into the image plane as the conic $\boldsymbol{\omega}$, which depends only on the intrinsic parameters of the camera.

3.1 The Role Of IAC

The geometry of imaging the absolute conic in a pinhole camera is shown in Figure 3.1. The general pinhole camera projects a 3D point $\mathbf{M}$ to an image point $\mathbf{m}$ via

$$\mathbf{m} \sim \mathbf{K}\mathbf{R}\left[\mathbf{I}\right| - \mathbf{C}]\,\mathbf{M}, \quad \mathbf{K} = \begin{bmatrix} f & s & u_o \\ 0 & \lambda f & v_o \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.1}$$

where $\sim$ implies equality up to an unknown non-zero scale factor, $\mathbf{R}$ is the rotation matrix from the world coordinate frame to the camera coordinate frame, $\mathbf{C}$ is the inhomogeneous camera projection center, and $\mathbf{K}$ is the camera intrinsic matrix containing the focal length f, the aspect ratio λ, the skew s, and the principal point $\mathbf{p} \sim [\mathbf{u_o}\ \mathbf{v_o}\ \mathbf{1}]^{\mathbf{T}}$.

The role of IAC in defining the camera geometry is better understood by examining the action of a finite camera on points that lie on the plane at infinity $\mathbf{\Pi}_\infty$. A point on $\mathbf{\Pi}_\infty$ can be written as

$\mathbf{M}_\infty \sim [\mathbf{d}\ \mathbf{0}]^\mathbf{T}$, where the 3-vector $\mathbf{d}$ defines the direction of the ray obtained by connecting the image of $\mathbf{M}_\infty$ and the camera projection center. Substituting $\mathbf{M}_\infty$ in (3.1), one can readily verify that $\mathbf{m}_\infty \sim \mathbf{KRM}_\infty$. It therefore follows that the absolute conic, which is the conic $\mathbf{\Omega}_\infty = \mathbf{I}$ on $\mathbf{\Pi}_\infty$ maps to the image conic

$$\boldsymbol{\omega} \sim (\mathbf{KR})^{-T}\mathbf{I}(\mathbf{KR})^{-1} \sim \mathbf{K}^{-T}\mathbf{K}^{-1} \tag{3.2}$$

known as the image of the absolute conic (IAC). Conversely, two image points $\mathbf{m_1}$ and $\mathbf{m_2}$ back-project to two rays with directions $\mathbf{d}_1 = \mathbf{K}^{-1}\mathbf{m_1}$ and $\mathbf{d}_2 = \mathbf{K}^{-1}\mathbf{m_2}$ in the camera coordinate system, where the angle between the two rays is given by the familiar cosine formula

$$\cos\theta = \frac{\mathbf{d}_1^T\mathbf{d}_2}{\sqrt{\mathbf{d}_1^T\mathbf{d}_1}\sqrt{\mathbf{d}_2^T\mathbf{d}_2}} = \frac{\mathbf{m_1^T}\boldsymbol{\omega}\mathbf{m_2}}{\sqrt{\mathbf{m_1^T}\boldsymbol{\omega}\mathbf{m_1}}\sqrt{\mathbf{m_2^T}\boldsymbol{\omega}\mathbf{m}_2}} \tag{3.3}$$

This shows that known angles between vanishing points can be used to impose constraints on the IAC to obtain the camera intrinsic matrix. For instance, given the images of three infinite points $\mathbf{v_i}, \mathbf{i} = \mathbf{1}, ..., \mathbf{3}$ along known directions, and assuming zero skew and unit aspect ratio, one can recover the remaining unknown camera intrinsic parameters. In particular if $\mathbf{v_i}$ are the vanishing points along three orthogonal directions then one can write three linear equations of the form $\mathbf{v_i^T}\boldsymbol{\omega}\mathbf{v_j} = \mathbf{0}, \mathbf{i} \neq \mathbf{j}$ to calibrate the camera [CT90, CDR99, LZ99, Zha00, Stu99, WMC03, CBP05]. This is essentially the core idea behind calibration using the vanishing points, which was formalized by [LZ99]. These works showed that only a simplified camera model with three unknown intrinsic parameters can be recovered from the vanishing points of three orthogonal directions, unless additional information is available (e.g. more images or the circular points [CBP05, LZ99]).

Table 3.1: Scene and internal constraints on IAC.

Condition	Constraint	type	# constraints
Orthogonality	$\mathbf{v_i^T \omega v_j} = \mathbf{0}, \mathbf{i} \neq \mathbf{j}$	linear	1
Pole-polar	$[\mathbf{l}]_\times \boldsymbol{\omega} \mathbf{v} = \mathbf{0}$	linear	2
Homography	$\mathbf{h}_1^T \boldsymbol{\omega} \mathbf{h}_2 = 0$	linear	2
	$\mathbf{h}_1^T \boldsymbol{\omega} \mathbf{h}_1 = \mathbf{h}_2^T \boldsymbol{\omega} \mathbf{h}_2$		
zero skew	$\omega_{12} = \omega_{21} = 0$	linear	1
unit aspect ratio	$\omega_{11} = \omega_{22}$	linear	1

Generally speaking, in recovering the camera geometry from a single view three sources of information have been commonly used in the past to impose constraints on the image of the absolute conic $\boldsymbol{\omega}$:

- metric information about a plane with a known world-to-image homography;
- vanishing points and lines corresponding to known (usually orthogonal) directions and planes;
- *a priori* constraints, such as unit aspect ratio, or zero skew.

These constraints are summarized by Hartley and Zisserman (Table 8.1, page 224 in [HZ04]), which is also reproduced in Table 3.1.

In this section, we re-examine the problem of recovering the camera geometry from a single view, when three vanishing points along world orthogonal directions are known.

3.2 Dissecting IAC

In the existing literature on camera calibration the role of IAC is primarily investigated in terms of its relationship with other geometric entities in the image plane, i.e. the vanishing points and the vanishing line. The relation between IAC and the internal parameters is often limited to equation

(3.2). In this section and the following one, we present some new constraints and their geometric meaning that are more intrinsic to the IAC, i.e. relate to the internal geometry of camera.

Theorem 3.2.1 *(Invariance)*

Let $\boldsymbol{\omega}$ *be the image of the absolute conic. The principal point* $\mathbf{p}$ *satisfies*

$$\boldsymbol{\omega}\mathbf{p} \sim \mathbf{l}_{\infty} \tag{3.4}$$

where $\mathbf{l}_{\infty} \sim [0\ 0\ 1]^T$ *is the canonical position of the line at infinity.*

The proof is straightforward and follows by performing the Choleskey factorization of the Dual Image of the Absolute Conic (DIAC), $\boldsymbol{\omega}^*$, and direct substitution of $\mathbf{p}$.

In the next section, we also provide an alternative proof, which reveals the geometric meaning of the constraint in (3.4).

Proposition 3.2.1 *(Scale)*

Let $\boldsymbol{\omega}$*, denote the image of the absolute conic. We have*

$$|\omega_{33}|\ \mathbf{p^T}\boldsymbol{\omega}\mathbf{p} - \mathtt{det}(\boldsymbol{\omega}) = \mathbf{0} \tag{3.5}$$

where $|\omega_{33}|$ *denotes the minor of IAC corresponding to its last component, and* $\mathtt{det}(\cdot)$ *is the determinant.*

Proposition 3.2.2 *(Ortho-Invariance)*

Let $\mathbf{v_i}, \mathbf{i} = \mathbf{1}, ..., \mathbf{3}$ *denote three vanishing points along mutually orthogonal directions. The image*

of the absolute conic relates these vanishing points via

$$\sum_i \frac{1}{\mathbf{v_i^T \omega v_i}} - \frac{1}{\mathbf{p^T \omega p}} = 0 \tag{3.6}$$

Proofs for all the above results follow by using the Cholesky decomposition of DIAC, $\boldsymbol{\omega}^*$, and direct substitution and algebraic simplification. Note that the result in (3.6) depends on the orthogonality conditions, and hence is dependent on the familiar linear orthogonality constraints $\mathbf{v}_i^T \boldsymbol{\omega} \mathbf{v}_j = 0, i \neq j$. However, (3.4) and (3.5) reflect some intrinsic properties of the IAC and do not depend on the scene geometry or the prior knowledge on the camera intrinsics. This is the key idea presented in this section.

3.2.1 Geometric Interpretation

The result in (3.4) is better understood if we provide its geometric interpretation. Clearly, (3.4) is independent of the image points. Therefore, it reflects some intrinsic property of the IAC. This intrinsic property is better understood if we rewrite (3.4) as the following two independent constraints

$$\mathbf{p^T \omega_1 = 0} \tag{3.7}$$

$$\mathbf{p^T \omega_2 = 0} \tag{3.8}$$

where $\boldsymbol{\omega}_i$ are the rows of the IAC (or equivalently its columns due to symmetry).

This shows that

$$\mathbf{p} \sim \boldsymbol{\omega_1} \times \boldsymbol{\omega_2} \tag{3.9}$$

which is true for a general camera model, i.e. no particular assumptions made about the aspect ratio, or the skew.

A geometric interpretation (see Figure 3.2) of this result is that the two rows $\boldsymbol{\omega}_1$ and $\boldsymbol{\omega}_2$ of the IAC correspond to two lines in the image plane that always intersect at the principal point regardless of the other intrinsic parameters. We may consider three cases: i.e. varying the skew s, the aspect ratio λ, or the focal length f. Although it is highly unlikely for a CCD camera to change its skew or the aspect ratio, it is useful to evaluate these effects on calibrating a general pinhole camera or a simplified one.

Varying the skew s: We may assume that we deal with two identical cameras that differ only in skew: one zero skew and the other non-zero. Let us denote the two corresponding IAC's by

$$\boldsymbol{\omega} \sim \begin{bmatrix} \boldsymbol{\omega}_1 \\ \boldsymbol{\omega}_2 \\ \boldsymbol{\omega}_3 \end{bmatrix} \quad \text{and} \quad \boldsymbol{\omega}_s \sim \begin{bmatrix} \boldsymbol{\omega}'_1 \\ \boldsymbol{\omega}'_2 \\ \boldsymbol{\omega}'_3 \end{bmatrix} \tag{3.10}$$

where $\boldsymbol{\omega}_i$ and $\boldsymbol{\omega}'_i$, $i = 1, ..., 3$ are the rows of the corresponding IAC's.

For the IAC with zero skew, i.e. $\boldsymbol{\omega}$, the two lines $\boldsymbol{\omega}_1$ and $\boldsymbol{\omega}_2$ are parallel to the image x and y axes respectively, and intersect at the principal point. For the general IAC with non-zeo skew,

$\boldsymbol{\omega}'$, the corresponding two lines $\boldsymbol{\omega}'_1$ and $\boldsymbol{\omega}'_2$ are not perpendicular anymore. However, they still intersect at the same image point, i.e. the principal point.

To demonstrate this formally, note that

$$\begin{aligned} \boldsymbol{\omega}' &\sim \mathbf{K}'^{-T}\mathbf{K}'^{-1} \\ &\sim \mathbf{K}'^{-T}\mathbf{K}^{T}\mathbf{K}^{-T}\mathbf{K}^{-1}\mathbf{K}\mathbf{K}'^{-1} \\ &\sim \mathbf{H}_s^{-T}\boldsymbol{\omega}\mathbf{H}_s^{-1} \end{aligned} \tag{3.11}$$

Therefore the transformation that maps the IAC with zero skew to the general IAC is given by the homography

$$\mathbf{H}_s \sim \mathbf{K}'\mathbf{K}^{-1} \tag{3.12}$$

It can be shown that this homography is of the form

$$\mathbf{H}_s \sim \begin{bmatrix} 1 & -\frac{w_{12}}{w_{11}} & \frac{w_{12}}{w_{11}}v_o \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.13}$$

If we now perform the eigen-decomposition of $\mathbf{H}_s$, we will find that this homography has only two distinct eigenvectors both of which correspond to unit eigenvalues. The two eigenvectors are $[0 \ v_o \ 1]$ and $[1 \ 0 \ 0]$. Geometrically, this is equivalent to saying that under the transformation $\mathbf{H}_s$ (i.e. if the skew of a camera changes from zero to a non-zero value), the point $[0 \ v_o \ 1]$ and

the vanishing point along the x-axis remain invariant. In other words, these are geometrically fixed points under $\mathbf{H}_s$. Since any linear combination of these two points is also an eigenvector, we deduce that the principal point $\mathbf{p}$, which lies on the line joining the two fixed points is also invariant under this transformation. This shows that equation (3.4) conveys an invariant property of the IAC, i.e. upon changing the skew the principal point should still lie on the intersection of the image lines defined by the first two rows of the IAC.

Another illuminating feature of $\mathbf{H}_s$ is that if we do the eigendecomposition of the transposed homography $\mathbf{H}_s^T$, we will find that there are also only two distinct eigenvectors, i.e. $[0\ 1\ -v_o]^T$ and $[0\ 0\ 1]^T$. Geometrically, this implies that the line $[0\ 1\ -v_o]^T$ and the line at infinity are invariant under changes in the skew. Since the principal point lies on the first line, it again confirms that the principal point is a fixed point under variations in the skew.

Varying aspect ratio λ: Interestingly enough, the same process as above can be used to establish that upon changing the aspect ratio λ, the principal point is also an invariant fixed point on the intersection of the two image lines defined by the first two rows of the IAC. Again, if the two IAC's are denoted by $\boldsymbol{\omega}$ and $\boldsymbol{\omega}'$, then their relationship is defined by a homography of the form

$$\mathbf{H}_\lambda \sim \mathbf{K}'\mathbf{K}^{-1} \tag{3.14}$$

$$\sim \begin{bmatrix} 1 & 0 & 0 \\ 0 & \lambda & v_o(1-\lambda) \\ 0 & 0 & 1 \end{bmatrix} \tag{3.15}$$

where $\lambda = \sqrt{\frac{\omega_{11}^2}{\omega_{11}\omega_{22}-\omega_{12}^2}}$.

Similar eigen-analysis reveals that $\mathbf{H}_\lambda$ shares the same two eigenvectors $[0\ v_o\ 1]$ and $[1\ 0\ 0]$ corresponding to its repeated unit eigenvalue, and a third eigenvector that corresponds to the point at infinity along the y-axis, i.e. $[0\ 1\ 0]$ with the eigenvalue equal to λ. This shows that the same two points are again geometric fixed points. However this time the infinite point along the y-axis is also fixed. Again using the fact that the linear combinations of eigenvectors corresponding to unit eigenvalues is also an eigenvector, we conclude that the principal point, which lies on the line joining the first two eigenvectors, is also geometrically a fixed point under variations of λ.

Varying the focal length f: Finally, if we let the focal length of a camera vary then the homography that relates the two IAC's is given by

$$\mathbf{H}_f \sim \mathbf{K}'\mathbf{K}^{-1} \tag{3.16}$$

$$\sim \begin{bmatrix} r & (r-1)\frac{\omega_{12}}{\omega_{11}} & (1-r)\left(u_o + v_o\frac{\omega_{12}}{\omega_{11}}\right) \\ 0 & r & (1-r)v_o \\ 0 & 0 & 1 \end{bmatrix} \tag{3.17}$$

where r is the ratio of two focal lengths.

The eigen decomposition of this homography indicates that the principal point is the eigenvector corresponding to the unit eigenvalue, and hence is a fixed point under $\mathbf{H}_f$. The last two eigenvectors, are repeated and correspond again to the point at infinity along the x-axis $[1\ 0\ 0]$, with the eigenvalue equal to r.

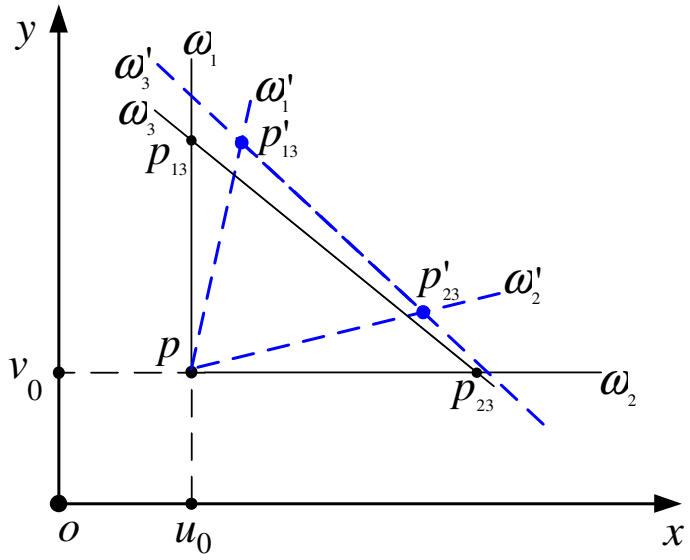


Figure 3.2: The geometry associated with the IAC: ω_1, ω_2, and ω_3 represent the lines associated with the IAC when the skew is zero, and ω'_1, ω'_2, and ω'_3 illustrate the case when the skew is not zero. In both cases the principal point is on the intersection of the first two lines, providing two linear constraints on the IAC. The ratio of line segments along the two lines (two rows) are preserved as the skew changes.

Remark If two cameras differ only by the intrinsic parameters s, λ, or f, then the corresponding IAC's, ω and ω', satisfy

$$\omega_1 \times \omega_2 \sim \omega'_1 \times \omega'_2 \tag{3.18}$$

Figure 3.2 illustrates this underlying geometry of IAC for the case of varying skew. As can be seen in Figure 3.2 the third row of IAC also corresponds to a line in the image plane which intersects the first two lines at two distinct points other than the principal point. These intersection points together with other points along the two lines ω_1 and ω_2 can be used to confirm that the ratio of line segments remain invariant, since all the homographies described above are affine. Unfortunately, the third row of IAC or the resulting invariant ratios do not provide new independent constraints.

Before we close this section, we also formalize the familiar constraint that the principal point is known *a priori* to be close to the center of the image $\mathbf{c}$, as the following "soft constraint"

$$\hat{\mathbf{p}} = \arg\min(\mathbf{p} - \mathbf{c})^{\mathbf{T}}(\mathbf{p} - \mathbf{c}) \tag{3.19}$$

This latter constraint is a very practical prior in self-calibration.

3.3 Single-View Calibration

The results of the previous section are both good news and bad news. The bad news is that we can not find more than two intrinsic constraints on the IAC from its internal geometry. The good news is that the two constraints that we find can be used to reprameterize the IAC. This is rather very useful, since it allows us to recover a more general camera model than the existing single-view calibration techniques such as [LZ99]: e.g. recover f, s and (u_o, v_o) with three vanishing points, or recover f and (u_o, v_o) with two vanishing points.

For instance, let us assume that the camera skew is zero. The IAC is then of the form

$$\boldsymbol{\omega} \sim \begin{bmatrix} 1 & 0 & \omega_{13} \\ 0 & \omega_{22} & \omega_{23} \\ \omega_{13} & \omega_{23} & \omega_{33} \end{bmatrix} \tag{3.20}$$

Given three orthogonal vanishing points, we can formulate the single-view calibration problem as

the solution to the following set of five equations:

$$\mathbf{v_i^T \omega v_j = 0}, \quad \mathbf{i \neq j}, \quad \mathbf{i, j = 1, ..., 3} \tag{3.21}$$

$$\mathbf{p^T \omega_1 = 0}, \tag{3.22}$$

$$\mathbf{p^T \omega_2 = 0} \tag{3.23}$$

These equations are linear in terms of the components of $\boldsymbol{\omega}$, and hence any four of them can be used to reparameterize $\boldsymbol{\omega}$ in (3.20) in terms of only the principal point $\mathbf{p}$. Suppose we use the first four equations for reparameterization then the resulting $\boldsymbol{\omega}$, which depends only on $\mathbf{p}$ should minimize

$$\hat{\mathbf{p}} = \arg\min \mathbf{p^T W p} \quad \text{where} \quad \mathbf{W = \omega_2 \omega_2^T} \tag{3.24}$$

We initialize $\hat{\mathbf{p}}$ at the center of the image, and minimize using a standard optimization method (e.g. Levenberg-Marquardt) in a window around the center of the image. Once the principal point is obtained, all components of the IAC can be recovered (since they are expressed in terms of $\mathbf{p}$), and hence the camera intrinsic matrix $\mathbf{K}$ can be computed by Choleskey decomposition. Note that the method recovers a more general camera model of four unknown parameters, e.g. f, λ and (u_o, v_o).

The three columns of the rotation matrix are then given by $\mathbf{r}_i = \pm \frac{\mathbf{K^{-1} v_i}}{\|\mathbf{K^{-1} v_i}\|}$ - the sign ambiguity can be removed using the cheirality constraint [HZ04]. The translation of the camera can also be recovered up to an unknown global scale, taking an image point as the projection of the world origin.

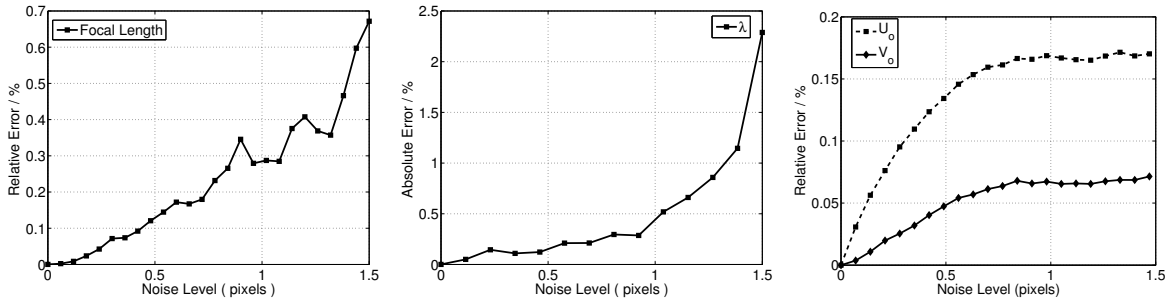
3.4 Results And Noise Resilience

In this section, we show an extensive set of experimental results on both synthetic and real data using the method described above. We have performed detailed experimentation on the effect of noise in the estimation error over 1000 independent trials. The simulated camera has a focal length of $f = 2000$, the aspect ratio $\lambda = \frac{1015}{2000}$, zero skew, and the principal point at $(510, 385)$, for image size of 1024×768.

Performance Versus Noise Level: In this experimentation, we compared estimated camera intrinsic and extrinsic parameters against the ground truth, while adding a zero-mean Gaussian noise varying from 0.1 pixels to 1.5 pixels. The results show the average performance over 1000 independent trials. Figure 3.3 summarizes the results for intrinsic parameters. For noise level of 1.5 pixels, which is larger than the typical noise in practical calibration [Zha00], the relative error for the focal length f is 0.7%. The maximum relative error for the aspect ratio is less than 0.01%, while that of the principal point is less than 0.2%. Excellent performance is also achieved for all extrinsic parameters as shown in Figure 3.4, i.e. less than 0.4% error for both t_x and t_y relative to f, and absolute errors of less than a tenth of a degree for all rotation angles.

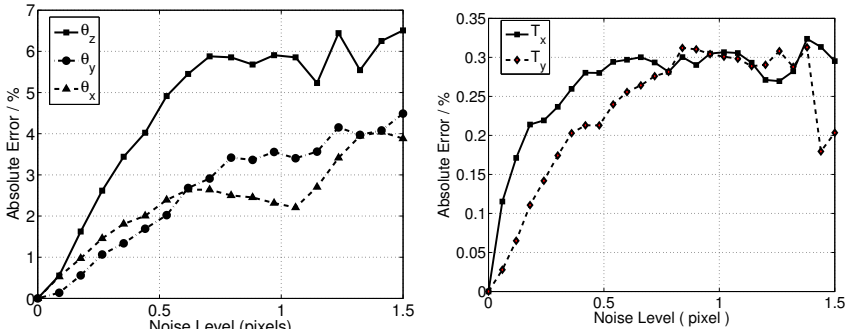
Performance against [LZ99]: We performed the comparison using the same setup as above. Figure 3.5 summarizes our results.

Performance on Real Data: For real data, in order to evaluate our results, we used an approach similar to [Zha00] using the uncertainty associated with the estimated intrinsic parameters characterized by their standard deviation over many images. Figure 3.6 shows two examples from the set of real images that were used in this experimentation. Results are summarized in table 3.2. The



(a) (b) (c)

Figure 3.3: Performance vs noise (in pixels) averaged over 1000 independent trials: (a) relative error for the focal length f, (b) the relative error for the aspect ratio λ, and (c) the relative in the coordinates of the principal point.



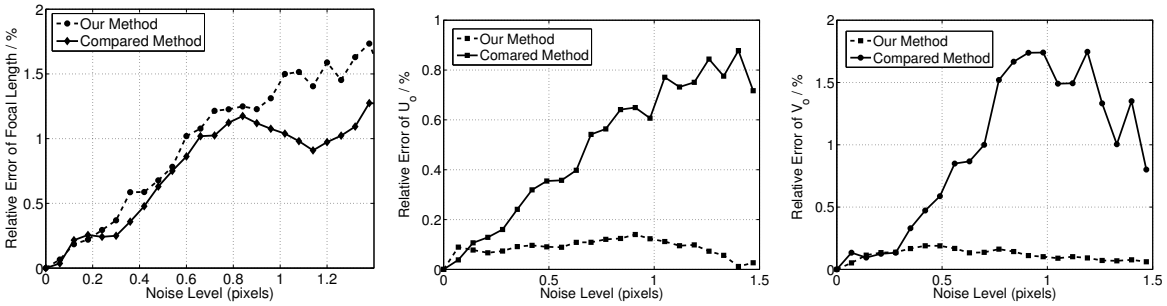
(a) (b)

Figure 3.4: Performance vs noise (in pixels) averaged over 1000 independent trials: (a) absolute error for the rotation angles, (b) absolute error for the translations along x and y axes.

Table 3.2: Uncertainty in experimental results with real data.

Parameter	Mean	Std.
f	460.52	5.74
λ	1.51	0.24
u_o	318.33	5
v_o	242.77	4.41

uncertainty is reasonable, but could be improved of course if we use more accurate approaches [MK95, LZ98, Shu99, VMP04] to finding the vanishing points, rather than using an unreliable manual point clicking.



(a) (b) (c)

Figure 3.5: Performance vs [LZ99] averaged over 1000 independent trials: (a) relative error for the focal length f, (b) & (c) the relative error in the coordinates of the principal point.

Figure 3.6: Two of many images used in evaluation with real data

Table 3.3: Intrinsic constraints of IAC. The first two are related to the invariant properties of the principal point, the third constraint cross-correlates this property and the orthogonality constraint (ortho-invariance), and the last one is a "soft constraint" on the position of the principal point in the image plane.

Condition	Constraint	Linear
Invariance	$\boldsymbol{\omega}\mathbf{p} \sim \mathbf{l}_\infty$	yes
Scale	$\lvert\omega_{33}\rvert \ \mathbf{p}^\mathbf{T}\boldsymbol{\omega}\mathbf{p} - \mathtt{det}(\boldsymbol{\omega}) = \mathbf{0}$	no
Ortho-invariance	$\sum_i \frac{1}{\mathbf{v_i^T}\boldsymbol{\omega}\mathbf{v_i}} - \frac{1}{\mathbf{p^T}\boldsymbol{\omega}\mathbf{p}} = 0$	no
"Soft"	$\mathbf{p} \sim \arg\min(\mathbf{p}-\mathbf{c})^\mathbf{T}(\mathbf{p}-\mathbf{c})$	no

3.5 Conclusion

In this chapter, we presented new constraints that are intrinsic to the image of the absolute conic. The constraints reflect the invariant properties of the IAC, and characterize its geometric structure. In particular, we showed that the rows of the IAC correspond to very specific image lines whose intersections bear the invariant properties of the IAC. An immediate application of this geometric characterization of the IAC is that it can extend our ability to estimate more complete set of camera parameters from a single view. We therefore propose the following table as an addendum to table given by Hartley and Zisserman (Table 8.1, page 224 in [HZ04]). Unfortunately, however, as described in the text, not all the constraints can be used independently. As a result, we believe that it is unlikely that one can recover all the five intrinsic parameters of the camera from a single view of three orthogonal vanishing points, unless some additional information is available.

CHAPTER 4
CAMERA CALIBRATION USING SHADOW PATHS

In this chapter, our main goal is to demonstrate that a camera can be calibrated by using the shadow trajectory of an object. An object casts its shadow on the ground plane. When observed over a period of time, this shadow forms a curve or a trajectory, which we refer to as a shadow trajectory. We require at least two shadow trajectories, i.e. at least a pair of objects. We require at least five points on this shadow trajectory to perform camera calibration. More object and more trajectory points can be used for a more robust solution. By fitting conics to these shadow trajectories, we are able to obtain the vanishing line of the ground plane.

The most related work is that of [CF06]. Cao and Foroosh [CF06] use multiple views of the objects. This limits the applicability of their method as having more than one camera is not always possible. Moreover, they require an object's bottom and top location to be always visible in the images, a condition which we have successfully relaxed in our proposed method. Compared to other methods on camera calibration from shadow trajectories, the proposed method is more robust and more precise, as it involves using multiple conics for the estimation of unknown camera calibration matrix [Hei00].

The main step of our approach is a novel method to extract the vanishing line of the ground plane from using only the shadow trajectories (Section 4.2). This step requires at least five images (> 5 for a robust solution) containing shadow trajectories of at least a pair of objects. The vanishing line along with an extracted vertical vanishing point is used to estimate camera parameters (Section 4.4). Accordingly, this chapter is divided into corresponding sections addressing each issue.

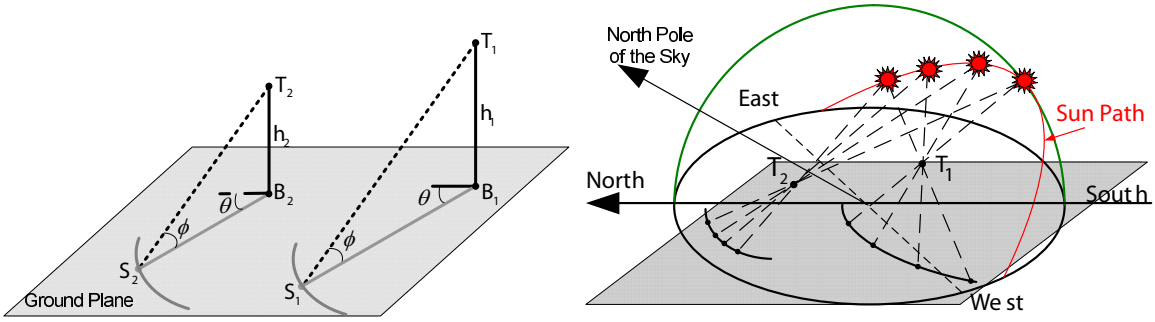


Figure 4.1: Two objects T_1 and T_2 casting shadows on the ground plane. The locus of shadow positions over the course of a day is a function of the sun altitude ϕ, the sun azimuth θ and the height h_i of the object.

4.1 The Setup

Let $\mathbf{T}$ be a 3D stationary point and $\mathbf{B}$ its footprint (i.e. its orthogonal projection) on the ground plane. As depicted in Fig. 4.1, the locus of shadow positions $\mathbf{S}$ cast by $\mathbf{T}$ on the ground plane is a smooth curve that depends only on the altitude and the azimuth angles of the sun in the sky and the vertical distance h of the object from its footprint. This geometric configuration is rather interesting, since the object point $\mathbf{T}$ together with the ground plane act as an artificial pinhole camera, where the camera projection center is the object point, the image plane is the ground plane, the focal length is the vertical distance h, and the principal point is the footprint $\mathbf{B}$.

Without loss of generality, we take the ground plane as the world plane $z = 0$, and define the x-axis of the world coordinate frame toward the true north point, where the azimuth angle is zero. Therefore, algebraically, the 3D coordinates of the shadow position can be unambiguously specified by their 2D coordinates in the ground plane as

$$\bar{\mathbf{S}}_\mathbf{i} = \bar{\mathbf{B}}_\mathbf{i} + \mathbf{h}_\mathbf{i} \cot\phi \begin{bmatrix} \cos\theta \\ \\ \sin\theta \end{bmatrix}, \tag{4.1}$$

where $\bar{\mathbf{S}}_\mathbf{i} = [\mathbf{S_{ix}} \;\; \mathbf{S_{iy}}]^\mathbf{T}$ and $\bar{\mathbf{B}}_\mathbf{i} = [\mathbf{B_{ix}} \;\; \mathbf{B_{iy}}]^\mathbf{T}$ are the inhomogeneous coordinates of the shadow position $\mathbf{S_i}$, and the object's footprint $\mathbf{B_i}$ on the ground plane, ϕ is sun altitude, and θ the sun azimuth. Equation (4.1) is based on the assumption that the sun is distant and therefore its rays, e.g. $\mathbf{T_i S_i}$, are parallel to each other. It follows that the shadows $\mathbf{S_1}$ and $\mathbf{S_2}$ of any two stationary points $\mathbf{T_1}$ and $\mathbf{T_2}$ are related by a rotation-free 2D similarity transformation as $\mathbf{S_2} \sim \mathbf{H_s^{12} S_1}$, where

$$\mathbf{H_s^{12}} \sim \begin{bmatrix} h_2/h_1 & 0 & B_{2x} - B_{1x}h_2/h_1 \\ 0 & h_2/h_1 & B_{2y} - B_{1y}h_2/h_1 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.2}$$

Note that the above relationship is for world shadow positions and valid for any day time.

4.2 Recovering The Vanishing Line

The goal in the calibration step in this chapter is to recover the vanishing line of the ground plane from the shadow trajectories. Once the vanishing line (l_∞) is recovered, it is used together with the vertical vanishing point, found by fitting lines to vertical directions, to recover the image of the absolute conic (IAC). There are two cases that need to be considered:

4.2.1 When Shadow Casting Object Is Visible

This case requires that the bottom point, and optionally the top point, of the shadow casting object be visible in the image. An example of this case is the light pole visible in image sequence shown in Figure 4.9. Figure 4.2 illustrates the general setup for this case. The vertical vanishing point is obtained by $\mathbf{v}_z = (\mathbf{T_1} \times \mathbf{B_1}) \times (\mathbf{T_2} \times \mathbf{B_2})$

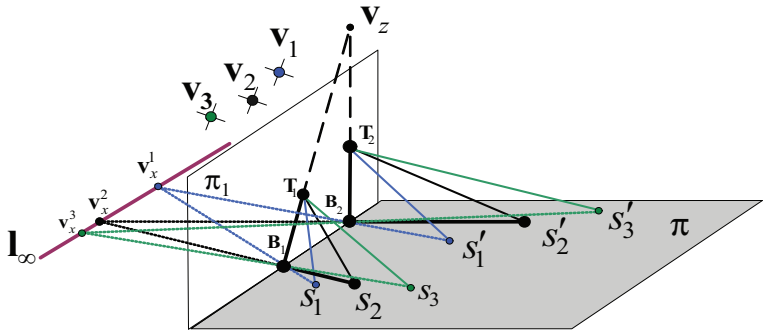


Figure 4.2: The setup used when the bottom and the top locations of the object are visible.

The estimation of $\mathbf{l}_\infty$ is as follows: at time instance $t = 1$, the sun located at vanishing point $\mathbf{v_1}$ casts shadow of $\mathbf{T_1}$ and $\mathbf{T_2}$ at points $\mathbf{S_1}$ and $\mathbf{S'_1}$, respectively. The sun is a distant object and therefore its rays, $\mathbf{T_1S_1}$ and $\mathbf{T_2S'_1}$, are parallel to each other. It then follows that the shadow rays, i.e. $\mathbf{S_1B_1}$ and $\mathbf{S'_1B_2}$, are also parallel to each other. These rays intersect at the vanishing point $\mathbf{v}_x^1$ on the ground plane. Similarly, for time instance $t = 2$ and $t = 3$, we obtain the vanishing points $\mathbf{v}_x^2$ and $\mathbf{v}_x^3$, respectively. These vanishing points all lie on the vanishing line of the ground plane on which the shadows are cast, i.e. $\mathbf{v}_x^{i\,\mathbf{T}}\mathbf{l}_\infty = 0$, where $i = 1, 2, \ldots n$ and n is number of instances for which shadow is being observed. Thus a minimum of two observations are required of at least two vertical objects to obtain $\mathbf{l}_\infty$.

4.2.2 When Shadow Casting Object Is NOT Visible

This is a more *general* case. The bottom point and/or the top point of the shadow casting object might not always be visible in a video sequence. Figure 4.3 shows a picture of downtown Washington D.C. One of the shadow casting object is the traffic light (marked with a blue dot) hanging by a horizontal pole (or a cable). This traffic light does not have a bottom point on the ground plane. In this setup, $\mathbf{l}_\infty$ can not be recovered as described above. Also, the vertical vanishing point

Figure 4.3: Few of the images in one of our data set that were taken from one of the live webcams in Washington D.C. The objects that cast shadows on the ground are highlighted. Shadows move to the left of the images as time progresses.

is now obtained by other vertical structures in the scene, not necessarily shadow-casting structures. Therefore, in order to recover $\mathbf{l}_\infty$, we have to only work with the shadow trajectories.

Given any five imaged shadow positions of the same 3D point, cast at distinct times during one day, one can fit a conic through them, which would meet the line at infinity at two points, which may be real or imaginary depending on whether the resulting conic is an ellipse, a parabola, or a hyperbola [HZ04]. Suppose now we have two world points $\mathbf{T_1}$ and $\mathbf{T_2}$ that cast shadows on the ground plane. Any five distinct shadow positions of $\mathbf{T_1}$ and $\mathbf{T_2}$ define two distinct and unique conics on the ground plane, which after camera projection yield the image conics $\mathbf{C_1}$ and $\mathbf{C_2}$, respectively. These two conics are related by $\mathbf{C_2} \sim (\mathbf{H}\mathbf{H}_\mathbf{s}^{\mathbf{12}}\mathbf{H}^{-\mathbf{1}})^{-\mathbf{T}}\mathbf{C_1}(\mathbf{H}\mathbf{H}_\mathbf{s}^{\mathbf{12}}\mathbf{H}^{-\mathbf{1}})^{-\mathbf{1}}$, where $\mathbf{H}$ is the world to image planar homography with respect to the ground plane. Since the two world conics are similar, owing to the distance of the sun from the observed objects, these two conics generally intersect at four points, two of which must lie on the image of the horizon line of the ground plane.

4.2.3 Computing Intersections

The basic idea of conic intersection is illustrated in Fig. 4.4. We now present the method for computing these intersections and expand on its relation to the recovery of the vanishing line $\mathbf{l}_\infty$.

All conics passing through the four points of intersection can be written as

$$\mathbf{C}_\mu \sim \mathbf{C_1} + \mu\mathbf{C_2}. \tag{4.3}$$

Equation (4.3) defines a pencil of conics parameterized by μ, where all the conics in the pencil intersect at the same four points $\mathbf{m_i}, \mathbf{i} = \mathbf{1}, ..., \mathbf{4}$. Four such points such that no three of them are collinear also give rise to what is known as the *complete quadrangle*.

It can be shown that in this pencil at most three conics are not full rank. For this purpose note that any such degenerate conic should satisfy

$$\det(\mathbf{C}_\mu) = \det(\mathbf{C_1} + \mu\mathbf{C_2}) = \mathbf{0}. \tag{4.4}$$

It can then be readily verified that (4.4) is a cubic equation in terms of μ. Therefore upon solving (4.4), we obtain at most three distinct values $\mu_i, i = 1, ..., 3$, which provide the three corresponding degenerate conics

$$\mathbf{C}_{\mu_\mathbf{i}} \sim \mathbf{C_1} + \mu_\mathbf{i}\mathbf{C_2}, \quad \mathbf{i} = \mathbf{1}, ..., \mathbf{3}. \tag{4.5}$$

In the general case (i.e. when the three parameters $\mu_i, i = 1, ..., 3$ are distinct), the three degenerate conics are of rank 2, and therefore can be written as

$$\mathbf{C}_{\mu_\mathbf{i}} \sim \mathbf{l_i l_i'^T} + \mathbf{l_i' l_i^T}, \quad \mathbf{i} = \mathbf{1}, ..., \mathbf{3}, \tag{4.6}$$

where $\mathbf{l_i}$ and $\mathbf{l_i'}$ are three pairs of lines as shown in Fig.4.4.

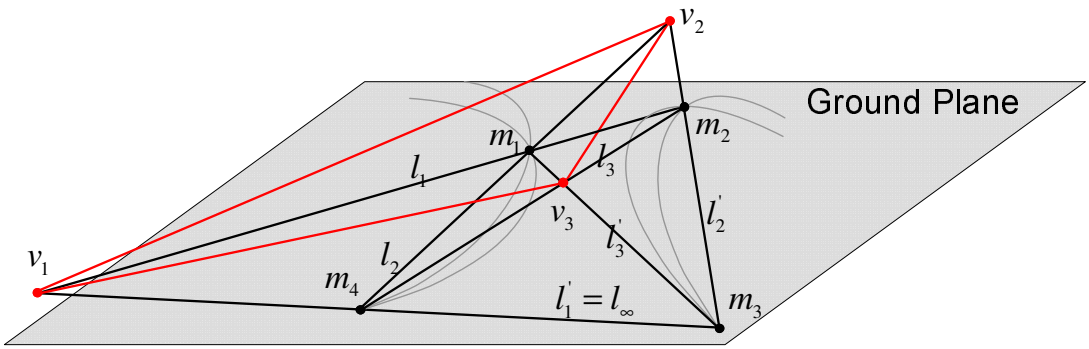


Figure 4.4: The two gray conics are fitted by two sets of five distinct shadow positions on the ground plane cast by two world points. Generally, the two conics intersect at four points $\mathbf{m_i}, \mathbf{i} = \mathbf{1}, ..., \mathbf{4}$ two of which must lie on the line at infinity. The four points form a quadrangle inscribed to any one of the gray conics. The diagonal triangle $\Delta\mathbf{v_1v_2v_3}$ is self-polar [SK79].

Now, let $\mathbf{C}^*_{\mu_\mathbf{i}}$ be the adjoint matrix of $\mathbf{C}_{\mu_i}$. It then follows from (4.6) that

$$\mathbf{C}^*_{\mu_\mathbf{i}}\mathbf{l_i} = \mathbf{C}^*_{\mu_\mathbf{i}}\mathbf{l'_i} = \mathbf{0}, \quad \mathbf{i} = \mathbf{1}, ..., \mathbf{3}, \tag{4.7}$$

which yields (by using the property that the cofactor matrix is related to the way matrices distribute with respect to the cross product [HZ04])

$$\mathbf{C}^*_{\mu_\mathbf{i}}\mathbf{l_i} \times \mathbf{C}^*_{\mu_\mathbf{i}}\mathbf{l'_i} = \mathbf{C}_{\mu_\mathbf{i}}(\mathbf{l_i} \times \mathbf{l'_i}) = \mathbf{0}, \quad \mathbf{i} = \mathbf{1}, ..., \mathbf{3}. \tag{4.8}$$

In other words, the intersection point $\mathbf{v_i}$ of the pair of lines, $\mathbf{l_i}$ and $\mathbf{l'_i}$, is given by the right null space of $\mathbf{C}_{\mu_i}$. Therefore, in practice, it can be found as the eigenvector corresponding to the smallest eigenvalue of the degenerate conic $\mathbf{C}_{\mu_i}$. The triangle formed by the three vertices $\mathbf{v_1v_2}$ and $\mathbf{v_3}$ is known as the *diagonal triangle* of the quadrangle [SK79].

Theorem 4.2.1 *(Self-Polar Triangle)*

Let $\mathbf{m_1}, \mathbf{m_2}, \mathbf{m_3}$ *and* $\mathbf{m_4}$ *be four points on the conic locus* $\mathbf{C}_\mu$*, the diagonal triangle of the quadrangle* $\mathbf{m_1 m_2 m_3 m_4}$ *is self-polar w.r.t.* $\mathbf{C}_\mu$*. Since two of the points lie on* $\mathbf{l}_\infty$*, one of the vertices of* $\Delta \mathbf{v_1 v_2 v_3}$ *also lies on* $\mathbf{l}_\infty$*.*

This theorem follows directly from the projective geometry and we omit the proof here. Thus the triangle $\Delta \mathbf{v_1 v_2 v_3}$ is the diagonal triangle of the quadrangle composed of points $\mathbf{m_i}, \mathbf{i} = \mathbf{1}, ..., \mathbf{4}$ inscribed in a conic. There also exists a harmonic relationship between any two sides of the quadrangle and $\mathbf{v_i}$ of $\Delta \mathbf{v_1 v_2 v_3}$ that meets that side. Exploring this harmonic relationship for obtaining further constraints is the topic of our future research.

Next, we verify that for any conic $\mathbf{C}_\mu$ in the pencil

$$(\mathbf{l_i} \times \mathbf{l'_i})^\mathbf{T} \mathbf{C}_\mu (\mathbf{l_j} \times \mathbf{l'_j}) = \mathbf{0}, \quad \mathbf{i} \neq \mathbf{j}, \quad \mathbf{i}, \mathbf{j} = \mathbf{1}, ..., \mathbf{3} \tag{4.9}$$

This means that any pair of right null vectors of the degenerate conics $\mathbf{C}_{\mu_i}, \mathbf{i} = \mathbf{1}, ..., \mathbf{3}$ are conjugate with respect to all conics in the pencil. In other words, their intersections form the vertices of a self-polar triangle with respect to all the conics in the pencil.

To obtain the intersection points of the two shadow conics, we use the fact that all the conics in the pencil intersect at the same four points. Therefore, the intersection points can also be found as the intersection of the lines $\mathbf{l_i}$ and $\mathbf{l'_i}$ with the lines $\mathbf{l_j}$ and $\mathbf{l'_j}$ ($i \neq j$). The lines $\mathbf{l_i}$ and $\mathbf{l'_i}$ can be simply found by solving

$$\mathbf{C}_{\mu_i} \sim \mathbf{l_i} \mathbf{l'}_\mathbf{i}^\mathbf{T} + \mathbf{l'_i} \mathbf{l}_\mathbf{i}^\mathbf{T} \tag{4.10}$$

Equation (4.10) provides 4 constraints on $\mathbf{l_i}$ and $\mathbf{l'_i}$ (5 due to symmetry minus 1 for rank deficiency). In practice it leads to two quadratic equations on the four parameters of the two lines, which can

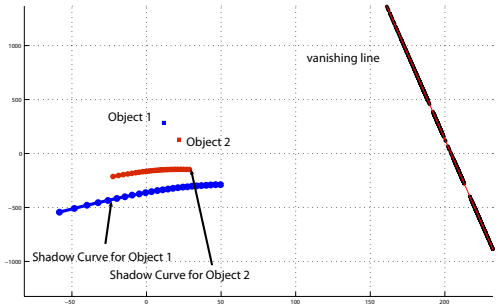


Figure 4.5: The horizon line detected from a sequence of self-polar triangles and the intersection of the conics fit on shadow trajectories of two objects.

be readily solved. The solution, of course, has a twofold ambiguity due to the quadratic orders, which is readily resolved by the fact that

$$\mathbf{l_i} \times \mathbf{l_i'} \sim \text{null}(\mathbf{C}_{\mu_i}) \tag{4.11}$$

The process can be repeated for $\mathbf{l_j}$ and $\mathbf{l_j'}$, and the intersections of the lines between the two sets would then provide the four intersection points of the shadow conics.

4.3 Robust estimation of $\mathbf{l}_\infty$

The shadow cast on the ground plane might not be very accurately localized. This is due to the nature of the problem, mainly because of the irregularities of the road, for example, or the shadow not being very sharp due to a cloudy weather. Therefore some scheme needs to be adopted to minimize the influence of outliers and noise on *true* data points so that accurate results may be obtained.

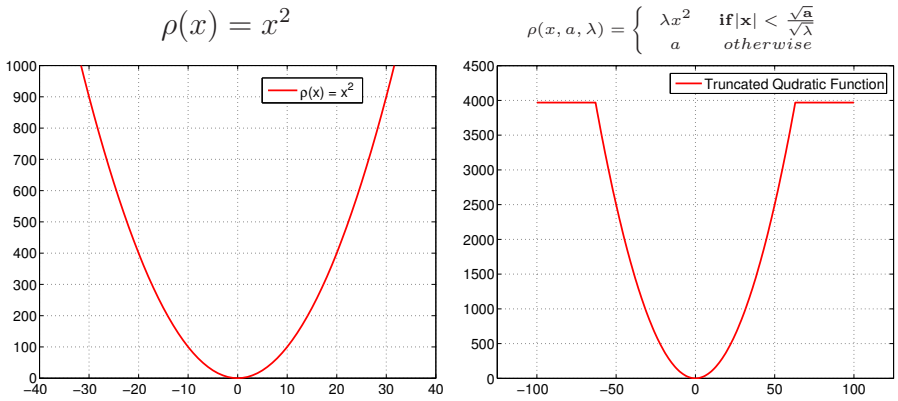
In our case, since two of the intersection points of the shadow conics are at infinity (without

loss of generality $\mathbf{l}'_1$ as shown in Fig.4.4), one of the vertices, $\mathbf{v_1}$, of the self-polar triangle must be a vanishing point, and thus also lies on the horizon line, $\mathbf{l}_\infty$, of the ground plane. Therefore given six or more corresponding image points on the shadow paths of the two objects, we can get six or more self-polar triangles, from which the horizon line of the ground plane can be recovered. Since, two of intersection points (2 points of the quadrangle) are also on the horizon line of the ground plane, they can be used together with one vertex of each self-polar triangle to recover the horizon line. As an example, Figure 4.5 illustrates the horizon line fitted to many points obtained through synthetic experiment, to be described shortly. Therefore, the system of overdetermined set of equations needed to solve for $\mathbf{l}_\infty$ can be given as:

$$\Phi^T \mathbf{l}_\infty = 0 \tag{4.12}$$

where Φ is a matrix containing the estimated vanishing points. Note that for $n \geq 6$ corresponding points on shadow paths of two objects, we obtain a total of $\frac{3n!}{(n-5)!5!}$ vanishing points. For instance, with only 10 corresponding shadow points, we would get 756 points on the horizon line. This would allow us to very accurately estimate the horizon line in the presence of noise. Φ is therefore a $\frac{3n!}{(n-5)!5!} \times 3$ matrix and we have to *robustly* estimate $\mathbf{l}_\infty$.

The main goals of robust statistics is to recover the best structure that fits the majority of the model while rejecting the outliers. We need to recover the best $\mathbf{l}_\infty$ such that $\mathbf{K}$ is closest to the actual calibration matrix. The popular standard least squares (LS) estimation, which minimizes the Euclidean norm of the residuals, is extremely sensitive to outliers i.e. it has a breakdown point of zero. Total Least Squares (TLS) method, on the other hand, minimizes the Frobenius norm.



(a) Quadratic ($L2$ Norm) (b) Truncated Quadratic

Figure 4.6: Two commonly used minimization cost functions.

Given an over-determined system of equations, TLS problem is to find the smallest perturbation to the data and the observation matrix to make the system of equations compatible. A suitable function also needs to be selected that is less forgiving to outliers, one such example is the *truncated quadratic* [BA96], commonly used in computer vision (cf. 4.6). The errors are weighted up to a fixed threshold, but beyond that, errors receive constant penalty. Thus the influence of outliers goes to zero beyond the threshold.

In order to remove the outlier influence, we use the truncated Rayleigh quotient. The quotients are estimated as:

$$\rho(\mathrm{l}_\infty) = \sum^{N} \frac{\mathbf{x^T A x}}{\mathbf{x^T x}} < \xi \tag{4.13}$$

where $\mathbf{x}$ represent the three parameters of l_∞, $A = \begin{bmatrix} v_x^i & v_y^i & 1 \end{bmatrix}^{\mathbf{T}} \begin{bmatrix} v_x^i & v_y^i & 1 \end{bmatrix}$ contains the determined vanishing points, and ξ is the threshold. The Rayleigh quotients are estimated from the observation points and the residual errors are estimated. The threshold ξ is set to the median of all the residual errors. Observation points obtained from Eq. 4.12 having residual errors greater than ξ are removed as outliers. After outlier removal, the *outlier-free* remaining observation points $\mathbf{Q}$

are used to construct the over-determined system of Eqs. (4.12). The system is then solved using the Singular Value Decomposition (SVD). The correct solution is the eigenvector corresponding to the smallest eigenvalue.

In summary, in order to minimize the influence of noise on our observation matrix $\mathbf{Q}$, we apply the Rayleigh quotient to *filter* out the noisy data points. Once the outliers are removed, the Total Least Squares method is applied to the remaining observation points to estimate the unknown parameter w_{11} of the IAC.

4.4 Camera Calibration

The computed horizon line $\mathbf{l}_\infty$, together with the vertical vanishing point $\mathbf{v}_z$, fitted from vertical objects, provide two constraints on the image of the absolute conic in the form of the pole-polar relationship $\mathbf{l}_\infty \sim \omega \mathbf{v_z}$ [HZ04]. Assuming a camera with zero skew, and unit aspect ratio, the IAC would be of the form

$$\boldsymbol{\omega} \sim [\omega_{\mathbf{1}}\ \omega_{\mathbf{2}}\ \omega_{\mathbf{3}}] \sim \begin{bmatrix} 1 & 0 & \omega_{13} \\ 0 & 1 & \omega_{23} \\ \omega_{13} & \omega_{23} & \omega_{33} \end{bmatrix} \tag{4.14}$$

In the existing literature on camera calibration the role of IAC is primarily investigated in terms of its relationship with other geometric entities in the image plane, i.e. the vanishing points and the vanishing line. The relation between IAC and the internal parameters is often limited to equation $\boldsymbol{\omega} \sim \mathbf{K}^{-T}\mathbf{K}^{-1}$. In a relation that is more intrinsic to the IAC. Geometric interpretation for this relation allows us to gain more insight into widely used the "closeness-to-the-center" constraint [CS05, HZ04].

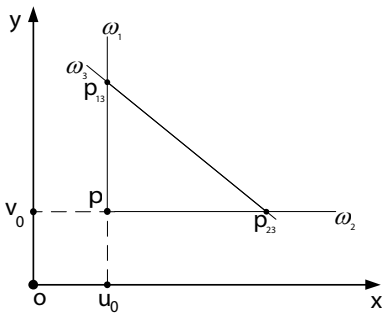


Figure 4.7: The geometry associated with the IAC: $\boldsymbol{\omega}_1$, $\boldsymbol{\omega}_2$, and $\boldsymbol{\omega}_3$ represent the lines associated with the IAC when the skew is zero. The principal point is located at the intersection of the first two lines, providing two linear constraints on the IAC.

4.4.1 Geometric Interpretation

The result in Theorem (3.2.1) is better understood if we provide its geometric interpretation. This intrinsic property of IAC is better understood if we rewrite (3.4) as:

$$\mathbf{p}^{\mathbf{T}}\boldsymbol{\omega_1} = \mathbf{0} \tag{4.15}$$

$$\mathbf{p}^{\mathbf{T}}\boldsymbol{\omega_2} = \mathbf{0} \tag{4.16}$$

from which, we get

$$\mathbf{p} \sim \boldsymbol{\omega_1} \times \boldsymbol{\omega_2} \tag{4.17}$$

which is true for a general camera model, i.e. no particular assumptions made about the aspect ratio, or the skew.

A geometric interpretation (see Figure 4.7) of this result is that the two rows $\boldsymbol{\omega}_1$ and $\boldsymbol{\omega}_2$ of

the IAC correspond to two lines in the image plane that always intersect at the principal point regardless of the other intrinsic parameters.

Using the two constraints provided by the pole-polar relationship, we express the IAC in terms of only one of its parameters, e.g. ω_{33}, and solve for it by enforcing the constraint that the principal point is close to the center of the image by minimizing

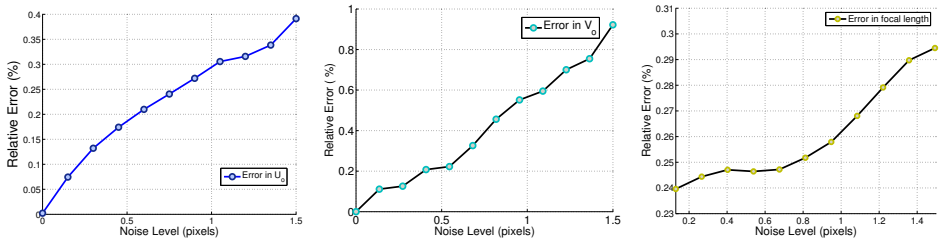
$$\hat{\omega}_{33} = \arg\min \|\omega_{\mathbf{1}} \times \omega_{\mathbf{2}} - \mathbf{c}\| \tag{4.18}$$

where $\mathbf{c}$ is the center of the image, and $\hat{\omega}_{33}$ is the optimal solution for ω_{33}, from which the other two parameters are computed to completely recover the IAC in (4.14). It must be noted that the pole-polar relationship could also be used on its own to recover a more simplified IAC without using the minimization in (4.18). Note also that the proposed auto-calibration method is independent of any scene structure [LZ99, Tri98, Zha00], or (special) camera motions [Har97, HA97, PKG99]. We only require the vertical vanishing point and that the shadow be cast on a plane without requiring any further information.

4.5 Experimental Results

We rigorously tested and validated our method on synthetic as well as real data sequences for self-calibration steps. Results are described below.

Synthetic Data: Two vertical objects of different heights were randomly placed on the ground plane. Using the online available version of SunAngle Software [Gro], we generated altitude and azimuth angles for the sun corresponding to our own geo-location with latitude 28.51°. The



(a) (b) (c)

Figure 4.8: Performance averaged over 1000 independent trials: (a) & (b) relative error in the coordinates of the principal point (u_o, v_o), (c) the relative error in the focal length f.

vertical objects and the shadow points were projected by a synthetic camera with a focal length of $f = 1000$, the principal point at $(u_o, v_o) = (320, 240)$, unit aspect ratio, and zero skew.

In order to test resilience of the proposed self-calibration method to noise, we gradually added Gaussian noise of zero mean and standard deviation of up to 1.5 pixels to the projected points. The estimated parameters were then compared with the ground truth values mentioned above. For each noise level, we performed 1000 independent trials. The final averaged results for calibration parameters are shown in Figure 4.8. Note that, as explained in [Tri98], the relative difference with respect to the focal length is a more geometrically meaningful error measure. Therefore, relative error of f, u_o and v_o were measured w.r.t f while varying the noise from 0.1 to 1.5 pixels. As shown in the figure, errors increase almost linearly with the increase of noise in the projected points. For the noise of 1.5 pixels, the error is found to be less than 0.3% for f, less than 0.5% for u_o and less than 1% for v_o.

Real Data: Several experiments on two separate data sets are reported below for demonstrating the proposed method. In the first set, 11 images were captured live from downtown Washington D.C. area, using one of the webcams available online at `http://trafficland.com/`. As shown in Figure 4.9, a lamp post and a traffic light were used as two objects casting shadows

Figure 4.9: Few of the images taken from one of the live webcams in downtown Washington D.C. The two objects that cast shadows on the ground are shown in red and blue, respectively. Shadows move to the left of the images as time progresses.

on the road. The shadow points are highlighted by colored circles in the figure. The calibration parameters were estimated as

$$\mathbf{K} = \begin{bmatrix} 700.357 & 0 & 172 \\ 0 & 700.357 & 124 \\ 0 & 0 & 1 \end{bmatrix}$$

4.6 Discussion And Conclusion

The auto-calibration step requires only the shadow trajectories of two objects on the ground plane to be visible in the images, along with the vertical vanishing point. Unlike shadow-based calibration methods such as [AB04, CF06], this step does not require the objects themselves to be seen in the images.

It is, however, important that the shadow trajectories can be used to fit conics. An exception, which leads to a degenerate case, happens twice a year during equinox, when the lengths of the day and the night are equal. As a result, it can be shown that, the shadow trajectories degenerate to straight lines. Two cases may occur: if the two objects casting shadows are not aligned along the east-west direction, then their shadow trajectories will be two distinct straight lines that are

parallel in the world. Therefore, their intersection would provide only a single point at infinity, which is insufficient to determine the horizon line; if the two objects are aligned along the east-west direction, then the shadow lines will coincide and no vanishing point can be found. In both cases auto-calibration cannot be performed using our method. However, this degenerate case is rather rare and happens only twice a year.

CHAPTER 5
CAMERA CALIBRATION FROM PEDESTRIANS

Observation of human activities from stationary cameras is of significant interest to many applications. This is mainly due to the fact that the computer vision research has advanced to systems that can accurately detect, recognize and track objects as they move through a scene. Most of the video surveillance involves, for instance, monitoring an area of interest (e.g. a building entrance, or an embassy) using stationary cameras where the intent is to monitor as large an area as possible. The goal for such a system can be to model the behavior of objects (e.g. cars or pedestrians, depending on the situation). Typically, one can employ path modeling techniques or activity learning techniques for single or multiple cameras (e.g. [GSR98]) and even establish relations between the camera system [MT04], as discussed in more detail later. It is known that due to perspective projection the measurements made from the images do not represent metric data. Thus the obtained object trajectories and consequently the associated probabilities represent projectively distorted data, unless we have a calibrated camera. This is evident from a simple observation: the objects grow larger and move faster as they approach the camera center, or two objects moving in parallel direction seem to converge at a point in the image. The projective camera thus makes it difficult to characterize objects - in terms of their sizes, motion characteristics, length ratios and so on - unless more information is available about the camera being used. This is where the camera calibration steps in.

This chapter proposes a robust auto-calibration method to estimate camera intrinsics and extrinsics by observing pedestrians in a scene. Many camera calibration techniques exits for different scenarios [HZ04] but we limit ourselves with related work on camera auto-calibration from observ-

ing pedestrians.

Lv et al. [LZN02] were the first to propose calibration by recovering the horizon line and the vanishing points from observed walking humans. However, their formulation does not handle robustness issues. Recently Krahnstoever and Mendonça [KM05] proposed a Bayesian approach for auto-calibration by observing pedestrians. Foot-to-head homology is decomposed to extract the vanishing point and the horizon line for calibration. They also incorporate measurement uncertainties and outlier models. However, their method requires prior knowledge about some unknown calibration parameters and prior knowledge about the location of people; and their algorithm is also non-linear. We also handle a more general scenario where the pedestrian does not need to walk on a straight line.

We propose a robust linear solution to estimate camera intrinsic and extrinsic parameters by observing pedestrians. See Fig. 5.1 for an example of the scenario. The detected head and feet locations of a person, over at least two instances, are used to estimate two harmonic homologies: head-to-foot and frame-to-frame. The former is referred to as the vertical homology, vertical vanishing points being the vertex. The later is referred to as the horizontal homology as the vertex lies on the horizon line. Linear constraints on the unknown camera parameters are obtained by using properties of these homologies. The noise in the data points is minimized by using total least squares method to solve an over-determined system of equations, where the outliers are removed by truncating the Rayleigh quotient [GL89].

We next discuss the method in detail and provide results for both synthetic and real data.

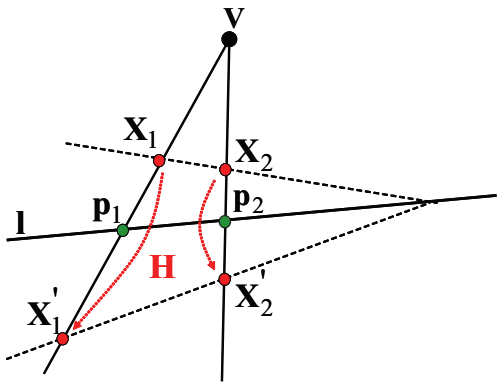


Figure 5.1: A homology defined by an axis $\mathbf{l}$ and a vertex $\mathbf{v}$. See text for more details.

5.1 Harmonic Homologies From Pedestrians

Our auto-calibration method, to be described shortly, is based on using a pair of homologies defined by a walking pedestrian. A plane projective transformation $\mathbf{H}$ is a homology if it has a line of fixed points (called the *axis*), and a fixed point not on the axis (called the *vertex*) [HZ04]. A homology $\mathbf{H}$ is completely specified by its axis $\mathbf{l}$, its vertex $\mathbf{v}$, and its characteristic invariant μ [HZ04, SK79], and is given by:

$$\mathbf{H} = \mathbb{I} - (\mu - \mathbf{1})\frac{\mathbf{v}\mathbf{l}^{\mathbf{T}}}{\mathbf{v}^{\mathbf{T}}\mathbf{l}} \tag{5.1}$$

This is depicted schematically in Fig. 5.1. Under the homology $\mathbf{H}$, the axis is mapped to itself. Each point $\mathbf{x}_i$ off the axis lies on a fixed line through the vertex $\mathbf{v}$, intersecting the axis at a point $\mathbf{p}_i$, and is mapped to another point $\mathbf{x}'_i$ on the line. As a result, the corresponding points $\mathbf{x}_i \longleftrightarrow \mathbf{x}'_i$, the vertex $\mathbf{v}$ and the intersection of their joint with the axis at $\mathbf{p}_i$ are collinear. The cross ratio given by these four collinear points defines the characteristic invariant μ of the homology (see [HZ04], Fig. A7.2, page 630).

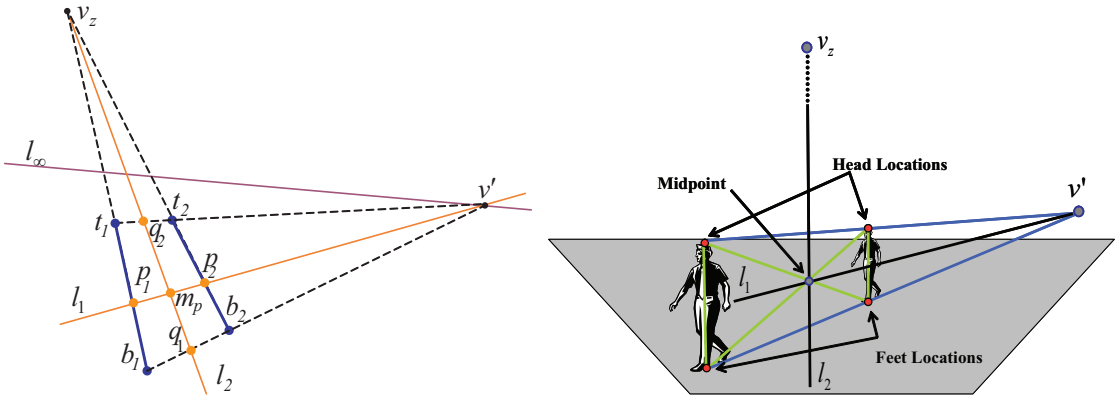


Figure 5.2: **Harmonic Homologies**: Tracking pedestrians over any two frames provides two harmonic homologies. See text for more details.

As an object or a pedestrian of height h traverses the ground plane, the line joining the top and bottom points (i.e. head and feet for pedestrian) at different time instances can be intersected to obtain the vertical vanishing point $\mathbf{v_z}$ (see Figure 5.2b), since the pedestrians can be viewed as vertical objects on a ground plane. Similarly, since the height of a pedestrian does not change (we ignore the case when a pedestrian might sit or jump), the line joining the head locations at two instances and similarly for the feet locations, intersect at a common point $\mathbf{v}'$ lying on the line at infinity $\mathbf{l}_\infty$ (see Figure 5.2b). For a simple case of two frames, the head to foot correspondence can be mapped by a homology. We refer to this homology as the *vertical* homology, since the vertical vanishing point $\mathbf{v_z}$ is the vertex of the homology:

$$\mathbf{H_v} = \mathbb{I} - (\mu_\mathbf{v} - \mathbf{1})\frac{\mathbf{v_z}\mathbf{l_1^T}}{\mathbf{v_z^T}\mathbf{l_1}} \tag{5.2}$$

where $\mathbf{v_z}$ and $\mathbf{l_1}$ are, respectively, the vertex and the axis of the homology. Therefore, $\mathbf{H_v}$ maps head locations to feet locations about the axis $\mathbf{l_1}$.

Another important geometric relation, so far ignored in existing literature on camera calibration from pedestrians, is the homology existing between different locations of a pedestrian. As shown in Fig. 5.2(b), since the height of a pedestrian is the same in all the frames, the line joining the head locations ($\mathbf{t_1}$ and $\mathbf{t_2}$) intersects the line joining the feet locations ($\mathbf{b_1}$ and $\mathbf{b_2}$) at a point $\mathbf{v}'$ on the line at infinity ($\mathbf{l}_\infty$), forming another homology, which we refer to as the *horizontal* homology:

$$\mathbf{H_h} = \mathbb{I} - (\mu_\mathbf{h} - \mathbf{1})\frac{\mathbf{v}'\mathbf{l_2^T}}{\mathbf{v}'^\mathbf{T}\mathbf{l_2}} \tag{5.3}$$

where $\mathbf{l_2}$ and $\mathbf{v}'$ are as depicted in Fig. 5.2(b) and $\mu_\mathbf{h}$ is the invariant of the homology.

In general, a homology has five degrees of freedom [HZ04], i.e. two for the axis, two for the vertex, and one for the characteristic invariant. Therefore, three point correspondences are sufficient to uniquely determine the homology. A special case occurs when $\mu = -1$, in which case the homology is said to be *harmonic* [SK79]. A simple inspection of the scenario at hand reveals that the above two homologies defined by a walking pedestrian are indeed harmonic. To demonstrate this note that in our homologies the vertex is always a vanishing point (i.e. the image of a point at infinity), and the intersection of the joint of corresponding points with the axis is always the imaged midpoint of the two corresponding points. As a result, the cross ratio for the vertical homology is given by

$$\mu_\mathbf{v} = \text{Cross}(\mathbf{v_z}, \mathbf{t_1}, \mathbf{p_1}, \mathbf{b_1}) = (\frac{\overline{\mathbf{v_z}\mathbf{b_1}}}{\overline{\mathbf{v_z}\mathbf{t_1}}}) \div (\frac{\overline{\mathbf{p_1}\mathbf{b_1}}}{\overline{\mathbf{p_1}\mathbf{t_1}}}) = \mathbf{-1} \tag{5.4}$$

This last result follows immediately from the fact that the cross ratio is a projective invariant, and that its value in the 3-D space is -1. Similarly, $\mu_\mathbf{h} = \text{Cross}(\mathbf{v}', \mathbf{t_2}, \mathbf{q_2}, \mathbf{t_1}) = \mathbf{-1}$ for $\mathbf{H_h}$. Hence,

only two point correspondences are sufficient to determine the head to foot mapping, completely. Moreover, knowing $\text{Cross}(\mathbf{v_z}, \mathbf{t_1}, \mathbf{p_1}, \mathbf{b_1}) = \text{Cross}(\mathbf{v}', \mathbf{t_2}, \mathbf{q_2}, \mathbf{t_1}) = -\mathbf{1}$ can be used to constrain the head-foot location in presence of noise, as shall be discussed shortly. The method in [KM05] employs only the vertical homology (not *two harmonic homologies*) and therefore requires more than two point correspondences to solve the problem.

This result is also closely related to the configuration resulting from the perspective image of an object with a bilateral symmetry, where corresponding points are related by a harmonic homology about the imaged axis of symmetry [CBP05, WMC03, CF04a]. To demonstrate this note that any two instances of a walking pedestrian form a rectangle in the world (connect the four red dots in Figure 5.2(b)). Since the intersection of lines is preserved under perspective projection, the intersection of the two diagonals is the center of this rectangle $\mathbf{m_p}$. For our case of vertical homology, and equivalently for the horizontal homology, $\mathbf{m_p}$ along with $\mathbf{q_1}, \mathbf{q_2}$ and $\mathbf{v_z}$ are *harmonic* i.e. there exists a representation in which the four points have parameters $0, -1, 1$ and ∞, respectively [SK79, pg.48]. Thus in such case the cross-ratio of the four points $\mu_{\mathbf{v}}$ (or $\mu_{\mathbf{h}}$ for the horizontal case), referred to as the harmonic cross-ratio, is equal to -1. The imaged mid-point $\mathbf{m_p}$ is given by, $\mathbf{m_p} = (\mathbf{b_1} \times \mathbf{t_2}) \times (\mathbf{b_2} \times \mathbf{t_1})$. As shown in Fig. 5.2, $\mathbf{t_1}, \mathbf{b_1}$ correspond to $\mathbf{t_2}, \mathbf{b_2}$, respectively to construct the harmonic homology ($\mathbf{H_h}$). Similarly, $\mathbf{t_1}, \mathbf{t_2}$ respectively correspond to $\mathbf{b_1}, \mathbf{b_2}$ to determine $\mathbf{H_v}$.

Initial homology estimation: $\mathbf{H_h}$ and $\mathbf{H_v}$ are estimated from the detected head/foot location of an observed pedestrian. To estimate $\mathbf{H_v}$, $\mathbf{v_z} = (\mathbf{b_1} \times \mathbf{t_1}) \times (\mathbf{b_2} \times \mathbf{t_2})$. The axis of vertical homology is obtained as $\mathbf{l_1} = \mathbf{p_1} \times \mathbf{p_2}$, where $\mathbf{p_1} = (\mathbf{m_p} \times \mathbf{v}') \times (\mathbf{b_1} \times \mathbf{t_1})$ and $\mathbf{p_2} = (\mathbf{m_p} \times \mathbf{v}') \times (\mathbf{b_2} \times \mathbf{t_2})$. $\mathbf{H_h}$ is obtained in a similar manner.

(a) (b) (c)

Figure 5.3: (a) shows an instance of a video sequences where a pedestrians is moving in the scene. (b) and (c) represent the detected pedestrian in two different frames. The head and foot location are denoted by $\mathbf{t_i}$ and $\mathbf{b_i}$. See text for more details.

Determining head/foot locations The proposed method requires point correspondences, which are head/foot positions of the pedestrians. Moving foreground objects (or region of interest), with shadows removed, can be extracted and tracked fairly accurately with statistical background models [GSR98, JS02, SM98]. Lv et al. [LZN02] perform eigendecomposition of the detected blob to extract head/feet location. An example of a detected pedestrian is shown in Fig. 5.3.

A simpler approach can be adopted to extract the head and foot location [KM05]. As shown in Fig. 5.3, these locations can be easily estimated by calculating the center of mass and the second order moment of the lower and the upper portion of the bounding box of the foreground region (cf. Fig. 5.3(b)(c)).

5.2 Robust Auto-Calibration

The main issue with camera auto-calibration by observing pedestrians is that head/feet detection is noisy. For example, a pedestrian may walk casually so that the posture might not be straight. Violations such as these result in measurements that can be viewed as *outliers*. Thus, some scheme needs to be adopted to minimize the influence of these outliers and noise on *true* data points so that accurate results may be obtained. An elegant way of doing this would be to enforce the constraint

that the noise-free homologies must be harmonic, i.e. $\mu_{\mathbf{h}} = \mu_{\mathbf{v}} = -1$.

For this purpose, we express the vanishing points in terms of the IAC, as follows:

$$\widehat{\mathbf{v}_\mathbf{z}} \sim \mathbf{l_2} \times \mathbf{l}_{\perp\mathbf{xy}} \sim \mathbf{l_2} \times \boldsymbol{\omega}\mathbf{v}' \tag{5.5}$$

$$\widehat{\mathbf{v}'} \sim \mathbf{l_1} \times \mathbf{l}_{\infty} \sim \mathbf{l_1} \times \boldsymbol{\omega}\mathbf{v_z} \tag{5.6}$$

where $\mathbf{l}_{\perp\mathbf{xy}}$ is any line orthogonal to the xy-plane given by the pole-polar relationship $\mathbf{l}_{\perp\mathbf{xy}} = \boldsymbol{\omega}\mathbf{v}'$ [HZ04].

Therefore, the harmonic cross ratios can be expressed now in terms of the IAC:

$$\text{Cross}(\widehat{\mathbf{v}_\mathbf{z}}, \mathbf{t_1}, \mathbf{p_1}, \mathbf{b_1}) + \mathbf{1} = (\frac{\overline{\widehat{\mathbf{v}_\mathbf{z}}\mathbf{b_1}}}{\overline{\widehat{\mathbf{v}_\mathbf{z}}\mathbf{t_1}}}) \div (\frac{\overline{\mathbf{p_1}\mathbf{b_1}}}{\overline{\mathbf{p_1}\mathbf{t_1}}}) + \mathbf{1} = \mathbf{0} \tag{5.7}$$

$$\text{Cross}(\widehat{\mathbf{v}'}, \mathbf{t_2}, \mathbf{q_2}, \mathbf{t_1}) + \mathbf{1} = (\frac{\overline{\widehat{\mathbf{v}'}\mathbf{t_1}}}{\overline{\widehat{\mathbf{v}'}\mathbf{t_2}}}) \div (\frac{\overline{\mathbf{q_2}\mathbf{t_1}}}{\overline{\mathbf{q_2}\mathbf{t_2}}}) + \mathbf{1} = \mathbf{0} \tag{5.8}$$

Unfortunately, Eqs. (5.7) and (5.8) are not independent. Hence, we have only one constraint on $\boldsymbol{\omega}$. Unless we have more information, we can only solve for one unknown in $\boldsymbol{\omega} = \text{diag}(\omega_{11}, \omega_{11}, 1)$. Fortunately, these two equations can be simplified into linear equations of the form: $a_i^j w_{11} + b_i^j = 0$, where the subscript i indicates the frame number and the superscript $j = \{1, 2\}$ indicates the two equations obtained per image pair. Thus from each pair of images we obtain two equations with one unknown. Consequently, as each combination provides two equations, for n frames, $2 \times \begin{pmatrix} n \\ 2 \end{pmatrix}$ such combinations are possible. Equations obtained from a sequence are used to construct an over-determined system of equations:

$$\underbrace{\begin{bmatrix} a_1^1 & b_1^1 \\ a_1^2 & b_1^2 \\ \vdots & \vdots \\ a_n^1 & b_n^1 \\ a_n^2 & b_n^2 \end{bmatrix}}_{\mathbf{Q}} \begin{bmatrix} w_{11} \\ 1 \end{bmatrix} = 0 \tag{5.9}$$

The main goal of robust statistics is to recover the best structure that fits the majority of the model while rejecting the outliers. Thus, we need to recover the best w_{11} such that $\mathbf{K}$ is closest to the actual calibration matrix. The popular standard least squares (LS) estimation is extremely sensitive to outliers i.e. it has a breakdown point of zero. Therefore, Total Least Squares (TLS) method is adopted to solve the system of Eqs (5.9). Given an over-determined system of equations, TLS problem is to find the smallest perturbation to the data and the observation matrix to make the system of equations compatible. A suitable function also needs to be selected that is less forgiving to outliers, one such example is the *truncated quadratic* [BA96], commonly used in computer vision. The errors are weighted up to a fixed threshold, but beyond that, errors receive constant penalty. Thus the influence of outliers goes to zero beyond the threshold.

We use the truncated Rayleigh quotient to remove outlier influence. The quotients are estimated as:

$$\rho(w_{11}) = \sum^{n} \frac{\mathbf{x^T A x}}{\mathbf{x^T x}} < \xi \tag{5.10}$$

where $\mathbf{x} = \begin{bmatrix} w_{11} \\ 1 \end{bmatrix}$, $A = \begin{bmatrix} a_i^j & b_i^j \end{bmatrix}^{\mathbf{T}} \begin{bmatrix} a_i^j & b_i^j \end{bmatrix}$ and ξ is the threshold. The Rayleigh quotients are estimated from the observation points and the residual errors are estimated. The threshold ξ is set to the median of all the residual errors. Observation points obtained from Eq. (5.9) having

residual errors greater than ξ are removed as outliers. After outlier removal, the *outlier-free* remaining observation points $\mathbf{Q}$ are used to construct the over-determined system of Eqs. (5.9). The system is then solved using the Singular Value Decomposition (SVD). The correct solution is the eigenvector corresponding to the smallest eigenvalue.

In summary, in order to minimize the influence of noise on our observation matrix $\mathbf{Q}$, we apply the Rayleigh quotient to *filter* out the noisy data points. Once the outliers are removed, the Total Least Squares method is applied to the remaining observation points to estimate the unknown parameter w_{11} of the IAC.

5.2.1 Estimating More Parameters

As described above, we are able to determine only the focal length f (by estimating w_{11}), along with extrinsic parameters. The proposed method considers a very general case - making no assumptions about pedestrian movements. However, if more camera parameters are to be obtained, some additional constraints need to be considered. Lv et al. [LZN02] assume a pedestrian walking in different directions for some duration. Thus more than one vanishing point of the ground plane are obtained, which enables them to calculate $\mathbf{l}_\infty$. Knowing $\mathbf{l}_\infty$ provides additional constraints on $\boldsymbol{\omega}$:

$$\mathbf{l}_\infty \sim \boldsymbol{\omega}\mathbf{v}_\mathbf{z} \tag{5.11}$$

Generally this relation provides two linear constraints on $\boldsymbol{\omega}$ but in our case it is dependent on Eqs. (5.7),(5.8). Moreover, if these different direction of pedestrian movements are mutually orthogonal, the third vanishing points can be obtained - enabling us to obtain a total of 3 camera

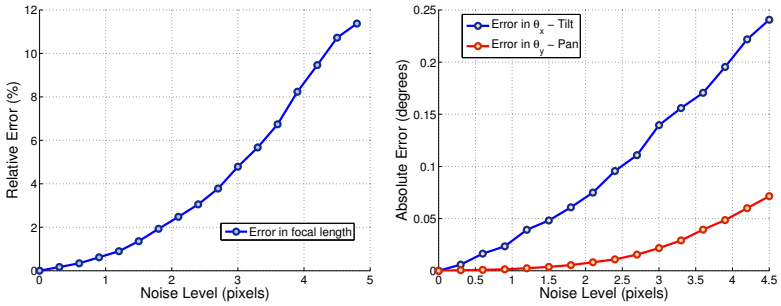
parameters [CDR99].

5.3 Results

The proposed system has been tested on multiple sequences with a variety of motion trajectories. The sequences have a resolution of 320×240 pixels and captured at multiple locations and each location contained multiple paths of travel. Three test sequences were used for evaluation purposes, named **Seq** $\#1$, **Seq** $\#2$, and **Seq** $\#3$. Our tracker is able to accurately establish correspondences over a variety of environmental conditions. Results on synthetic and real data are presented below.

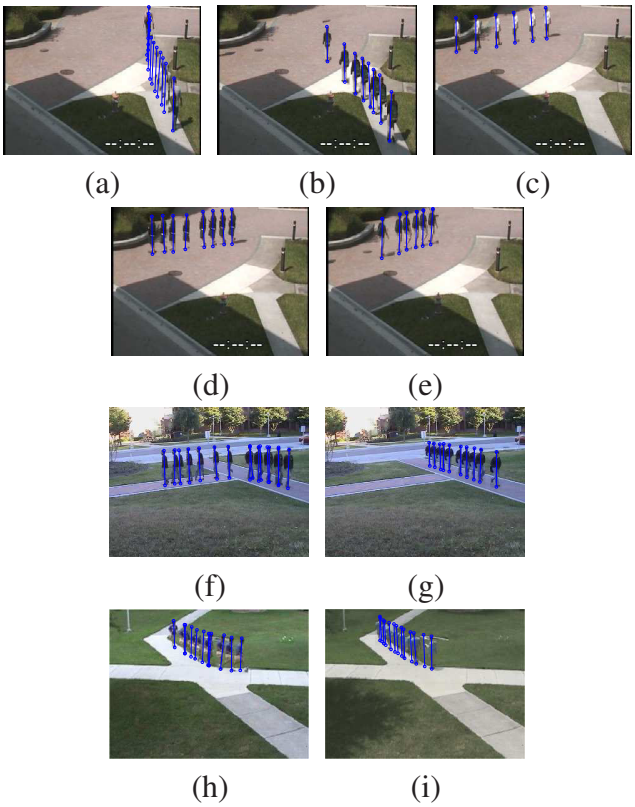
Synthetic data: We rigourously test the proposed method for estimating the camera parameter i.e. f. Nine vertical lines of same height but random location are generated to represent a pedestrian in our synthetic data. The ends of the lines indicate the head or the foot locations. We gradually add a Gaussian noise with $\mu = 0$ and $\sigma \leq 5$ pixels to the data-points making up the vertical lines. Taking two vertical lines at a time, the four points i.e. two head and two foot location are used to obtain $\mathbf{H_h}$ and $\mathbf{H_v}$. Vanishing points derived in Eqs. (5.5),(5.6) are substituted into Eqs. (5.8), (5.7) to construct the over-determined system of equations, as described in Section 5.2. While varying the noise from 0.1 to 5 pixel level, we perform 1000 independent trials for each noise level, the results are shown in Fig. 5.4. The relative error in f increases almost linearly with respect to the noise level. For a maximum noise of 5 pixels, we found that the error was under 12%. The absolute error in the estimated rotation angles, i.e. pan θ_y and tilt θ_x, also increase linearly and is well under 1° degree.

Real Data: The proposed system has been tested on multiple sequences. The image sequences have a resolution of 320×240 pixels and captured at multiple locations. Different pedestrians from



(a) (b)

Figure 5.4: Performance of auto-calibration method VS. Noise level in pixels.



(a) (b) (c)

(d) (e)

(f) (g)

(h) (i)

Figure 5.5: The figure depicts instances of the data sets used for testing the proposed auto-calibration method. The estimated head and foot locations are marked with circle. Different frames are super-imposed on the background image to better visualize the test data.

a single sequences are used to obtain the camera parameters. As reported by [Zha00], the mean of the estimated focal length is taken as the ground truth and the standard deviation as a measure of

Table 5.1: The recovered focal length for (*starting from the left column, going clock wise direction*) **Seq** $\#1$, **Seq** $\#2$ and **Seq** $\#3$. Obtained results are compared to the method proposed in [LZ99].

Seq $\#1$	Recovered Focal Length (f)
Fig. 5.5a	$f = 2362.48$
Fig. 5.5b	$f = 2341.72$
Fig. 5.5c	$f = 2287.68$
Fig. 5.5d	$f = 2295.54$
Fig. 5.5e	$f = 2252.24$
from [LZ99]	$f = 2248.56$

Seq $\#2$	Recovered Focal Length (f)
Fig. 5.5f	$f = 2046.06$
Fig. 5.5g	$f = 1905.12$
from [LZ99]	$f = 1885.65$
Seq $\#3$	Recovered Focal Length (f)
Fig. 5.5f	$f = 840.68$
Fig. 5.5g	$f = 837.84$
from [LZ99]	$f = 799.68$

uncertainty in the results. Additionally, we compare our results to the method proposed in [LZ99]. This comparison of the results should be a good test of the stability and consistency of the proposed method.

Three video sequences are used for testing. **Seq** $\#1$ contains less than 5 minutes of data. As shown in Fig. 5.5(a)-(e), different pedestrians are chosen for auto-calibration. Using the method described above, the focal length is determined using the robust TLS method. The results for this sequence are given in Table 5.1(left column). The standard deviation is low and the estimated focal length is $f = 2307.932 \pm 44.12$. **Seq** $\#2$ is another sequence used for testing, a couple of instances are shown in Fig. 5.5(f)-(g). The estimated focal lengths are very close to each other, as shown in Table 5.1 (right column - top). Similarly, results for **Seq** $\#3$ are shown in Table 5.1 (right column - bottom). The results are also compared to a standard camera calibration method proposed by Liebowitz and Zisserman [LZ99], shown in the last row for each corresponding sequence in Table 5.1. The focal lengths obtained from both methods are comparable.

The error in the results can be attributed to many factors. One of the main reason is that only a few frames are used per sequence to emulate a more practical scenario. If a large data sequence

is used, the system of equations (i.e. Eq. (5.9)) becomes more stable and thus better results may be obtained. The standard deviation in f for all our experiments is found to be less than the results reported in [KM05].

5.4 Conclusion

This chapter presented a robust and a more general solution to camera calibration by observing pedestrians. Compared to existing methods, the solution does not assume any special kind of pedestrian motion. We recognize the special geometry of the problem and present a more general and robust formulation than the existing methods. Two harmonic homologies are extracted from a pair of images containing instances of a pedestrian. Using unique properties of these homologies, linear constraints are derived to obtain the unknown camera parameters. The detected head/feet locations are used to robustly estimate the unknown camera parameters. We successfully demonstrate the proposed method on synthetic as well as on real data.

CHAPTER 6
SELF-CALIBRATION OF FREELY MOVING CAMERAS

Self-calibration differs from conventional calibration where the camera internal parameters are determined from the image of a known calibration grid or properties of the scene, such as vanishing points of orthogonal directions. The prefix *self-* is added as soon as the world's Euclidean structure is unknown, which can be seen as a case of "0D" calibration. In self-calibration the metric properties of the cameras are determined directly from constraints on the internal and/or external parameters.

The first self-calibration method, originally introduced in computer vision by Faugeras *et al.* [FLM92], involves the use of the Kruppa equations. The Kruppa equations are two-view constraints that require only the fundamental matrix to be known, and consist of two independent quadratic equations in the elements of the dual of the absolute conic. Algorithms for computing the focal lengths of two cameras given the corresponding fundamental matrix and knowledge of the remaining intrinsic parameters are provided by Hartley [Har92]. Mendonça [Men01] generalized the results in [Har92] for an arbitrary number of cameras and introduced a built-in method for the detection of critical motions for each pair of images in the sequence. Thorough analysis of critical motions which would result in ambiguous solutions by Kruppa-based methods are described in [Stu97a].

An alternative direct method for self-calibration was introduced by Triggs [Tri97], which estimates the absolute dual quadric over many views. The basic idea is to transfer a constraint on the dual image of absolute conic to a constraint on the absolute dual quadric, and hence determine the matrix representing the absolute dual quadric, from which a rectifying 3D homography can be

decomposed that transforms from projective to metric reconstruction. Heyden and Astrom [HA97] showed that metric reconstruction was possible knowing only skew and aspect ratio, and Pollefeys *et al.* [PKG99] and Heyden and Astrom [HA99] showed that zero skew alone was sufficient.

Special motions can also be used for self-calibration. Agapito *et al.* [AHR01] and Seo and Hong [SH99] solved the self-calibration of a rotating and zooming camera using the infinite homography constraint. Before their work, Hartley [Har97] solved the special case where the camera's internal parameters remain constant throughout the sequence. Frahm and Koch [FK03] showed it was also possible to solve the problem of generally moving camera with varying intrinsics but known rotation information.

In this chapter we focus on extracting internal parameters of a freely moving camera and present a simple and novel global linear solution. We do not assume any special camera motion or known camera rotation matrix as used by [AHR01, SH99, FK03, PKG99, Har97]. The proposed method relies only on point correspondences between different views from a single camera. We test our method on synthetic as well as on real data and present encouraging results.

We allow the camera to vary its internal parameters by zooming in/out. As argued by [PKG99, AHR01, Zha00, HA97], it is safe to assume zero skew, unit aspect ratio and principal point at the center of an image for currently available CCD cameras. These general assumption are used to estimate the varying focal length. The notation i and j represent any two consecutive frames from a single camera.

Figure 6.1 depicts an illustration of two images taken from a camera. Generally, two consecutive images from a camera contain some overlapping area. This overlapping area can be used to obtain the fundamental matrix $\mathbf{F_{i,j}}$, which relates a point in image I_j to a line in image I_i. As the

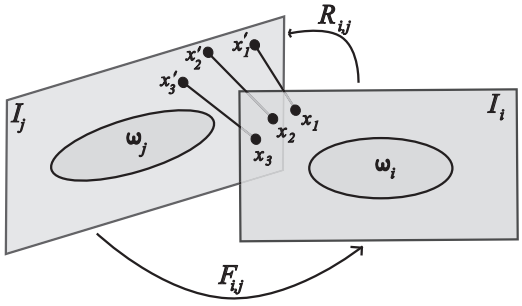


Figure 6.1: Illustration of two views from a camera: Two consecutive images from a camera contain an overlapping area. This overlapping area can be used to obtain the fundamental matrix $\mathbf{F_{i,j}}$, which relates a point in image I_j to a line in image I_i. As the internal parameters change at each view, the IAC $\boldsymbol{\omega}$ also changes.

internal parameters change at each view, IAC $\boldsymbol{\omega}$ also changes. Thus $\boldsymbol{\omega}$ needs to be computed for each image of the camera.

6.1 Linear Solution With Varying Focal Length

Consider an image sequence of n frames and let $\mathbf{K_i}$ be the intrinsic matirx for a camera at i^{th} frame, then $\mathbf{K_i}$ is of the form:

$$\mathbf{K_i} = \begin{bmatrix} f_i & 0 & 0 \\ 0 & f_i & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where $\gamma = 0, \lambda = 1, (u_o = 0, v_o = 0)$.

For a freely moving camera, the fundamental matrix can be easily obtained from successive frames and is thus used for self-calibration based on Kruppa equations [FLM92]. In order to deal with noise in an image, many techniques exist to robustly estimate the fundamental matrix [CZZ97, BGK96]. Once the fundamental matrix is computed between two different views i and j

of a camera, we have (see [Men01, FLM92]):

$$\mathbf{F_{i,j}}\boldsymbol{\omega}_\mathbf{i}^*\mathbf{F_{i,j}^T} \sim [\mathbf{e}']_\times \boldsymbol{\omega}_\mathbf{j}^* [\mathbf{e}']_\times, \tag{6.1}$$

where $\boldsymbol{\omega}_i^*$ and $\boldsymbol{\omega}_j^*$ represent the **dual IAC** for two different views, i and j, respectively. If the intrinsic parameters remain constant over different views then $\boldsymbol{\omega}_i^* \sim \boldsymbol{\omega}_j^*$ and Eq. (6.1) can be expressed as $\mathbf{F_{i,j}}\boldsymbol{\omega}_\mathbf{i}^*\mathbf{F_{i,j}^T} \sim [\mathbf{e}']_\times \boldsymbol{\omega}_\mathbf{i}^* [\mathbf{e}']_\times$.

Eq. (6.1) amounts to 3 linearly independent equations with an unknown scale, allowing for the symmetry and rank deficiency. Eq. (6.1) is not in a form that can be easily applied and traditional methods cross multiply to eliminate the unknown scale [HZ04, Men01]. Instead of taking this approach, we directly solve for the unknown scale involved in the three equations obtained from Eq. (6.1).

For a camera with unknown focal length, $\boldsymbol{\omega}^*$ for the j^{th} frame is given as:

$$\boldsymbol{\omega}_j^* = \begin{bmatrix} W_j & 0 & 0 \\ 0 & W_j & 0 \\ 0 & 0 & \alpha_j \end{bmatrix} \tag{6.2}$$

where $W_j = \alpha_j f_j^2$. The the unknown scale, i.e. α_j, is different for every image pair. For $\boldsymbol{\omega}_i^*$, the left hand side of Eq. (6.1), the unknown scale is normalized to 1. Hence for a pair of images the three unknowns are α_j, W_i and W_j.

For any $\mathbf{K_i}$ Eq. (6.1) gives us only three equations to solve for the three unknowns, owing to rank deficiency and symmetry. We formulate the problem as:

$$\mathbf{A_{i,j}Y_{i,j}} = \boldsymbol{B}_{\mathbf{i,j}} \quad \textbf{where} \quad \mathbf{Y_{i,j}} = \begin{bmatrix} W_i & W_j & \alpha_j \end{bmatrix}^{\mathbf{T}} \tag{6.3}$$

and $\mathbf{A_{i,j}}$ is a 3×3 matrix containing the coefficients of W_i, W_j and α_j; and $\boldsymbol{B}_{i,j}$ contains the known $\mathbf{F_{i,j}}$ and $[\mathbf{e}']_\times$. From the solution vector $\mathbf{Y_{i,j}}$, the intrinsic parameters for each view can be obtained as:

$$f_i = \sqrt{\mathbf{Y_{i,j_{(1)}}}}, \quad f_j = \sqrt{\mathbf{Y_{i,j_{(2)}}}/\alpha_{\mathbf{j}}}, \quad \alpha_j = \mathbf{Y_{i,j_{(3)}}}$$

A global solution for computing intrinsic parameters for a varying focal length camera over k frames is given by cascading the above equation into:

$$\underbrace{\begin{bmatrix} \mathbf{A_{i,j}} & 0 & \cdots \\ 0 & \mathbf{A_{i+1,j+1}} & \cdots \\ \vdots & \ddots & \vdots \\ 0 & 0 & \mathbf{A_{i+k,j+k}} \end{bmatrix}}_{\mathcal{A}} \underbrace{\begin{bmatrix} \mathbf{Y_{i,j}} \\ \mathbf{Y_{i+1,j+1}} \\ \vdots \\ \mathbf{Y_{i+k,j+k}} \end{bmatrix}}_{\mathcal{Y}} = \underbrace{\begin{bmatrix} \boldsymbol{B}_{i,j} \\ \boldsymbol{B}_{i+1,j+1} \\ \vdots \\ \boldsymbol{B}_{i+k,j+k} \end{bmatrix}}_{\mathcal{B}} \tag{6.4}$$

Eq. (6.4) computes a linear solution for an entire image sequence, which is fairly efficient and easy to implement. If the intrinsic parameters do not vary, Eq. (6.4) can be reformulated so that it becomes an over-determined system. This system of equations can then be solved using least squares method for the entire image sequence. Degenerate configurations for self-calibration methods are numerous and it is out of the scope of the current work to elaborate on various such configurations. See [HZ04, ZLA98] for detailed discussion on critical motion sequences that result

in degenerate conditions.

6.2 Varying Focal Length With Unknown λ

In the previous section we assumed that the aspect ratio (λ) is unity. Practically, λ remains unchanged for any single camera through its life span. Eq. (6.1) can be extended to solve for an unknown λ by selecting a reference frame q. Three images i.e. two instances of Eq. (6.1) are sufficient to solve for six unknowns. Eq. (6.1) for an image j with respect to the reference frame q can be expressed as:

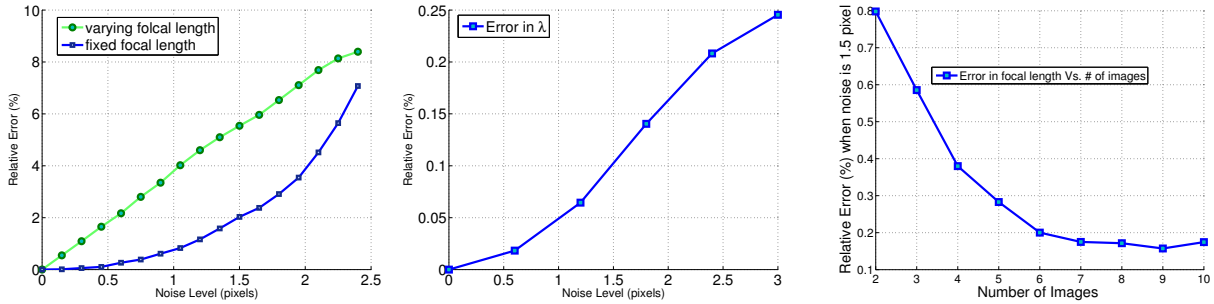
$$\mathbf{F_{q,j}} \begin{bmatrix} \lambda W_q & 0 & 0 \\ 0 & W_q & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{F_{q,j}^T} \sim [\mathbf{e}']_\times \begin{bmatrix} \lambda W_j & 0 & 0 \\ 0 & W_j & 0 \\ 0 & 0 & \alpha_j \end{bmatrix} [\mathbf{e}']_\times \tag{6.5}$$

Thus the first pair introduces four unknowns $(\lambda, W_q, W_j, \alpha_j)$ and every subsequent frame introduces only 2 unknowns (unknown scale and new focal length). Once λ is determined non-linearly, it is substituted into Eq. (6.1) for improving the estimated focal length. Eq. (6.1) can not be used to solve for any more unknown intrinsics parameters (see [HZ04]).

An obvious advantage of the above linear solution is its simplicity and computational efficiency, making it suitable for many real time applications.

6.3 Experiments And Results

Synthetic Data: In order to validate the robustness of the proposed self-calibration method, a point cloud of 1000 points [AHR01] was generated inside a unit cube to determine point correspon-



(a) Relative Error in f (b) Relative Error in λ (c) Error in f Vs. number of views

Figure 6.2: Performance of the self-calibration method VS. noise level in pixels: **(a)** The relative error of the fixed focal length when the noise is increased up to 2.5 pixels is plotted in blue, while the relative error when the focal length randomly changes between views is plotted in green. **(b)** Depicts the relative error of the aspect ratio relative to the focal length when f remains fixed. **(c)** Relative error in f estimation when the used number of views increase. The more views we use, the lesser the error rate.

dences. The synthetic camera parameters were chosen as: $f = 1000, \lambda = 1, \gamma = u_o = v_o = 0$. Gaussian noise with zero mean and standard deviation of $\sigma \leq 3$ was added to the data points used for computing the fundamental matrix. Rotation and translation between views was chosen subjectively to avoid degenerate configurations. As argued by [Tri98, Zha00], the relative difference with respect to the focal length rather than the absolute error is a more geometrically meaningful error measure. Therefore, we measure the relative error of estimated f with respect to true f while varying the noise level from 0.01 to 3 pixels. For each noise level, we performed 1000 independent trials and the results are shown in Figure 6.2.

The relative error in f increases almost linearly with respect to the noise level, as shown in Figure 6.2(a). For a maximum noise of 3 pixels, we found that the error was under 9%. The blue curve in the figure depicts the relative error when f was kept constant. We also test the proposed method for the case when f is varying randomly between the views, depicted by the green curve in Figure 6.2(a). For aspect ratio(λ), we measure the relative error w.r.t. itself (cf. Figure 6.2(b)),

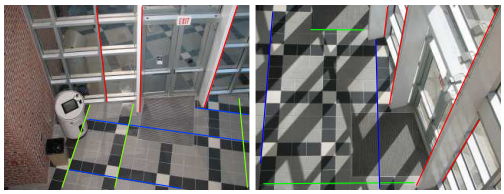
Table 6.1: Computed focal length from our method compared with vanishing points based calibration technique.

View	Our Method	Compared Method
Figure 6.3(a) (left)	3048.77	3290.36
Figure 6.3(b) (left)	1590.24	1766.74
Figure 6.3(b) (right)	3000.35	3350.17
Figure 6.3(a) (right)	2598.47	2482.24

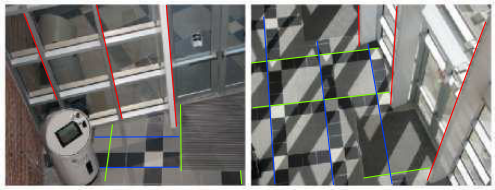
which is less than 0.25%. Relative error in estimating f (when the noise is fixed to 1.5 pixels) compared to the number of views used for the estimation is plotted in Figure 6.2(c). The relative error reduces as the number of the views increase.

Real Data: Using the method described in Section 6.1, we tested the proposed camera calibration algorithm on a number of sequences. In the first data set, two cameras, labeled l and r, are located on the second and third floor of a building monitoring a lobby entrance. The cameras are zooming in/out while translating and rotating at the same time. The height and motion of each camera is subjectively selected to allow observation of the specified area. We compared our method to the standard three parameter estimation technique using three orthogonal vanishing points [HZ04]. Results obtained from the two methods are compared in Table 6.1 and the images used are shown in Figure 6.3. The results obtained from the two methods are comparable to each other.

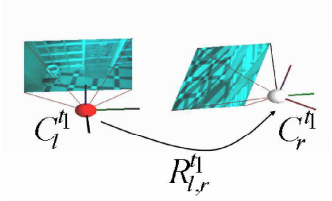
The second data set consist of a zooming in/out video taken from a driving car while looking at some houses. Figure 6.5 depicts four such instances from the sequences taken from a camera different from the one used in above data set. The focal length for each instance is shown below each image of Figure 6.5. Another set of test data is shown in Figure 6.6. The camera in this situation has fixed focal length. The figure shows only two images from the dataset with computed



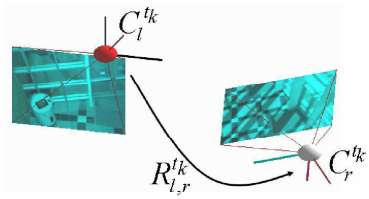
(a) A view from two neighboring cameras



(b) A view from two neighboring cameras



(c) Recovered 3D Geometry of cameras



(d) Recovered 3D Geometry of cameras

Figure 6.3: (a) and (b) are views taken from two disjoint FoV cameras looking at a lobby entrance. The two cameras are free to rotating and translating. The 3D rendering in (c) and (d) demonstrates the computed dynamic geometry of the network. This network geometry is unique at each instance of time.

Camera # 1-Estimated f(left to right): $1091.14, 1135.35, 1155.76, 1162.52, 1113.01, 1124.15$

Camera # 2-Estimated f(left to right): $1121.14, 1124.35, 1103.436, 1181.191, 1190.05, 1171.96$

Figure 6.4: Some images from a test sequence using two cameras. The cameras are translated as well as rotated. The green line indicate the knowledge of a line in world. In this particular case, the line in one camera is orthogonal to the corresponding line in the second camera.

focal lengths.

Some of the images from another test sequence are shown in Figure 6.4. The top row of the Figure depicts images from camera 1, while the bottom from camera 2. Self-calibration is performed on the sequence and the results are shown below the images in Figure 6.4. The fundamental matrix

$f = 1146.01$ $f = 843.92$ $f = 1184.21$ $f = 1723.93$

Figure 6.5: Four instances from a video sequence taken from a road while looking at some houses.

(a)$f = 2067.1957$ (b) $f = 2074.483$

Figure 6.6: (a) Two of the many images taken from a camera inside a lab, with lines used for computing the vertical vanishing points superimposed.

is computed between consecutive frames obtained from each single camera to determine the calibration matrix. As reported by Zhang [Zha00], the mean of the estimated focal length is taken as the ground truth and the standard deviation as a measure of uncertainty in the results. Thus, with a low standard deviation $\sigma = 32.05$, f is determined to be 1139.50.

6.4 Conclusion

We have successfully demonstrated a novel global linear solution approach to recovering the intrinsic parameters of a camera where each camera is assumed to undergo a general motion. Once the fundamental matrix is determined, by using just point correspondences, we solve for the internal parameters linearly. We also provide a non-linear solution for extracting the aspect-ratio for each camera. Experiments are carried out on several real and synthetic data sequences.

CHAPTER 7
PTZ CAMERA CALIBRATION

Rotating and zooming cameras are now common tools used in camera networks, with applications ranging from security and surveillance to tele-conferencing, distant learning, and virtual classrooms. A key issue with many of these applications is that the traditional off-line calibration methods [Tsa87, Zha00] are not practical due to the dynamic changes in internal and external parameters of the camera. As a result it is important that one can auto-calibrate the camera online, when required.

The first auto-calibration method was due to Faugeras et al. [FLM92] who considered a freely moving camera with unknown but constant internal parameters. Since then, several methods have been proposed [Har94, KZR03, LZ99, HA97, Tri97] some of which consider special camera motions such as pure translation [MGP96] or pure rotation [Har97]. More recent methods also consider auto-calibration under varying internal parameters [HHA99, HA97, HA99, PKG99, KTA00]. The most related work to ours is the auto-calibration method for rotating and zooming cameras by Agapito et al [AHR01], who used the mapping of the image of the absolute conic (IAC) between two images by the infinite homography to impose constraints on camera internal parameters in a pair of images. The approach that we propose in this chapter, however, is based on direct matrix decompositions of the infinite homography. The goal in most matrix decompositions is to reduce the matrix into some canonical form [GL89]. For our application, we consider two possible decompositions: one which allows to decompose the 3×3 infinite homography into a pair of projectively equivalent upper-triangular matrices, and a second one based on eigen-decomposition and direct construction of a system of homogeneous equations, which we use for solving degenerate cases.

Compared to Agapito's work our method has the advantage that we can solve for a more general camera model in the degenerate cases (i.e. solve for 5 unknowns). However, in the degenerate cases, our method does not provide any constraint on the camera aspect ratio. Also, for the non-degenerate case, our method can only allow for varying focal length. The remainder of this chapter consists of a brief description of background and notations, two main sections discussing the general case and the degenerate scenarios, and a thorough validation of the results.

7.1 Background and Notations

For a pinhole camera model, a 3D point $\mathbf{M} = [\mathbf{X}\ \mathbf{Y}\ \mathbf{Z}\ \mathbf{1}]^{\mathbf{T}}$ and its corresponding image projection $\mathbf{m} = [\mathbf{u}\ \mathbf{v}\ \mathbf{1}]^{\mathbf{T}}$ are related via a 3×4 matrix $\mathbf{P}$ by

$$\mathbf{m} \sim \underbrace{\mathbf{K}[\mathbf{r_1}\ \mathbf{r_2}\ \mathbf{r_3}\ \mathbf{t}]}_{\mathbf{P}}\mathbf{M}, \quad \mathbf{K} = \begin{bmatrix} \lambda f & \gamma & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{7.1}$$

where $\sim$ indicates equality up to multiplication by a non-zero scale factor, $\mathbf{r_i}$ are the columns of the rotation matrix $\mathbf{R}$, $\mathbf{t}$ is the translation vector, and $\mathbf{K}$ is a nonsingular 3×3 upper triangular matrix known as the camera calibration matrix including five parameters, i.e. the focal length f, the skew γ, the aspect ratio λ and the principal point at $(u_0,\ v_0)$.

The IAC, denoted by $\boldsymbol{\omega}$, is an imaginary point conic directly related to the camera internal matrix $\mathbf{K}$, via $\boldsymbol{\omega} \sim \mathbf{K^{-T}K}{-}\mathbf{1}$.

7.2 General Case: Arbitrary Rotation & Varying Focal Length

Our solution for the general case is based on using a sequence of Givens rotations [GL89], whereby we decompose the infinite homography into a pair of projectively equivalent upper-triangular matrices that provide up to 5 constraints directly on the camera parameters from only two images. As described in [GL89], a Givens rotation in the 3D space corresponds to a rotation in the plane spanned by any pair of coordinate axes. When applied to a 3×3 homography, a Givens rotation would rotate each column of the homography counter-clockwise in the plane of the two axes through an angle defined by Givens rotation matrix. By an appropriate choice of the rotation angle one can then selectively nullify any one of the entries in a homography.

Now, let $\mathbf{K}_1$ and $\mathbf{K}_2$ be the camera calibration matrices for a pair of images obtained by a fixed rotating and zooming camera. Let also $\mathbf{R}_{12}$ denote the relative rotation between the two orientations of the camera. As is well-known, independently of the scene structure, the two images are related by the infinite homography given by

$$\mathbf{H}_{21} \sim \mathbf{K}_1 \mathbf{R}_{21} \mathbf{K}_2^{-1}, \tag{7.2}$$

If we rearrange this homogeneous equation as follows

$$\mathbf{K}_1^{-1} \mathbf{H}_{21} \sim \mathbf{R}_{21} \mathbf{K}_2^{-1}, \tag{7.3}$$

then the right hand side will be merely the camera intrinsic matrix for the second image up to some unknown rotation. Therefore it can be restored to an upper-triangular matrix by a sequence

of Givens rotations, as follows: Let $\mathbf{K}_1^{-1} = [\mathbf{k}_1^T \ \mathbf{k}_2^T \ \mathbf{k}_3^T]^T$, where $\mathbf{k}_i$, $i = 1, 2, 3$ are the rows of $\mathbf{K}_1^{-1}$. Let also $\mathbf{H} = [\mathbf{h}_1 \ \mathbf{h}_2 \ \mathbf{h}_3]$, where $\mathbf{h}_i$, $i = 1, 2, 3$ are the columns of the infinite homography. Consider the Givens rotation defined by

$$\mathbf{G}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_1 & \sin\theta_1 \\ 0 & -\sin\theta_1 & \cos\theta_1 \end{bmatrix} \tag{7.4}$$

where

$$\cot\theta_1 = \frac{\mathbf{k}_2^T\mathbf{h}_1}{\mathbf{k}_3^T\mathbf{h}_1} \tag{7.5}$$

It can be verified that $\mathbf{G}_1$ rotates each side of equation (7.3) to align the last two components of the first column with the x-axis. As a result it would nullify the third element in the first column on each side of the equation. In a similar manner, we define $\mathbf{G}_2$ and $\mathbf{G}_3$ as follows:

$$\mathbf{G}_2 = \begin{bmatrix} \cos\theta_2 & \sin\theta_2 & 0 \\ -\sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{7.6}$$

where θ_2 can be obtained from

$$\cot\theta_2 = \frac{\mathbf{k}_1^T\mathbf{h}_1}{(\mathbf{k}_2^T\mathbf{h}_1\mathbf{h}_1^T\mathbf{k}_2 + \mathbf{k}_3^T\mathbf{h}_1\mathbf{h}_1^T\mathbf{k}_3)^{\frac{1}{2}}} \tag{7.7}$$

and

$$\mathbf{G}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_3 & \sin\theta_3 \\ 0 & -\sin\theta_3 & \cos\theta_3 \end{bmatrix} \tag{7.8}$$

where

$$\begin{aligned} \cot\theta_3 &= \frac{\mathbf{k}_3^T\mathbf{h}_2 \sin\theta_1 \cos\theta_2 + \mathbf{k}_2^T\mathbf{h}_2 \cos\theta_1 \cos\theta_2}{\mathbf{k}_3^T\mathbf{h}_2 \cos\theta_1 - \mathbf{k}_2^T\mathbf{h}_2 \sin\theta_1} \\ &- \frac{\mathbf{k}_1^T\mathbf{h}_2 \sin\theta_2}{\mathbf{k}_3^T\mathbf{h}_2 \cos\theta_1 - \mathbf{k}_2^T\mathbf{h}_2 \sin\theta_1} \end{aligned} \tag{7.9}$$

Applying the sequence of Givens rotations to both sides of (7.3), we get

$$\mathbf{G}_3\mathbf{G}_2\mathbf{G}_1\mathbf{K}_1^{-1}\mathbf{H}_{21} \sim \mathbf{K}_2^{-1} \tag{7.10}$$

The *significance of Givens rotations* here is that the relative rotation $\mathbf{R}_{21}$ is eliminated from equation (7.3). As a result, we obtain a homogeneous equality between two upper-triangular matrices that depend only on the unknown intrinsic parameters. Therefore let

$$\mathbf{G}_3\mathbf{G}_2\mathbf{G}_1\mathbf{K}_1^{-1}\mathbf{H}_{21} = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ 0 & k_{22} & k_{23} \\ 0 & 0 & k_{33} \end{bmatrix} \tag{7.11}$$

Assuming that the principal point remains invariant and that the skew is zero, we get the following

four independent constraints to solve for the unknown components of $\mathbf{K}_1$:

$$k_{13} + u_0 k_{11} = 0 \tag{7.12}$$

$$k_{23} + v_0 k_{22} = 0 \tag{7.13}$$

$$k_{22} - \lambda k_{11} = 0 \tag{7.14}$$

$$k_{12} = 0 \tag{7.15}$$

Although, these equations are non-linear, it turns out that they are all independent of the focal length f_2 for the second image, and all lead to low-order polynomials, which can be readily solved without resorting to optimization methods. This closed-form solution yields the unknown focal length f_1, the aspect ratio λ and the principal point (u_0, v_0). To obtain the focal length f_2 for the second camera, note that the above discussion holds symmetrically if we interchange the role of $\mathbf{K}_1$ and $\mathbf{K}_2$, and replace $\mathbf{H}_{21}$ by $\mathbf{H}_{12}$. Therefore, in the general case, our method recovers five unknown parameters in closed-form from only two images, i.e. the varying focal length, the aspect ratio and the principal point.

7.3 Degenerate Cases: Pure Pan & Pure Tilt

An important issue for calibration of a rotating and zooming camera is how a method performs when the rotation reduces to either pure pan or pure tilt. This is of particular practical importance, since existing applications such as surveillance, and tele-conferencing use PTZ cameras that are often operated under these degenerate conditions.

When the camera rotation is reduced to either pure pan or pure tilt, many existing solutions

[Har97, AHR01] in the literature, including our general solution based on Givens rotations of the infinite homography, degenerate. As a result they cannot provide all the unknown parameters from only two images. Below, we describe a new approach that allows to solve for 4 intrinsic parameters and the unknown rotation angle from two images in both pure pan and pure tilt.

Pure Pan: We show that the case of pure pan can be solved by direct construction of a set of homogeneous equations. For pure pan, we obtain 5 independent equations from two images in terms of the unknown intrinsic parameters using eigendecomposition of the infinite homography and direct use of equation (7.2).

The analysis that we present below are similar to Liebowitz and Zisserman [LZ99]. However, we investigate the case when the rotation degenerates and the camera is allowed to vary its focal length. We then investigate how the degenerate rotations such as pure pan affect the general analysis. We provide an alternative interpretation of the circular points, by correlating the eigendecomposition of the infinite homography $\mathbf{H}_{21}$ to that of $\mathbf{H}_{21}^T$.

As pointed out in [LZ99] the eigendecomposition of the infinite homography $\mathbf{H}_{21}$ provides three fixed points under the homography given by the eigenvectors: one real eigenvector $\mathbf{v}$, which corresponds to the vanishing point of the rotation axis, and two complex ones $\mathbf{I}$ and $\mathbf{J}$ that correspond to the imaged circular points of any plane orthogonal to the rotation axis. When the camera intrinsic parameters are fixed, these points provide four independent constraints on the image of the absolute conic $\boldsymbol{\omega}$ [LZ99]:

$$\mathbf{I}^T\boldsymbol{\omega}\mathbf{I} = 0, \qquad \mathbf{J}^T\boldsymbol{\omega}\mathbf{J} = 0, \qquad \mathbf{l}_\mathbf{v} \sim \mathbf{I} \times \mathbf{J} \sim \boldsymbol{\omega}\mathbf{v} \tag{7.16}$$

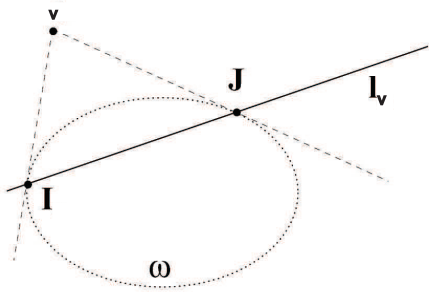


Figure 7.1: Constraints on IAC induced by the infinite homography.

where the first two impose the constraints that the circular points of a plane must lie on the IAC and the third one impose the constraint that the vanishing point of the rotation axis direction has pole-polar relationship with the vanishing line of any plane orthogonal to the axis of rotation. The construction is depicted in Figure 7.1

The question now is *what happens to these constraints if we allow the focal length to vary, and let the rotation degenerate to pure pan or tilt*. To answer these questions we also look at the line homography $\mathbf{H}_{21}^{T}$. The homography $\mathbf{H}_{21}^{T}$ also has one real eigenvector corresponding to a real eigenvalue, and two complex ones corresponding to a pair of complex conjugate eigenvalues. Let $\mathbf{a}_y \sim [0\ 1\ 0]^T$ be the axis of rotation for a panning camera. By definition this axis must be invariant to panning, i.e. $\mathbf{R}_{21}^{T}\mathbf{a}_y = \mathbf{R}_{12}\mathbf{a}_y = \mathbf{a}_y$. Since the infinite homography $\mathbf{H}_{21}$ is a conjugate rotation matrix, we have

$$\mathbf{H}_{21}^{T}\mathbf{K}_1^{-T}\mathbf{a}_y \sim \mathbf{K}_2^{-T}\mathbf{R}_{21}^{T}\mathbf{a}_y \tag{7.17}$$

$$\sim \mathbf{K}_2^{-T}\mathbf{a}_y \tag{7.18}$$

Therefore, the vanishing line of the pencil of planes perpendicular to the axis of rotation is also given by $\mathbf{K}_2^{-T}\mathbf{a}_y$.

Proposition 1 *For a zero-skew camera, under pure pan, the real eigenvector of the line homography* $\mathbf{H}_{21}^{T}$ *is the vanishing line of the pencil of planes perpendicular to the axis of rotation, if and only if the focal length and* v_0 *are fixed, but is invariant to the aspect ratio and* u_0*.*

Proposition 2 *For a zero-skew camera, under pure pan, the three eigenvectors of the line homography* $\mathbf{H}_{21}^{T}$*, given by* $\mathbf{K}_1^{-T}\mathbf{a}_y$*,* $\mathbf{l_I}$ *and* $\mathbf{l_J}$ *satisfy the pole polar relationship with the real eigenvector of* $\mathbf{H}_{21}$*, and the circular points, respectively, if and only if the focal length and* v_0 *are fixed, but is invariant to the aspect ratio and* u_0*.*

Therefore $\mathbf{l_I}$ and $\mathbf{l_J}$ may be viewed as the imaged vanishing lines of some imaginary planes that intersect the absolute conic at the circular points. As a result, the four constraints imposed by the infinite homography on the IAC are encoded in the following three homogeneous equations:

$$\mathbf{l_v} \sim \mathbf{K}_1^{-T}\mathbf{a}_y \sim \boldsymbol{\omega}\mathbf{v}, \qquad \mathbf{l_I} \sim \boldsymbol{\omega}\mathbf{I}, \qquad \mathbf{l_J} \sim \boldsymbol{\omega}\mathbf{J} \tag{7.19}$$

To see what happens when the rotation degenerates note that these equations are linear in $\boldsymbol{\omega}$, and upon taking cross-products of both sides as usual [HZ04], they can reduce to a homogeneous equation of the form

$$\mathbf{A}\mathbf{c}_{\boldsymbol{\omega}} = 0 \tag{7.20}$$

where $\mathbf{c}_{\boldsymbol{\omega}}$ is the vector of unknown components of IAC arranged in some order. When the rotation is general it can be shown that $\mathbf{A}$ has a one dimensional null space representing the solution to the

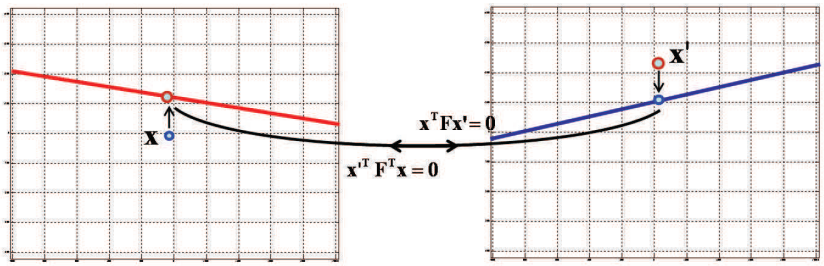


Figure 7.2: Depiction of the classical geometric error function under general camera motion based on minimizing the reprojection error subject to the epipolar constraint.

four unknowns of $\boldsymbol{\omega}$. However, when the rotation degenerates to pure pan, or pure tilt the null space becomes 2-dimensional, and only two independent constraints can be imposed on the IAC from the set of equations in (7.19). In particular, one of the constraints applies directly to the principal point:

Proposition 3 *In a zero-skew camera, for pure pan the principal point lies on the vanishing line of the pencil of planes that are perpendicular to the axis of rotation, if and only if the focal length and v_0 are fixed, but is invariant to the aspect ratio and u_0.*

To demonstrate this, denote the principal point by $\mathbf{p} \sim [u_0 \; v_0 \; 1]^T$. It follows that

$$\mathbf{a}_y^T \mathbf{K}_2^{-1} \mathbf{p} = \mathbf{a}_y^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = 0 \tag{7.21}$$

which proves the result being sought.

Remark: The above propositions hold for pure tilt if we simply exchange the role of u_0 with v_0.

In summary:

- Under degenerate rotation the eigenvector $\mathbf{l_v}$ corresponding to the real eigenvalue of $\mathbf{H}_{21}^{T}$ provides one constraint on the location of the principal point in the form

$$\mathbf{p}^T\mathbf{l_v} = 0 \tag{7.22}$$

  It is important to note that (7.22) does not hold under general rotation.

- Under degenerate camera rotation the IAC can be written as a one parameter family of conics given by

$$\boldsymbol{\omega}(\alpha) = \boldsymbol{\omega}_1 + \alpha\boldsymbol{\omega}_2 \tag{7.23}$$

  where $\boldsymbol{\omega}_1$ and $\boldsymbol{\omega}_2$ span the right null-space of $\mathbf{c}_{\boldsymbol{\omega}}$. This can be solved linearly by applying an additional constraint, for instance, by assuming known or fixed aspect ratio. Note that one could also formulate the problem similarly for DIAC. However, the constraints would then be quadratic leading to two-fold ambiguity. For degenerate rotations, it can be verified that the zero-skew constraint cannot resolve the ambiguity.

To conclude this section, in order to solve for a more general camera model under pure pan and zoom from a minimum set of two images, we resort to a solution based on direct construction of a set of homogeneous equations. For this purpose, we first verify that under pure pan and zoom the

imaged circular points of the plane perpendicular to the axis of rotation will become of the form

$$\begin{bmatrix} a \pm ib \\ v_0 \\ 1 \end{bmatrix} \tag{7.24}$$

where a and b can be written in terms of the unknown intrinsic parameters and the panning angle. Therefore the real and imaginary parts of the circular points may be used directly to impose constraints on the intrinsic parameters and the rotation angle. On the other hand, we can also construct additional homogeneous equations directly from (7.2) as follows:

Let $\mathbf{H}_{21} = [\mathbf{h}_1^T, \mathbf{h}_2^T, \mathbf{h}_3^T]^T$, $\mathbf{K}_1 = [\mathbf{k}_{11}^T, \mathbf{k}_{12}^T, \mathbf{k}_{13}^T]^T$, $\mathbf{R}_{21} = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3]^T$, and $\mathbf{K}_2 = [\mathbf{k}_{21}, \mathbf{k}_{22}, \mathbf{k}_{23}]$, where $\mathbf{H}_{21}$ and $\mathbf{K}_1$ are expressed in terms of their rows, and $\mathbf{R}_{21}$ and $\mathbf{K}_2$ are expressed in terms of their columns. We can then write the following set of homogeneous equations

$$\mathbf{h}_i^T \mathbf{k}_{2j} \sim \mathbf{k}_{1i}^T \mathbf{r}_j, \quad i, j = 1, ..., 9 \tag{7.25}$$

The above equations together with the two constraints derived from the circular points provide only 5 independent constraints on the unknown rotation angle and the intrinsic parameters. Unfortunately, unlike the general case described earlier, for pure panning and zooming it is not possible to establish a constraint on the aspect ratio λ. Therefore, assuming that the aspect ratio is known (e.g. $\lambda = 1$), and that except for the focal length all other intrinsic parameters remain invariant, our constraints lead to low order polynomials, which can be readily solved. Therefore, our solution provides four unknown intrinsic parameters (other than λ) and the rotation angle from only two

images for pure panning under variable focal length and zero skew.

Pure Tilt: The case for pure tilt is quite similar to pure pan, with minor differences. All the analyses can be equally applied to tilting. In particular, as in pure pan, it can be proved that for pure tilt and zooming the principal point must lie on the vanishing line of the pencil of planes that are perpendicular to the axis of rotation. This provides a constraint similar to (7.22) on the principal point of the camera. Also, the real and the imaginary parts of the imaged circular points depend on the intrinsic parameters and the rotation angle as before, and can be used to impose constraints on the unknown parameters. However, the construction in (7.25) is somewhat different for the case of pure tilt, because the infinite homography in the case of pure tilt is of the form

$$\mathbf{H}_{2,1} \sim \begin{bmatrix} 1 & h_{12} & h_{13} \\ 0 & h_{22} & h_{23} \\ 0 & h_{32} & h_{33} \end{bmatrix} \tag{7.26}$$

providing only 5 equations. Again, it can be shown that in the case of pure tilt, none of the above constraints depends on the camera aspect ratio λ. As a result, it is not possible to recover λ for a purely tilting and zooming camera. Therefore, our solution provides again four unknown intrinsic parameters (i.e. the two focal lengths, and the principal point) plus the rotation angle from only two images for pure tilting under zero skew and variable focal length.

Cascading degenerate cases: One interesting and practical solution for the degenerate case occurs when the camera first pans and then tilts (or vice versa), leading to a minimum case of three images, with the corresponding infinite homographies $\mathbf{H}_{21}$ and $\mathbf{H}_{32}$. In such case, the principal

point can be recovered immediately using

$$\mathbf{p} \sim \mathbf{l}_{\mathbf{v}}^{21} \times \mathbf{l}_{\mathbf{v}}^{32} \tag{7.27}$$

where $\mathbf{l}_{\mathbf{v}}^{21}$ and $\mathbf{l}_{\mathbf{v}}^{32}$ are the eigenvectors corresponding to the real eigenvalues of $\mathbf{H}_{21}^{T}$ and $\mathbf{H}_{32}^{T}$.

Therefore, the problem would immediately reduce to the simple case of known principal point, which in most auto-calibration methods, including ours, simplifies the remaining set of equations. This scenario can be, for instance, used in a network of PTZ cameras at the cold start, for determining the principal point once and use it throughout the operation of the network, assuming that it remains invariant. Note also that in this case our method recovers all camera parameters including the aspect ratio, since the first and the third image have general rotation, although the other two pairs of combinations are degenerate.

7.4 Geometrically Optimized Refinement

Most practical auto-calibration methods comprise of two steps [Har98, HZ04]: In the first step an initial solution is found by solving directly a set of algebraic constraints that are often linear - although in some cases such as ours or Kruppa's equations may also be non-linear; In the second step the initial solution is refined by minimizing an error function, which preferably should reflect the geometry of the configuration [Har98, HZ04]. The most versatile geometric error function is based on minimizing the reprojection error [HZ04], which aims to simultaneously refine the point correspondences and the camera parameters. To make the problem tractable and less sensitive to initialization, under general camera motion the reprojection error is often minimized subject to the

constraint that the orthogonal distance between a reprojected point and the corresponding epipolar line is minimized. This is depicted schematically in Figure 7.2.

For pure rotation, however, the epipolar geometry does not exist. As a result, in existing literature the general form of the reprojection error is used. In this section, we derive a novel geometric error for a purely rotating camera, similar in spirit to the epipolar constraint, which increases noise resilience and tractability, and reduces sensitivity to initial point correspondences. We first briefly describe the classical error functions used for pure rotation and then derive our new geometric error function.

7.4.1 Classical Error Functions

When a set of matches $\mathbf{x_i} \leftrightarrow \mathbf{x_i'}$ are known between a pair of images, it is generally assumed that there are errors in measurements of both $\mathbf{x_i}$ and $\mathbf{x_i'}$. In order to minimize this error, one of the first techniques generally used, specially for a PTZ camera, involves minimizing the cost function:

$$\mathcal{C}_{\mathbf{alg}} = \sum\nolimits_{\mathbf{j=1}}^{\mathbf{n-1}} \parallel \mathbf{K_j K_j^T} - \mathbf{H_j K_0 K_0^T H_j^T} \parallel_{\mathbf{F}}^{\mathbf{2}} \tag{7.28}$$

where subscript F indicates the use of Frobenius norm. This cost function minimizes the algebraic error. The disadvantage is that the quantity being minimized is not geometrically or statistically meaningful [HZ04]. The solutions based on algebraic distances are generally used as starting points for other non-linear methods.

Alternative error functions are based on geometric distances in the image plane that usually involve minimizing the error between the measured and the estimated reprojected image coordinates. Thus we seek a Maximum Likelihood (ML) solution assuming that the error in the measurement is

Gaussian. For a geometrically meaningful minimization of the overall error and for camera parameters refinement, researchers [AHR01, SH99, TMH99] have used a bundle adjustment approach. Given n images and m corresponding points, the maximum likelihood estimate can be obtained by minimizing the following error function:

$$\mathcal{C}_{\mathbf{ml}} = \sum_{\mathbf{i=1}}^{\mathbf{n}} \sum_{\mathbf{j=1}}^{\mathbf{m}} \parallel \hat{\mathbf{x}}_{\mathbf{ij}} - \mathbf{K_i R_i} \bar{\mathbf{X}}_{\mathbf{j}} \parallel^{\mathbf{2}} \tag{7.29}$$

Thus the squared error sum between the image measurement ($\hat{\mathbf{x}}_{ij}$) and the projection of the true image points for all points across all views is minimized. Minimizing (7.29) is a non-linear problem, which is solved by Levenberg-Marquardt iterative minimization method [PFT88]. Minimizing (7.29) is equivalent to the Maximum Likelihood (ML) estimate. Agapito et al. [AHR01] show that prior knowledge of the parameters can also be incorporated for a ML estimate.

The bundle adjustment solution is geometrically meaningful and it can be visualized as *adjusting* the *bundle* of rays between each camera center and a set of 3D points. It provides a ML solution while being tolerant to missing data. This method can also be viewed as minimizing the reprojection error between two images. In fact it assumes that the optimal (ML) solution lies close to the initial solution. Thus it aims to change (or perturb) the estimated points and the camera parameters such that the cost function is minimized subject to the reprojection model defined by the homography relationship between the views. Therefore the probability of a true solution will follow a normal distribution. Formally, the measured location $\hat{\mathbf{x}}$ is related to the true location by a Gaussian additive noise η:

$$\hat{\mathbf{x}} = \mathbf{x} + \eta = \mathcal{F}(\mathbf{K}, \mathbf{R}) + \eta \tag{7.30}$$

where $\mathcal{F}(\mathbf{K}, \mathbf{R})$ is the reprojection model for the true values of the image points given an estimate of the parameters $\mathbf{K}$ and $\mathbf{R}$. Therefore the probability of the true solution is:

$$p(\hat{\mathbf{x}}|\mathbf{K}, \mathbf{R}, \sigma) = \mathcal{N}(\hat{\mathbf{x}}|\mathcal{F}(\mathbf{K}, \mathbf{R}), \sigma) \tag{7.31}$$

which one aims to maximize.

7.4.2 Optimal Geometric Error

In contrast to the above solution, we propose a geometrically *optimized* error function. By *optimized* we mean a cost function tailored specifically to our special camera model i.e. pure rotation and zoom. We initially explain our cost function for the simple case of single axis rotation and then extend the results to the more general case of pan-tilt motion.

Pure Pan: For a panning PTZ camera, a point $\mathbf{x}$ in the first image $\mathbf{I_1}$ is related to the corresponding point $\mathbf{x}'$ in the second image $\mathbf{I_2}$ via the infinite homography:

$$\mathbf{x}' \sim \mathbf{K_2 R_y K_1^{-1} x} \tag{7.32}$$

where the rotation matrix $\mathbf{R_y}$ is parameterized as $\mathbf{R}_y = \begin{bmatrix} c & 0 & -s \\ 0 & 1 & 0 \\ s & 0 & c \end{bmatrix}$ where $c = \cos\theta_y$ and $s = \sin\theta_y$. Using the two linear constraints given by

$$\mathbf{x}' \times (\mathbf{K_2 R_y K_1^{-1} x}) = \mathbf{0} \tag{7.33}$$

we then express c and s in terms of $\mathbf{K}_i$ and the feature points $\mathbf{x}$ and $\mathbf{x}'$. Upon substituting c and s into the Pythagorean identity

$$c^2 + s^2 - 1 = 0 \tag{7.34}$$

and rearranging, we get:

$$\mathbf{x'^T Q x'} = \mathbf{0} \tag{7.35}$$

where $\mathbf{Q}$ is a conic given by the 3×3 symmetric matrix,

$$\mathbf{Q} = \begin{bmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{bmatrix} \tag{7.36}$$

$$\text{with } a = (\mathbf{x_y} - \mathbf{v_0})^{\mathbf{2}} \tag{7.37}$$

$$b = 0 \tag{7.38}$$

$$c = -f_1^2 - (\mathbf{x_x} - \mathbf{u_0})^{\mathbf{2}} \tag{7.39}$$

$$d = (4u_0v_0\mathbf{x_y} - \mathbf{2u_0x_y^2} - \mathbf{2u_0v_0^2}) \tag{7.40}$$

$$e = (2v_0\mathbf{x_x^2} - \mathbf{4x_xv_0u_0} + \mathbf{2v_0u_0^2} + \mathbf{2v_0f_1^2}) \tag{7.41}$$

$$f = u_0^2\mathbf{x_y^2} - \mathbf{2v_0u_0^2x_y} + \mathbf{f_2^2v_0^2} - \mathbf{f_1^2v_0^2} \\ -2f_2^2v_0\mathbf{x_y} + \mathbf{f_2^2x_y^2} + \mathbf{2v_0^2u_0x_x} - \mathbf{v_0^2x_x^2} \tag{7.42}$$

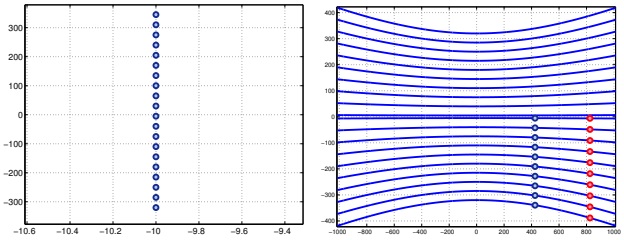
where f_1 and f_2 are the camera focal lengths in views $\mathbf{I_1}$ and $\mathbf{I_2}$, respectively.

The conic $\mathbf{Q}$, in addition to the camera parameters, is parameterized by the image point $\mathbf{x} = [\ \mathbf{x_x} \quad \mathbf{x_y} \quad 1\ ]^{\mathbf{T}}$. What equation (7.35) implies is that for every point $\mathbf{x}$ in $\mathbf{I_1}$, the corresponding point $\mathbf{x}'$ in $\mathbf{I_2}$ must lie on the conic $\mathbf{Q}$, which is defined by the camera parameters and the point $\mathbf{x}$. Similarly, for transformation from $\mathbf{I_2}$ to $\mathbf{I_1}$, it can be shown that for every point $\mathbf{x}'$ in $\mathbf{I_2}$, the corresponding point $\mathbf{x}$ in $\mathbf{I_1}$ must lie on a conic $\mathbf{Q}'$:

$$\mathbf{x^T Q' x} = \mathbf{0} \tag{7.43}$$

where $\mathbf{Q}'$, in contrast to $\mathbf{Q}$, is defined by the camera parameters and the point $\mathbf{x}' = [\ \mathbf{x'_x} \quad \mathbf{x'_y} \quad 1\ ]^{\mathbf{T}}$ in $\mathbf{I_2}$.

In summary, as a camera pans the points in the image plane trace a conic trajectory. It can be



(a) view $\mathbf{I_1}$ (b) view $\mathbf{I_2}$

Figure 7.3: (a) image points $\mathbf{x_i}$ in $\mathbf{I_1}$. (b) For pure pan the corresponding points lie on a conic in $\mathbf{I_2}$.

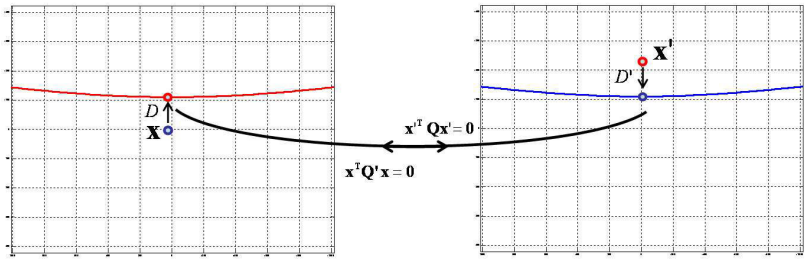


Figure 7.4: Depiction of the proposed new geometric error function under pure rotation.

readily verified from (7.37)-(7.39) that these conics are in fact hyperbolas. This is demonstrated in Figure 7.3. Points corresponding to $\mathbf{x_i}$ in view $\mathbf{I_1}$ lie on a hyperbolic trajectory in $\mathbf{I_2}$. Exactly where a corresponding point lies on the hyperbola depends on the rotation angle. As shown in the Figure 7.3(b), the blue dots are the corresponding points when the pan angle was $\theta_y = 20°$ whereas it was $\theta_y = 35°$ for the red dots. Therefore, in minimizing the reprojection error, instead of searching in the neighborhood of a points in all directions, we can minimize the orthogonal distance of points from the hyperbolic curves.

7.4.2.1 Derivation of the Cost function

While a fundamental matrix for a general camera motion defines a correlation mapping from points to lines, the discussion above shows that a PTZ camera, undergoing pan motion (or tilt for that

matter), defines quadratic curves for mapping of the corresponding image points $\mathbf{x} \leftrightarrow \mathbf{x}'$. Thus, instead of minimizing the distance of feature points to epipolar lines [ZDF95], for pure rotation we can minimize the distance of points to conics.

The geometric distance $\mathcal{D}$ of a point $\mathbf{x}$ to a conic $\mathbf{Q}'$ can be obtained using Sampson's rule [HZ04]

$$\mathcal{D} = \epsilon^{\mathcal{T}}(\mathbf{J}\mathbf{J}^{\mathbf{T}})^{-\mathbf{1}}\epsilon \tag{7.44}$$

where $\epsilon = \mathbf{x}^{\mathbf{T}}\mathbf{Q}'\mathbf{x}$ is the cost associated with $\mathbf{x}$ and $\mathbf{J} = \left[\frac{\partial(\mathbf{x}^{\mathbf{T}}\mathbf{Q}'\mathbf{x})}{\partial \mathbf{x}_{\mathbf{x}}}, \frac{\partial(\mathbf{x}^{\mathbf{T}}\mathbf{Q}'\mathbf{x})}{\partial \mathbf{x}_{\mathbf{y}}} \right]$ is a matrix of partial derivatives.

Using the chain rule, the elements of $\mathbf{J}$ are computed as:

$$\frac{\partial(\mathbf{x}^{\mathbf{T}}\mathbf{Q}'\mathbf{x})}{\partial \mathbf{x}_{\mathbf{x}}} = \frac{\partial(\mathbf{x}^{\mathbf{T}}\mathbf{Q}'\mathbf{x})}{\partial \mathbf{x}_{\mathbf{x}}}\frac{\partial \mathbf{x}}{\partial \mathbf{x}_{\mathbf{x}}} = 2(\mathbf{Q}'\mathbf{x})_{\mathbf{1}}$$

and similarly

$$\frac{\partial(\mathbf{x}^{\mathbf{T}}\mathbf{Q}'\mathbf{x})}{\partial \mathbf{x}_{\mathbf{y}}} = 2(\mathbf{Q}'\mathbf{x})_{\mathbf{2}}$$

where the subscripts 1 and 2 denote the first and the second component of the vector, respectively.

Using (7.44), the distance of a point $\mathbf{x}$ to a conic $\mathbf{Q}'$ thus reduces to:

$$\mathcal{D} = \frac{(\mathbf{x}^{\mathbf{T}}\mathbf{Q}'\mathbf{x})^{\mathbf{2}}}{\triangle((\mathbf{Q}'\mathbf{x})_{\mathbf{1}}^{\mathbf{2}} + (\mathbf{Q}'\mathbf{x})_{\mathbf{2}}^{\mathbf{2}})} \tag{7.45}$$

For symmetric error minimization, the cost function would be then of the form

$$\sum_{i=1}^{n}\left(\frac{(\mathbf{x_i^T Q'_i x_i})^2}{4((\mathbf{Q'_i x_i})_1^2+(\mathbf{Q'}_i\mathbf{x_i})_2^2)}+\frac{(\mathbf{x_i'^T Q_i x'_i})^2}{4((\mathbf{Q_i x'_i})_1^2+(\mathbf{Q_i x'_i})_2^2)}\right)$$
$$=\sum_{i=1}^{n}(\mathcal{D}+\mathcal{D}') \tag{7.46}$$

That is, the camera intrinsic and extrinsic parameters and the correct feature point locations must minimize the sum of distances to the conics (cf. Figure 7.4). The minimum of this non-linear cost function is sought using the Levenberg-Marquardt algorithm. Thus we have reduced the search space of true feature locations to quadratic curves.

Tilt Motion: The above discussion equally applies to pure tilt, or in fact to any single axis rotation.

7.4.3 Pan-Tilt Motion

For a PTZ camera undergoing both pan and tilt motion, (7.32) is modified as:

$$\mathbf{x}' \sim \mathbf{K_2 R_x R_y K_1^{-1} x} \tag{7.47}$$

where $\mathbf{R_y}$ is as defined above, and $\mathbf{R_x}$ defines rotation around the x-axis by θ_x. In principle, there are sufficient number of constraints to eliminate the two angles. However, due to non-linearity, this is not straightforward. Therefore, we parameterize $\mathbf{R_y}$ as before in terms of c and s, and also parameterize $\mathbf{R_x}$ by $c' = \cos\theta_x$ and $s' = \sin\theta_x$. Similar to pan case, we then express c and s in terms of feature points and the camera parameters to obtain a conic as defined in (7.35). The difference now is that the conic $\mathbf{Q}$ (and similarly $\mathbf{Q}'$) contains the tilt angle components c' and s', which are used as additional parameters in the cost function (7.46).

Our overall algorithm is thus as follow: For a PTZ camera, we solve for the unknown $\mathbf{K_i}$ and $\mathbf{R}$ using the method described in Section 7.2. If the camera motion is just pan or just tilt, we use the method described in Section 7.3. We then refine the estimated parameters by minimizing the new geometric error.

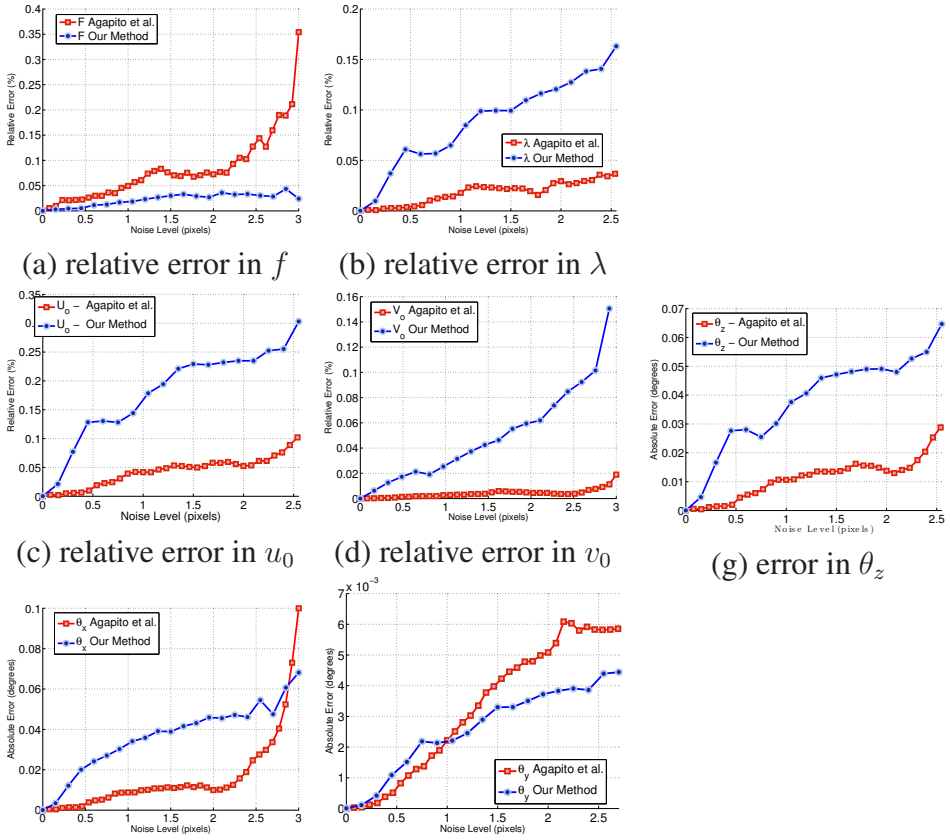
7.5 Experimental Results

In this section, we show an extensive set of experimental results on both synthetic and real data to evaluate the proposed solutions and compare with the state of the art.

7.5.1 Synthetic Data

We performed detailed experimentation on the effect of noise on camera parameter estimation over 1000 independent trials. For this purpose, a point cloud of 1000 random points [AHR01] was produced inside a unit cube to generate image point correspondences. Simulated camera has a focal length of 1000, aspect ratio of $\lambda = 1.5$, skew $\gamma = 0$, and the principal point at $(u_0, v_0) = (512, 384)$, for image size of 1024×768.

Performance vs. Noise Level: In this experimentation, we compare our results to Agapito et al. [AHR01] without performing the refinement proposed in section 7.4. Errors for estimated camera intrinsic and extrinsic parameters are measured with respect to the ground truth, while adding a zero-mean Gaussian noise varying from 0.1 pixels to 3 pixels. The results show the average performance over 1000 independent trials. As argued by [Tri98, Zha00], the relative difference with respect to the focal length rather than the absolute error is a more geometrically meaningful error measure for f, λ and (u_0, v_0). Figure 7.5 summarizes the results for intrinsic parameters. For



(a) relative error in f (b) relative error in λ

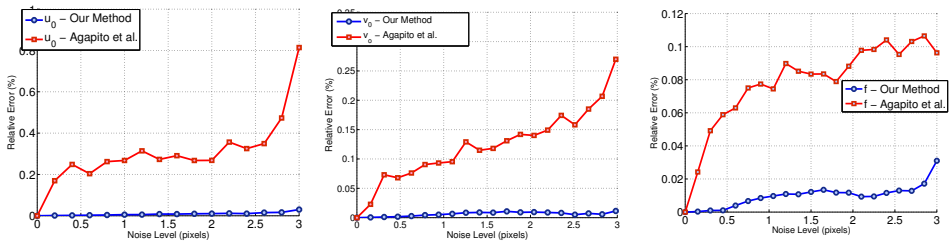
(c) relative error in u_0 (d) relative error in v_0 (g) error in θ_z

(e) error in θ_x (f) error in θ_y

Figure 7.5: Performance vs. Noise Level: averaged over 1000 independent trials. Results without geometric optimization compared to Agapito et al.

noise level of 3 pixels, which is larger than the typical noise in practical calibration [Zha00], the relative error for the focal length f is 0.1%. The maximum relative error for the aspect ratio is less than 0.2%, the relative error in u_0 is less than 0.35%, and the relative error in v_0 is less than 0.16%. Excellent performance is also achieved for all extrinsic parameters as shown in the figure, i.e. absolute errors of less than a tenth of a degree for all rotation angles θ_x, θ_y and θ_z.

Comparison with Agapito et al. [AHR01]: We perform the comparison using the same setup as above. Figure 7.5 summarizes our results, where the results of our method are drawn in blue and those of Agapito et al. [AHR01] in red. Without refinement, the errors for both methods are of the



(a) relative error in u_0 (b) relative error in v_0 (c) Relative Error in f

Figure 7.6: Performance vs. Noise Level: averaged over 1000 independent trials. Results after geometric optimization compared to ML-optimized Agapito et al.

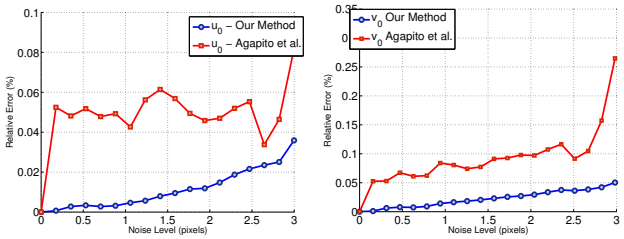
same order, although we obtained slightly better performance for the focal length, while Agapito's method did slightly better on other parameters. The main advantage of our method here is that we obtain more parameters using fewer images, by trading off linearity.

7.5.1.1 Results After Refinement

We refined the results obtained in the previous subsection by minimizing the geometric cost function that we derived in (7.46). We compare our refined results with the ML estimate method proposed by [AHR01] as defined in (7.29). As demonstrated below, our refinement approach consistently outperforms the classical ML refinement.

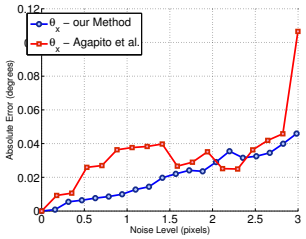
Pan Motion: The results are shown in Figure 7.6. Figure 7.6(a) shows the relative error in u_0, which is found to be less than 0.2% for a noise of up to 3 pixels. Similarly, noise for the v_0 and f is also very low. The error in the proposed estimated method is comparably lower than the classical ML estimation method.

Pan-Tilt Motion: For the case when the camera is both panning and tilting, the error curves are shown in Figure 7.7. The error for the parameters u_0, v_0, f, and θ_x is lower than 0.04%,0.1%, 0.04% and 0.05°, respectively.

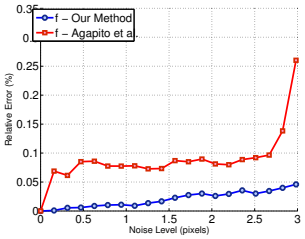


(a) relative error in u_0

(b) relative error in v_0



(c) Absolute Error in θ_x



(d) Relative Error in f

Figure 7.7: Performance vs. Noise Level: averaged over 1000 independent trials for pan-tilt motion. Results after geometric optimization compared to ML-optimized Agapito et al.

The above results indicate that minimization based on the optimal geometric error function derived in this chapter consistently give better results than the traditional ML estimate for the PTZ camera.

7.5.2 Real Data

Several experiments are performed on real data. The data was obtained by a SONY® SNC-RZ30N PTZ camera with an image resolution of 320×240. Hence the ground truth rotation angles are known. Image features and correspondences are obtained by using the SIFT algorithm [Low04]. In order to evaluate our results, we use an approach similar to [Zha00], i.e. use the uncertainty associated with the estimated intrinsic parameters characterized by their median deviation over many images, while taking into account the ground truth rotation angles. We deliberately keep f_1 and f_2 same so that we can estimate the accuracy of parameters estimations in the absence of

(a)

Combination	θ_y	u_0	v_0	f_1	f_2
$\mathcal{C}_1$	0.97	196.185	206.07	682.07	678.01
$\mathcal{C}_2$	0.91	120.5	187.91	637.81	635.98
$\mathcal{C}_3$	1.02	195.35	210.89	785.97	774.81
$\mathcal{C}_4$	0.88	124.24	147.58	662.13	659.43
$\mathcal{C}_5$	1.26	187.33	198.13	759.02	756.57
$\mathcal{C}_6$	0.8	128.92	179.62	768.4	765.42
$\mathcal{C}_7$	1.08	130.96	188.92	662.13	659.43
$\mathcal{C}_8$	0.93	135.49	155.49	786.22	778.48
$\mathcal{C}_9$	1.16	165.89	182.02	786.22	778.48
$\mathcal{C}_{10}$	0.89	141.03	194.5	673.24	668.83
$\mathcal{C}_{11}$	1.19	118.52	198.93	666.35	660.04
$\mathcal{C}_{12}$	0.94	196.9	198.13	756.17	755.07
$\mathcal{C}_{13}$	1.05	184.08	181.48	770.77	775.22
$\mathcal{C}_{14}$	0.87	175.58	200.17	662.59	660.79
$\mathcal{C}_{15}$	0.97	153.52	162.65	797.4	790.03
Mean	0.99	156.97	186.17	723.77	719.77
M. Deviation	0.08	29.28	9.29	41.23	34.95

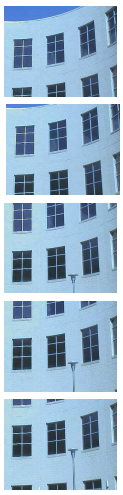
(b)

Figure 7.8: Sample images from pan sequence. Estimated parameters and their statistics.

ground truth for intrinsic camera parameters.

Pan Motion: Around 15 images were captured while panning the camera. The rotation between the successive frames is 1°. In order to further investigate the stability of the proposed method, we apply it to all the combinations of 14 images out of the 15 images. The results are shown in Figure 7.8(b). A few of the images are shown in Figure 7.8(a). The second column depicts the estimated rotation angles to be $.99^\circ$, which is almost equal to the ground truth rotation angle. Camera zoom remained constant in the sequences; hence column 5 and 6 i.e. f_1 and f_2 are very close to each other. The results also demonstrate low median deviation for the estimated parameters.

Tilt Motion: Another sequence for the degenerate condition, i.e. tilt, was taken while keeping the focal length the same. Around 21 images were captured with a tilt rotation of 1°. We apply our method to all the combinations of 20 images out of the total 21 images, as in the pan case. The results are shown in Figure 7.9(b) and a few images are shown in Figure 7.9(a). The rotation



(a)

Comb.	θ_x	u_0	v_0	f_1	f_2
$\mathcal{C}_1$	0.97	196.185	206.07	682.07	678.01
$\mathcal{C}_2$	0.91	120.5	187.91	637.81	635.98
$\mathcal{C}_3$	1.02	195.35	210.89	785.97	774.81
$\mathcal{C}_4$	0.88	124.24	147.58	662.13	659.43
$\mathcal{C}_5$	1.26	187.33	198.13	759.02	756.57
$\mathcal{C}_6$	0.8	128.92	179.62	768.4	765.42
$\mathcal{C}_7$	1.08	130.96	188.92	662.13	659.43
$\mathcal{C}_8$	0.93	135.49	155.49	786.22	778.48
$\mathcal{C}_9$	1.16	165.89	182.02	786.22	778.48
$\mathcal{C}_{10}$	0.89	141.03	194.5	673.24	668.83
$\mathcal{C}_{11}$	1.19	118.52	198.93	666.35	660.04
$\mathcal{C}_{12}$	0.94	196.9	198.13	756.17	755.07
$\mathcal{C}_{13}$	1.05	184.08	181.48	770.77	775.22
$\mathcal{C}_{14}$	0.87	175.58	200.17	662.59	660.79
$\mathcal{C}_{15}$	0.97	153.52	162.65	797.4	790.03
Mean	0.95	178.26	222.07	823.62	844.33
M. Dev.	0.15	17.95	34.62	9.44	49.8

(b)

Figure 7.9: (a) sample images. (b) Results obtained from the tilt sequence and their statistics.

angle is estimated to be 0.95° and the two estimated focal lengths are very close to each other as expected.

Pan-Tilt Motion: Another sequence for evaluating the general rotation, as described in Section 7.2, is taken while panning with $\theta_y = 2^\circ$ and tilting with $\theta_x = 2^\circ$, and keeping the focal length fixed for the camera. We apply the method to all the combinations of 6 images from the total of 7 images. The results are shown in Figure 7.10. The pan angle θ_y is estimated at 1.84°, whereas the tilt angle θ_x was estimated as 2.06°. The aspect ratio λ is estimated as 1.06, the two focal lengths between the images are also very close to each other. The principal point is also estimated to be close to the center of the image.

7.6 Discussion and Concluding Remarks

This chapter makes three main contributions to auto-calibration of rotating and zooming camera: (i) By successive rotations of the infinite homography and axis alignments, we derive a new

(a)

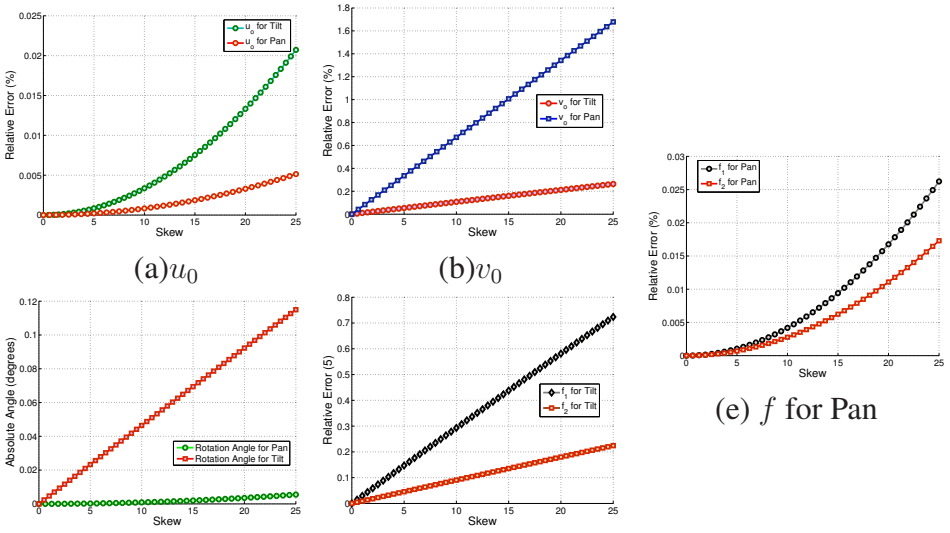
Comb.	λ	u_0	v_0	f_1	f_2	θ_x	θ_y
$\mathcal{C}_1$	1.97	185.98	127.21	588.98	610.06	2.34	1.15
$\mathcal{C}_2$	0.36	118.65	166.2	676.78	738.14	1.94	1.16
$\mathcal{C}_3$	0.85	164.27	121.88	827.88	765.56	2.69	1.58
$\mathcal{C}_4$	0.61	171.69	168.06	892.77	780.88	1.24	2.98
$\mathcal{C}_5$	1.45	261.95	150.47	942.02	1021.1	1.85	1.84
$\mathcal{C}_6$	0.56	177.80	139.42	672.32	696.32	2.76	2.2
$\mathcal{C}_7$	1.62	152.92	168.65	821.43	837	1.58	1.95
Mean	1.06	176.17	148.84	774.6	778.44	2.06	1.84
M. Dev.	0.49	14.29	17.59	120.59	63.29	0.4	0.36

(b)

Figure 7.10: (a) Sample images from pan-tilt sequence. (b) Estimated parameters and their statistics.

non-linear solution that provides five intrinsic parameters (i.e. $f_1, f_2, u_0, v_0, \lambda$) from only two images; (ii) we focus on PTZ camera applications by performing thorough analysis of degenerate single-axis rotations; (iii) we derive a new geometric error function for refinement of solution that outperforms classical ML reprojection error. Although Agapito et al. [AHR01] use more images than required by our method, they do provide a linear solution, whereas our solution is non-linear but in terms of low-order polynomials.

On the other hand, pure pan or tilt are unstable cases for their method. Therefore, using $\gamma = 0$ constraint is not sufficient and they have to assume known λ. Although assuming a non-zero skew introduces instability in our method as well, we are able to solve for 4 intrinsic parameters (i.e. f_1, f_2, u_0, v_0) and the rotation angle (θ_x or θ_y) using only an image pair. We have investigated the effect of increasing/non-zero skew on the stability of estimating other parameters. Results are shown in Figure 7.11. Except for v_0, error in other estimated parameters increases non-linearly when we have a panning camera, as seen in Figure 7.11. The error in parameter estimation while the camera is tilting is linear, except for u_0 (cf. Figure 7.11). A particular remark to be made here



(a)u_0 (b)v_0

(c) rotation angle (d)f for Tilt (e) f for Pan

Figure 7.11: Effect of non-zero skew on the error in estimation of other parameters.

is that u_0 is less sensitive to non-zero skew for pan, and conversely v_0 is less sensitive to non-zero skew in tilt. Also, we found that other parameters were in general less sensitive to non-zero skew under panning.

CHAPTER 8
CONFIGURING A NETWORK OF CAMERAS

In Part II, we addressed the problem of calibrating any individual camera in the network. Our goal in this chapter is to demonstrate that one can establish a common world reference frame to recover absolute and relative camera orientations even with non-overlapping FoVs.

The main motivation for deploying networked cameras is that a single camera, even if allowed to rotate or translate, is not sufficient to cover a large area. Figure 8.1 shows an active example of a configuration where two fixed cameras are monitoring one particular area. A more general case with a wide range of applications is when the deployed disjoint FoV cameras may be allowed to move freely in 3D space, e.g. on roaming security vehicles. By employing multiple cameras with non-overlapping or disjoint FoV, we would like to maximize the monitoring area in addition to inferring the network configuration. By network configuration we mean the absolute and the relative orientations of cameras in the network assuming that their relative location is determined by either GPS or surveyed points in the 3D world. We propose a framework for auto-configuration of such a dynamic network, thereby obtaining the dynamic geometry of the network along with self-calibrating each camera in the network. By configuring such a camera network we can (i) direct cameras to follow a particular object [DDZ01], (ii) calibrate cameras so that the observations are more coordinated and perform measurements (with known scale) and possibly construct a 3-D world model [MK04, CT04], (iii) solve the camera hand-over problem i.e. establish correspondence between tracked objects in different cameras (iv) generate image/video scene mosaic (v) infer network topology [ME05], and (vi) build terrain model [CT98] or do spatial learning for navigation [YB96, Tan96].

8.1 Related Work And Our Approach

For a general configuration, each camera in the network needs to be self-calibrated by any of the method described in Part II, depending on scenario restrictions. Recently, tracking across multiple non-overlapping cameras, for video surveillance as well as topology inference, has attracted considerable amount of attention. Makris et al. [MT04] estimate camera topology from observations by assuming Gaussian transition distribution. Departures and arrivals within a chosen time window are assumed to be corresponding. Recently, Tieu et al. [TDG05] generalized the work in [MT04] to a multi-modal transition distributions, and handled correspondences explicitly. Camera connectivity is formulated in terms of statistical dependence, and uncertain correspondences are removed in a Bayesian manner. Javed et al. [JSS05] demonstrate that the brightness transfer functions from a given camera to another camera lie in a low dimensional subspace. Their method learns this subspace of mappings for each pair of cameras from the training data. Using the subspace of brightness transfer functions, the authors attempt to solve the camera hand-over problem. Kang et al. [KCM03] use an affine transform between each consecutive pair of images to stabilize moving camera sequences. A planar homography computed by point correspondences is used to register stationary and moving cameras. Zhao et al. [ZAK05] formulate tracking in a unified mixture model framework. Ground-based space-time cues are used to match trajectories of objects moving from one camera to another. It is well known that due to perspective projection the measurements made from the images do not represent metric data. Thus the obtained object trajectories and consequently the associated probabilities, used in most of the work cited above, represent projectively distorted data, unless we have a calibrated camera. For example, a person

Figure 8.1: Two cameras in a network several blocks apart from each other.

moving slowly but close to a camera induces large image motion compared to person walking at a distance with a quicker pace. Also, appearance based features exhibit undesirable results under varying lighting conditions. On the other hand, inter-camera relationships can not be correctly established unless dynamic positions and orientations between cameras are known at any point in time.

The most related work is that of Jaynes [Jay04]. Assuming a common ground plane for all cameras, relative rotation of each camera to the ground plane is computed independently. The motion trajectories of objects tracked in each camera are then reprojected on to a plane in front of the camera frame in order to compute corresponding unwarped trajectories. Camera-to-ground-plane rotation and plane-to-plane transform computed from the matched trajectories is then used to compute relative transform between a pair of cameras. This method assumes that all cameras are calibrated, requires motion trajectories on objects, and each camera is considered to be stationary looking at a common ground plane.

We present a more general solution for registering a network of disjoint cameras. We do not assume any special camera motion or known camera rotation matrix, as used by [AHR01, SH99, FK03, PKG99, Har97]. Instead of relying only on the color features for performing video surveillance or inferring network configuration, computed metric information from the calibrated cameras

can be used to determine correct correspondences. We present a novel technique to configure the network as a whole. The target is that each calibrated camera should be able to communicate its intrinsic and extrinsic parameters with other cameras in the network. We demonstrate that a (vertical) vanishing point and the knowledge of a line in a plane orthogonal to the vertical direction is sufficient to perform this task.

Our key contribution includes a method to compute the relative orientation between non-overlapping cameras using only vertical vanishing point, and a novel approach to calculate the infinite homography between a pair of cameras in the network. As an application, we apply our method to configure a Mixed Reality(MR) environment (Chapter 11).

8.2 Geometry Of Networked Cameras

Our goal in this section is to demonstrate that one can establish a common world reference frame to recover absolute camera orientations even with non-overlapping FoVs. The key to establishing a common reference frame is the fact that all cameras share the same plane at infinity and, in our case, also the same vertical vanishing point. In addition, we require a line to be visible in each image in order to completely determine the orientation between the cameras with disjoint FoV. The lines in each image need not to be parallel in the world; orthogonal lines can be used as well (explanation follows in the next subsection).

Assuming that each camera as a unit has been calibrated in the network using the method described in Part II, we would like the entire camera network to recover its own configuration. That is, each camera should learn its relative orientation with respect to every other camera.

Figure 8.2(a) shows a typical configuration of a camera network. Cameras are moving freely in

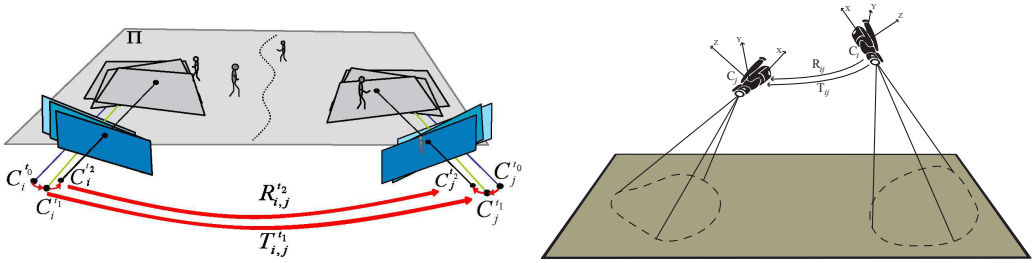


Figure 8.2: **A typical configuration** (a) *Dynamic Epipolar Geometry*: figure demonstrates a dynamic camera network where each camera is moving with respect to itself and with respect to all the cameras in the network thereby inducing a different epiploar geometry at each time instance. For a camera i at any time instance t, its center is labeled as C_i^t. The camera can be looking at a planar as well as non planar scene while translating and rotating. Each camera has an associated FoV and all the cameras in the network have disjoint FoVs. The relative orientation between cameras is denoted by $\mathbf{R_{i,j}^t}$ and the translation by $\mathbf{T_{i,j}^t}$. (b) shows an instance of the dynamic epipolar geometry. The figure contains two cameras having disjoint FoVs with some rotation and translation between each camera.

space, inducing a unique epipolar geometry at each time instance. For any camera i at time instance t, its center is labeled as C_i^t. Figure 8.3 shows a broader picture of the camera network. Each camera is mounted on a moving or a stationary platform while varying its intrinsic and extrinsic parameters. Each camera has an associated FoV and all the cameras in the network have disjoint FoVs. The relative orientation between cameras at any time instance t is denoted by $\mathbf{R_{i,j}^t}$ and the relative translation by $\mathbf{T_{i,j}^t}$. We assume that the relative translations $\mathbf{T}_{i,j}^t$ can be computed either by a set of surveyed points in the scene, or given by GPS. From here on we omit the superscript t to keep the notation simple.

8.2.1 Relative Orientation Estimation Using Vanishing Points

Vertical vanishing point ($\boldsymbol{v}_z^i$) [CT90] can be readily obtained from most naturally occurring or man-made scenes, e.g. scenes containing buildings or other structures. Similarly, people or objects

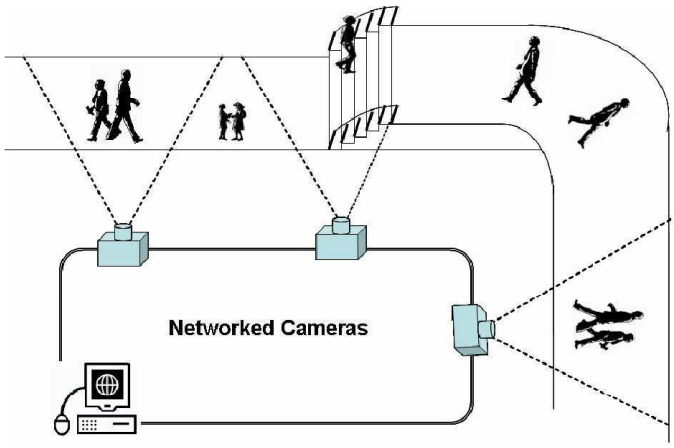


Figure 8.3: *A Network of Cameras*: The figure shows a general view of the network where each camera may be mounted on a moving platform while detecting/tracking objects.

in the FoV of each camera can be used to determine $\boldsymbol{v}_z^i$. Several researchers [LZN02, KM05] and recently we presented a method [JF06b] where motion of a tracked pedestrian is used to obtain the vertical vanishing point. For a camera i at any time instance, given a vertical vanishing point $\boldsymbol{v}_z^i$, the vanishing line $\boldsymbol{l}_\infty^i$ can be determined by using the pole-polar relationship [HZ04]:

$$\boldsymbol{l}_\infty^i = \boldsymbol{\omega}_i \boldsymbol{v}_z^i \tag{8.1}$$

$\boldsymbol{l}_\infty^i$ intersects the **IAC** $\boldsymbol{\omega}_i$ at two complex points called the circular points.

In addition, we require that a line be visible in each image. This line can lie on any plane that is orthogonal to the vertical direction, and may be specified either by the user, extracted by registering to architectural plans or maps, or determined by other vision-based methods [CRZ99, BZ99]. For example, checkered tiles on the floor, or brick lining on the wall, or other lines abundant in indoor and outdoor setting, can be used to serve our purpose. Two situations, simplified to two-image

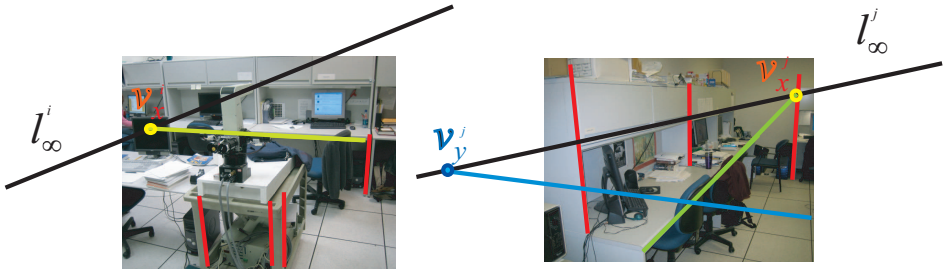


Figure 8.4: Views from two non-overlapping cameras: A pair of parallel lines intersect l_∞ at a vanishing point $\boldsymbol{v}^i_x$ in the left image and $\boldsymbol{v}^j_x$ in the right image, respectively. Above, the vanishing line for each view is drawn in black while the parallel lines, an example of case 1, are drawn in green. The green line in each view intersect the vanishing line at a point. This point is the corresponding vanishing point between the two views. As an example for case 2, the blue line in right image is orthogonal to the green line in the left image. Red color is selected to denote lines used for estimating the vertical vanishing point.

cases, can occur with such a configuration, as shown in Figure 8.4:

1 *When the visible lines are parallel to each other in world*: In this case, intersection of the imaged line, $\boldsymbol{l}_i$, with the $\boldsymbol{l}^i_\infty$ yields a vanishing point orthogonal to $\boldsymbol{v}^i_z$:

$$\boldsymbol{v}^i_x \sim \boldsymbol{l}_i \times \boldsymbol{l}^i_\infty \tag{8.2}$$

where $\boldsymbol{v}^i_x$, without loss of generality, is taken as the vanishing point along the x-axis for an image i.

2 *When the visible lines are perpendicular to each other in world*: The intersection of the imaged line with the line at infinity yields vanishing point in each image that represent mutually orthogonal directions in the world. In addition to Eq. 8.2, for the second image (j) we get:

$$\boldsymbol{v}^j_y \sim \boldsymbol{l}_j \times \boldsymbol{l}^j_\infty \tag{8.3}$$

As an example for case 1 (cf. Figure 8.4), note that $\boldsymbol{l}_i$ (i.e. green line) is visible in the left image and $\boldsymbol{l}_j$ (i.e. green line) is visible in the right image only (since we are dealing with non-overlapping FOV). But since $\boldsymbol{l}_i$ and $\boldsymbol{l}_j$ are parallel in the world, they intersect at $\boldsymbol{v}_x^i$ and $\boldsymbol{v}_x^j$, respectively. These two points are the corresponding vanishing points in the two views. As an example for case 2, the blue line in right image is orthogonal to the green line ($\boldsymbol{l}_i$) in the left image, hence the vanishing point $\boldsymbol{v}_y^j$ is orthogonal to the vanishing point $\boldsymbol{v}_x^i$.

Absolute rotation w.r.t. the world reference frame: Given two vanishing points $\boldsymbol{v}_x^i$ and $\boldsymbol{v}_z^i$ from each view of a single camera, the rotation of camera i with respect to a common world coordinate system can be computed as:

$$\boldsymbol{r}_3 = \pm\frac{\mathbf{K_i^{-1}}\boldsymbol{v}_\mathbf{z}^\mathbf{i}}{\|\mathbf{K_i^{-1}}\boldsymbol{v}_\mathbf{z}^\mathbf{i}\|}, \quad \boldsymbol{r}_1 = \pm\frac{\mathbf{K_i^{-1}}\boldsymbol{v}_\mathbf{x}^\mathbf{i}}{\|\mathbf{K_i^{-1}}\boldsymbol{v}_\mathbf{x}^\mathbf{i}\|}, \quad \boldsymbol{r}_2 = \frac{\boldsymbol{r}_3 \times \boldsymbol{r}_1}{\|\boldsymbol{r}_3 \times \boldsymbol{r}_1\|}, \tag{8.4}$$

where $\boldsymbol{r}_1, \boldsymbol{r}_2$ and $\boldsymbol{r}_3$ represent three columns of the rotation matrix. The sign ambiguity can be resolved by the cheirality constraint [HZ04] or by known world information, like the maximum rotation possible for the camera.

Relative orientation is obtained from the obtained absolute orientation for each camera view. Care must be taken in using Eq. (8.2) and Eq. (8.3). Based on the obtained vanishing points ($\boldsymbol{v}_y$ or $\boldsymbol{v}_x$), appropriate equations from Eq. (8.4) must be selected for determining the absolute orientations.

8.2.2 Alternate Solution: Using Infinite Homography Relationship

An alternate solution is to use the infinite homography. A rotating and/or a zooming camera induces an infinite homography $\mathbf{H}_{\mathbf{i,j}}^{\infty}$, which relates two cameras i and j via the plane at infinity

(Π_∞). For such a case, infinite homography may be calculated directly from point or line correspondences using Eq. (2.11) using the method described in [AHR01] (see [ZH94, HJL89] for more on pose estimation). But for a camera undergoing a general motion the correspondences can not be obtained as the FoV is disjoint. However, by determining points or lines lying on Π_∞ it is possible to estimate $\mathbf{H}^\infty_{\mathbf{i,j}}$ from such ideal point/line correspondences. The idea is as follows: *Eq. (2.11) should be simplified so that instead of solving for* $\mathbf{H}^\infty_{\mathbf{i,j}}$*, we only solve for the relative rotation matrix* $\mathbf{R}_{\mathbf{i,j}}$ *between two cameras* i *and* j.

Any point, let us say $\boldsymbol{v}^i_x$, lying on $\boldsymbol{l}^i_\infty$, for a camera i satisfies the orthogonality constraint ${\boldsymbol{v}^i_x}^T \boldsymbol{\omega}_i \boldsymbol{v}^i_z = 0$. Thus $\boldsymbol{v}^i_x$ is chosen as a vanishing point orthogonal to $\boldsymbol{v}^i_z$. Any such point in camera i is transformed via Π_∞ to a point $\boldsymbol{v}^j_x$ on another camera j as:

$$\mathbf{H}^\infty_{\mathbf{i,j}} \boldsymbol{v}^{\mathbf{i}}_{\mathbf{x}} \sim \boldsymbol{v}^{\mathbf{j}}_{\mathbf{x}}, \tag{8.5}$$

and similarly

$$\mathbf{H}^\infty_{\mathbf{i,j}} \boldsymbol{v}^{\mathbf{i}}_{\mathbf{z}} \sim \boldsymbol{v}^{\mathbf{j}}_{\mathbf{z}}, \tag{8.6}$$

where $\mathbf{H}^\infty_{\mathbf{i,j}}$ is the infinite homography between camera i and j; and $\boldsymbol{v}^i_x$ is obtained from the method described in the last subsection.

We need more constraints if we are to solve for a general $\mathbf{H}^\infty_{\mathbf{i,j}}$ as it contains 8 unknowns (nine minus the scale). However, we only need to compute the relative orientation $\mathbf{R}_{\mathbf{i,j}}$ between each camera since the calibration matrix for each camera is already computed. Therefore, Eq.(8.6) can be simplified to:

$$\begin{aligned} \mathbf{K_j R_{i,j} K_i^{-1} v_z^i} &\sim \boldsymbol{v}_z^j \\ or\ \mathbf{R_{i,j} r_3^i} &\sim \mathbf{r_3^j} \end{aligned} \tag{8.7}$$

where $\mathbf{r_3^s} = \frac{\mathbf{K_s^{-1} v_z^s}}{\|\mathbf{K_s^{-1} v_z^s}\|}$ with $s = \{i, j\}$. The third column of the rotation matrix thus computed can provide two unknown angles for each camera as follows.

$$\theta_y^s = \sin^{-1}(\mathbf{r}^{\mathbf{s}}_{\mathbf{3}_{(1)}}) \textbf{ and } \theta_{\mathbf{x}}^{\mathbf{s}} = \frac{\sin^{-1}(\mathbf{r}^{\mathbf{s}}_{\mathbf{3}_{(2)}})}{\cos(\theta_{\mathbf{y}}^{\mathbf{s}})}$$

Eq.(8.5) is also simplified to:

$$\mathbf{R_{i,j} K_i^{-1} v_x^i} \sim \mathbf{K_j^{-1} v_x^j} \tag{8.8}$$

where $\mathbf{K_i}$ and $\mathbf{K_j}$ are the computed calibration matrices for camera i and j, respectively.

The third angle, θ_z^s for each camera need not be computed explicitly in order to get the relative rotation between cameras. The relative rotation matrix is simplified to,

$$\begin{aligned} \mathbf{R_{i,j}} &= \mathbf{R_{x_j} R_{y_j} R_{z_j} R_{z_i}^T R_{y_i}^T R_{x_i}^T} \\ or\ \mathbf{R_{i,j}} &= \mathbf{R_{x_j} R_{y_j} R_{z_{ij}} R_{y_i}^T R_{x_i}^T} \end{aligned} \tag{8.9}$$

where $\mathbf{R_{x_i} R_{y_i} R_{z_i}}$ represents rotation around x-axis, y-axis and z-axis, respectively, for a camera

i.

Replacing $sine$ and $cosine$ with unknown x and y respectively, we solve Eq. (8.9) linearly w.r.t. x and y. Scale ambiguity is removed by taking the cross ratio of the left and right hand side of Eq. (8.9) while substituting $\mathbf{R_{x_i}}, \mathbf{R_{y_i}}, \mathbf{R^T_{y_j}}$ and $\mathbf{R^T_{x_j}}$ with the angles calculated above. Singular Value Decomposition is applied to obtain the unknown relative angle $\theta_{z_{ij}}$. Knowing all the angles allows us to recover relative orientation between each pair of cameras in the network.

The two methods described above require same information i.e. $\boldsymbol{v}_x$ and $\boldsymbol{v}_z$, and provide similar results. The methods are indeed alternate: in first method the relative camera orientations is obtained from absolute camera orientation whereas in the second method we directly solve for the relative rotation matrix $\mathbf{R_{i,j}}$. For experimental validation, the method described in Subsection 8.2.1 is chosen due to its simplicity.

8.3 Singularities

The camera or network calibration algorithms, like any other algorithms, have *singularities*. This is also often referred to as *degenerate configurations* by some researchers. It is important to be aware of such situations in order to get an insight into the problem and obtain reliable results.

By degenerate configurations we mean situations where a particular camera motion does not result in any constraint on the camera intrinsics. For example, [WKS04] shows that it is possible to obtain a closed-form solution for the only unknown f for a fronto-parallel or panning configuration of a rotating camera. But it is not possible to obtain a closed-form solution for λ when f, λ are unknown parameters for a panning camera. Note that for rotating fixed cameras or freely moving cameras it is always favorable to have large rotations. If there is no rotation between views then the

Kruppa equations do not provide any constraint on ω_i^* as for such case $\mathbf{F} = [\mathbf{e}']_\times$ and the equation is reduced to $[\mathbf{e}']_\times \boldsymbol{\omega}_\mathbf{i}^* [\mathbf{e}']_\times \sim [\mathbf{e}']_\times \boldsymbol{\omega}_\mathbf{i}^* [\mathbf{e}']_\times$.

It is beyond the scope of the current work to expound on all degenerate configurations for self-calibration. Therefore, we only focus on critical configuration for one (f) or two parameters (f, λ) estimations. Zisserman et al. [ZLA98] examine ambiguities arising from motions with single direction of the rotation axis when all the parameters are unknown but constant. When the axis of rotation is perpendicular to the image plane, specified skew, principal point and aspect ration are not sufficient to remove the ambiguity. For variable focal length cameras, [Stu99] derives conditions under which it is not possible to calculate the value of f. He shows that critical configuration arises when: optical centers of stereo cameras are collinear, optical centers lie on ellipse/hyperbola pair, or when the optical axes are parallel. Kahl et al. [KTA00] generalize [Stu99] to include cases when other parameters vary as well and show that criticality is independent of the values of the intrinsic camera parameters. For methods based on Kruppa's equations, when only f is unknown, motions are critical *iff* the optical axes of the two cameras intersect or when the optical axes planes are orthogonal.

We now consider critical configuration for the proposed method. We showed that it is possible to determine absoulte/relative rotations for cameras comprising a network and that only one vanishing point is required. Critical configuration occurs only when we are unable to determine the vanishing point for image sequences. Projection of the vertical vanishing point is given as:

$$\boldsymbol{v}_z \sim \begin{bmatrix} \lambda f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_1 & r_2 & r_3 & t_x \\ r_4 & r_5 & r_6 & t_y \\ r_7 & r_8 & r_9 & t_z \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}^T$$

$$or\ \boldsymbol{v}_z \sim \begin{bmatrix} fr_3 & fr_6 & r_9 \end{bmatrix}^T, \tag{8.10}$$

assuming known aspect ratio (λ). Degenerate configuration occurs when:

1. $r_9 = \cos\theta_x \cos\theta_y = 0$: This happens when either $\theta_x = 90°$ and $\theta_y = 90°$. This is the case when our camera viewing direction is perpendicular to the vertical direction (z).

2. $\boldsymbol{v}_z = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$ i.e. $\theta_x = 90°$ and $\theta_y = 0°$: This situation occurs when camera is located on the vertical axis with viewing direction perpendicular to the $x - y$ plane. In this case $\boldsymbol{v}_z$ coincides with the principal point (since our principal point is at $(0, 0)$).

3. $f \longrightarrow \infty$: The camera becomes an instance of affine camera. In such a configuration it is not possible to measure any vanishing points as parallel lines are invariant under affine transformations. An example would be that of distant aerial imagery.

Although of significant theoretical importance, the above cases do not commonly occur in general settings.

8.4 Results

In this section we present some experimental results with synthetic as well as real data.

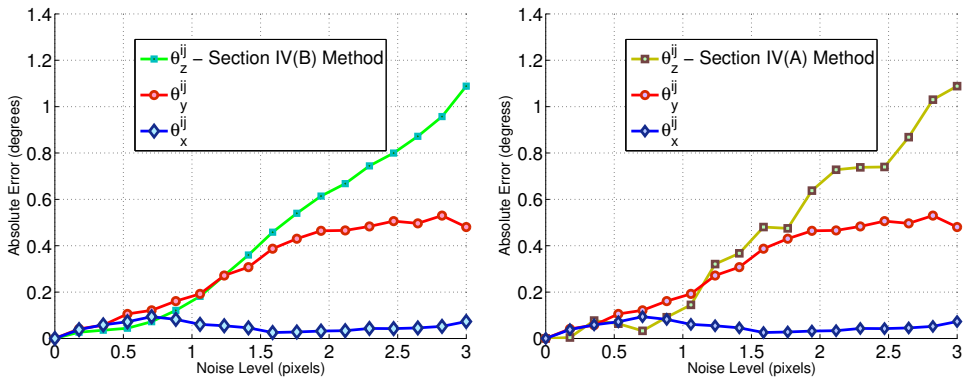


Figure 8.5: Performance of network configuration method VS. Noise level in pixels: **Left** - Absolute error in angles obtained by using the method described in Section 8.2.2. **Right** - Absolute error in errors obtained from the method described in Section 8.2.1. Notice that while the curve for θ_z is somewhat different, the curve for the other two angles is exactly the same. This is due to the fact that we are using the same vertical vanishing point to estimate θ_x and θ_y for both the methods.

Synthetic Data: We rigourously test the proposed method for estimating the relative angles between different cameras. Hundred vertical lines of random length and random location are generated to approximate the vertical vanishing points. Similarly, we chose hundred points (arbitrary number) to represent the line (l_i) which is visible in image i (see Section 8.2). We gradually add a Gaussian noise with $\mu = 0$ and $\sigma \leq 3$ to the data points making up the vertical lines. Vertical vanishing point is obtained using *SVD* on the vertical lines. Similarly, *SVD* is applied to the points making up $\boldsymbol{l}_i$ to obtain the point of intersection of $\boldsymbol{l}_i$ and $\boldsymbol{l}_i^{\infty}$. Translation and rotation are selected subjectively to avoid degenerate configurations. While varying the noise from 0.1 to 3 pixel level, we perform 1000 independent trials for each noise level, the results are shown in Figure 8.5. The absolute error is found to be less than 1.2° for the maximum noise of 3 pixel in our tests using both the methods described in Section 8.2.1 and Section 8.2.2 as shown in Figure 8.5.

Real Data-Using PTZ Camera for ground-truth: In order to obtain ground truth for relative camera rotations, we employ a SONY® SNC-RZ30N PTZ cameras. The purpose of this demon-

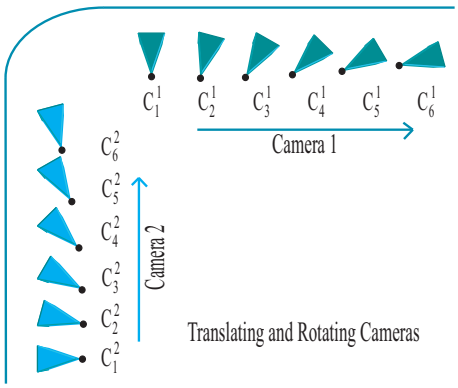


Figure 8.6: Outline map of the test sequence setup. Two cameras, initially with orthogonal FoV, are translated and rotated. A camera is represented by $\mathcal{C}_{\mathbf{i}}^{\mathbf{k}}$, where k is a camera label and i is a frame or an instance number. See text for more details.

Camera # 1-Estimated f(left to right): $1091.14, 1135.35, 1155.76, 1162.52, 1113.01, 1124.15$

Camera # 2-Estimated f(left to right): $1121.14, 1124.35, 1103.436, 1181.191, 1190.05, 1171.96$

Figure 8.7: Some images from a test sequence using two cameras. The cameras are translated as well as rotated. The green line indicate the knowledge of a line in world. In this particular case, the line in one camera is orthogonal to the corresponding line in the second camera.

stration is to verify the accuracy and applicability of the proposed method. The outline of the test sequence is shown in Figure 8.6. Two PTZ cameras are used for the demonstration. The cameras are represented by $\mathcal{C}_{\mathbf{i}}^{\mathbf{k}}$, where k is a camera label and i is a frame number.

Some of the images from the test sequence are shown in Figure 8.7. The top row of Figure depicts images from camera 1, while the bottom from camera 2. The ground truth rotation for the shown images is known by controlling the PTZ cameras. Self-calibration is performed on the sequence and the results are shown in Figure 8.7. The fundamental matrix is computed between

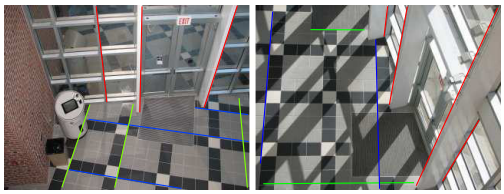
Table 8.1: **Ground Truth θ_z Vs. Estimated $\hat{\theta}_z$:** Column represent Camera # 1 denoted by $\mathcal{C}_{\mathbf{i}}^{\mathbf{1}}$, and rows represent Camera # 2 denoted by $\mathcal{C}_{\mathbf{i}}^{\mathbf{2}}$. Since the orientation between cameras is symmetric(only a sign change), values of the lower left triangle of the table are denoted by *.

CAMERA # 1

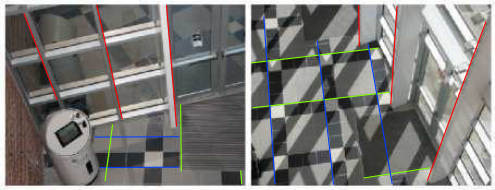
CAMERA # 2	$\mathcal{C}_{\mathbf{1}}^{\mathbf{1}}$ $(\theta_{\mathbf{z}};\hat{\theta}_{\mathbf{z}})$	$\mathcal{C}_{\mathbf{2}}^{\mathbf{1}}$ $(\theta_{\mathbf{z}};\hat{\theta}_{\mathbf{z}})$	$\mathcal{C}_{\mathbf{3}}^{\mathbf{1}}$ $(\theta_{\mathbf{z}};\hat{\theta}_{\mathbf{z}})$	$\mathcal{C}_{\mathbf{4}}^{\mathbf{1}}$ $(\theta_{\mathbf{z}};\hat{\theta}_{\mathbf{z}})$	$\mathcal{C}_{\mathbf{5}}^{\mathbf{1}}$ $(\theta_{\mathbf{z}};\hat{\theta}_{\mathbf{z}})$	$\mathcal{C}_{\mathbf{6}}^{\mathbf{1}}$ $(\theta_{\mathbf{z}};\hat{\theta}_{\mathbf{z}})$
$\mathcal{C}_{\mathbf{1}}^{\mathbf{2}}$	$(90°; 90.66°)$	$(105°; 100.59°)$	$(120°; 117.6°)$	$(135°; 132.22°)$	$(150°; 150.94°)$	$(165°; 157.56°)$
$\mathcal{C}_{\mathbf{2}}^{\mathbf{2}}$	*	$(90°; 96.91°)$	$(105°; 113.92°)$	$(120°; 124.54°)$	$(135°; 137.26°)$	$(150°; 153.89°)$
$\mathcal{C}_{\mathbf{3}}^{\mathbf{2}}$	*	*	$(90°; 92.69°)$	$(105°; 108.29°)$	$(120°; 123.02°)$	$(135°; 133.64°)$
$\mathcal{C}_{\mathbf{4}}^{\mathbf{2}}$	*	*	*	$(90°; 96.56°)$	$(105°; 111.91°)$	$(120°; 121.53°)$
$\mathcal{C}_{\mathbf{5}}^{\mathbf{2}}$	*	*	*	*	$(90°; 88.26°)$	$(105°; 103.82°)$
$\mathcal{C}_{\mathbf{6}}^{\mathbf{2}}$	*	*	*	*	*	$(90°; 89.64°)$

consecutive frames obtained from each single camera to determine the calibration matrix. The computed fundamental matrix is decomposed to obtain the relative translation and relative rotation between the two frames. The technique presented by [Low04] automatically detects scene features that can be used to robustly compute the fundamental matrix. If the scene contains moving objects, the vertical vanishing point can be obtained automatically, as demonstrated by [KM05, JF06b] and Lv et al. [LZN02]. As reported by Zhang [Zha00], the mean of the estimated focal length is taken as the ground truth and the standard deviation as a measure of uncertainty in the results. Thus, with a low standard deviation $\sigma = 32.05$, f is determined to be 1139.50.

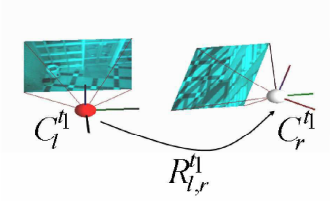
As is evident from Sections 8.2, the most difficult angle to obtain is the *relative* θ_z (we omit superscripts ij), as it can not be obtained from $\boldsymbol{v}_z$ alone. Therefore, we set up the experiment to vary θ_z. Initially, the two cameras are separated by an angle of $\theta_z = 90°$(pan angle) i.e. for $\mathcal{C}_{\mathbf{1}}^{\mathbf{1}}$ vs. $\mathcal{C}_{\mathbf{1}}^{\mathbf{2}}$ (see Figure 8.6). While translating, the cameras are rotated by some known angle. The images shown in Figure 8.7 are selected such that the rotation angle between different instances/frames is



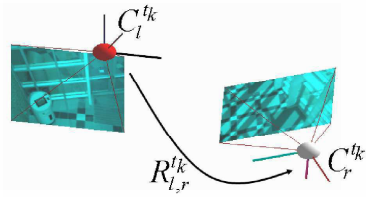
(a) A view from two neighboring cameras



(b) A view from two neighboring cameras



(c) Recovered 3D Geometry of cameras



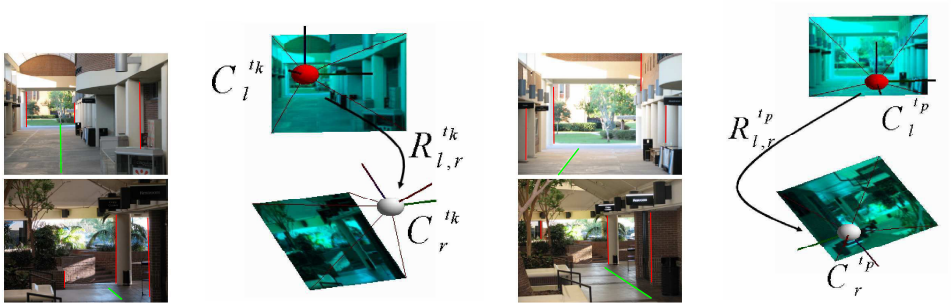
(d) Recovered 3D Geometry of cameras

Figure 8.8: (a) and (b) are views taken from two disjoint FoV cameras looking at a lobby entrance. The two cameras are free to rotating and translating. The 3D rendering in (c) and (d) demonstrates the computed dynamic geometry of the network. This network geometry is unique at each instance of time.

$\theta_z = 15°$(an arbitrary angle). For example, the difference between the orientation of $\mathcal{C}_\mathbf{2}^\mathbf{1}$ and $\mathcal{C}_\mathbf{1}^\mathbf{1}$ is $\theta_z = 15°$. After self calibration, the method described in Section 8.2.1 is used to obtain the relative camera orientation. The obtained results are presented in Table 8.1.

Table 8.1 compares the obtained $\hat{\theta}_z$ with the ground truth θ_z. Each column of the table represents an instance from camera 1, while each row represents an instance from camera two. For example, intersection of row 3 and column 3 represented the orientation between 3^{rd} frame/instance of each camera $\mathcal{C}_\mathbf{3}^\mathbf{1}$ and $\mathcal{C}_\mathbf{3}^\mathbf{2}$. Since the relative rotation between two cameras is symmetric, we denote the lower left triangle of the table by *. The mean error in estimated angle and the standard deviation is found to be $3.53°$ and $2.5°$, respectively, which is very low.

Errors can be attributed to many factors. Main source of error in a PTZ camera is the radial distortion, as visible in the test images. Another important factor is the inherent error present in localizing pixels for determining vanishing points.



(a) Neighboring Cameras (b) Recovered 3D Geometry (c) Neighboring Cameras (d) Recovered 3D Geometry

Figure 8.9: (a) and (c) are instances from a data sequence looking from inside a hallway. The two cameras have disjoint FoV as they are looking in almost opposite direction. At each time instance the camera network has a unique geometry. The 3D rendering in (b) and (d) only demonstrates the computed dynamic geometry of the network and the images inside the rendering do not represent registered images.

Real Data-Moving Cameras: For further experimental validation, two sequences of real data were obtained from two pairs of moving cameras fitted with GPS receivers. GPS data is required to pinpoint exact camera location allowing us to compute the translation between each camera. Unlike the results demonstrated in the previous subsection, the ground-truth is not available for this experimentation and visual inspection is the only goodness of measure.

The data was collected over a long period of time and two instance from the first sequence are shown in Figure 8.8. The left camera is denoted by its center C_l and the right camera is denoted by C_r, omitting the superscript used to indicate different time instances. Using computed vanishing points, inter-camera rotation matrix $\mathbf{R_{l,r}}$ is computed, which is then used to compute the $\mathbf{H_{i,j}^{\infty}}$. The resulting angles obtained are presented in Table 8.2 (row 1 and 2). Figure 8.8(c) and Figure 8.8(d) render the recovered network geometry, which is intended to help visualize the obtained results; and the rendered scene images are only texture maps and do not depict the actual image registration.

Table 8.2: External Parameters obtained from test dataset.

Views	Recovered Relative Rotation $(\theta_x^{ij}, \theta_y^{ij}, \theta_z^{ij})$ in degrees
Figure 8.8(a)	$(12.84, 11.56, 44.99)$
Figure 8.8(b)	$(13.58, 13.51, 134.99)$
Figure 8.9(a)	$(-154.25, -1.04, 45.04)$
Figure 8.9(c)	$(-176.42, -1.7, 94.96)$
Figure 6.6	$(9.53, 3.748, -86.22)$

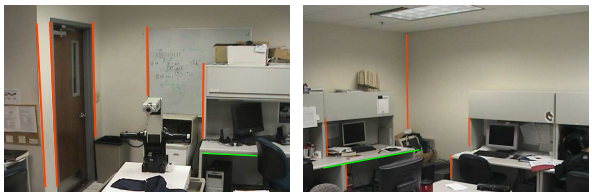
The second data sequence contains cameras looking in opposite directions in a hallway. Instances of this data sequence are shown in Figure 8.9(a) and Figure 8.9(c). The cameras are in continuous motion at every time instance; the network geometry is rendered in Figure 8.9(b) and Figure 8.9(d). Generally, scenes containing abundant architectural structures are well desirable if we are to compute the vanishing points.

The rotation angles calculated from the second data sets are presented in row 3 and 4 of table 8.2. Since the cameras are looking in opposite direction, θ_x is close to -180^o.

The errors could be attributed to several sources. Besides noise, non-linear distortion and imprecision of the extracted features, one source is the causal experimental setup using minimal information, which is deliberately targeted for a wide spectrum of applications. Despite all these factors, our experiments indicate that the proposed algorithm provides good results.

SPECIAL CASE - PURE ROTATION: The proposed self-calibration method (Chapter 6) is based on the Kruppa equations. However, these equations rely on accurate estimation of the fundamental matrix. For a special case when no translation occurs, the fundamental matrix degenerates and our self-calibration technique would not be applicable.

In order to self-calibrate a pure rotating camera, without loss of generality, the projection matrix



(a) Ground Truth Angles $(5°, 0°, 65°)$: Calculated Angles $(5.41°, 0.261°, 64.04°)$

(b) Ground Truth Angles $(11°, 0°, 55°)$: Calculated Angles $(16.81°, 1.82°, 58.95°)$

(c) Ground Truth Angles $(0°, 0°, 80°)$: Calculated Angles $(1.08°, 1.79°, 78.15°)$

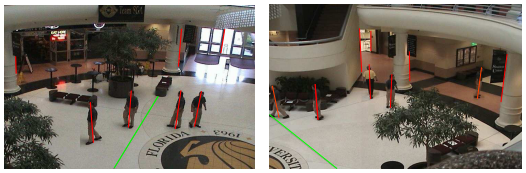
(d) Ground Truth Angles $(10°, 0°, 45°)$: Calculated Angles $(15.04°, 0.73°, 44.8184°)$

Figure 8.10: Four of the many test sequences taken from a PTZ camera. The ground truth relative rotation angles are compared to the obtained rotation angles. Green line indicates a common lines parallel in real world) while the lines used to compute the vertical vanishing point are drawn in red.

for the first view can be formulated as $\mathbf{P_i} = \mathbf{K_i}[\mathbf{R_i}|\mathbf{0}]$, where the translation $\mathbf{t_i} = -\mathbf{R_i}\mathbf{C} = \mathbf{0}$. The projection of any scene point $\mathbf{X}$ onto an image plane is expressed as $\mathbf{x} = \mathbf{K_i}\mathbf{R_i}\mathbf{X}$.

For a scene point projected onto two different images, a 2D projective transformation $\mathbf{H_{i,j}}$ relates the corresponding points as $\mathbf{x_j} = \mathbf{H_{i,j}}\mathbf{x_i}$, where $\mathbf{H_{i,j}} = \mathbf{K_j}\mathbf{R_{i,j}}\mathbf{K_i^{-1}}$. This 2D projective transformation maybe calculated directly from point or line correspondences between images.

Using the property $\mathbf{R} = \mathbf{R^{-T}}$, the definition of $\mathbf{H_{i,j}}$ leads to some constraints on the **IAC**:



Ground Truth $(0°, 0°, 55°)$: Calculated Angles $(1.08°, 3.78°, 54.49°)$

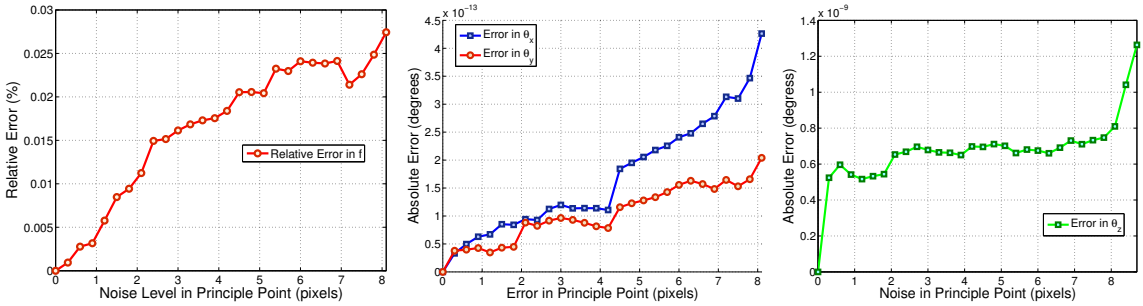
Figure 8.11: A test sequence taken from a PTZ camera with people walking. The ground truth relative rotation angles are compared to the obtained rotation angles. See text for more details.

$$(\mathbf{K_j K_j^T}) = \mathbf{H_{i,j}}(\mathbf{K_i K_i^T})\mathbf{H_{i,j}^T} \tag{8.11}$$

where $\boldsymbol{\omega}_j = (\mathbf{K_j K_j^T})^{-1}$ and $\boldsymbol{\omega}_i = (\mathbf{K_i K_i^T})^{-1}$. Linear constraints on the unknowns of $\boldsymbol{\omega}$ are obtained by further assuming zero skew and unit aspect ratio. See [AHR01], [BR97] for further details and discussions about calibrating rotating and zooming cameras.

Some test sequences are performed for this special case of camera motion. Four of the test cases are shown in Figure 8.10. The ground-truth relative rotation angles are compared to the obtained relative rotation angles. Two PTZ cameras are used for this sequence. The lines which are parallel in the world are drawn in green, while the lines used for the vertical vanishing point are drawn in red. After self-calibrating each rotating camera, as described above, the angles are estimated as described in Section 8.2. The estimated rotation angles are shown below the figure. Another set of a test sequence captured with a PTZ camera is shown in Figure 8.11. Here pedestrians are walking in the FoV of each camera. Different frames are supperimposed on one image as shown in the Figure. The method proposed by Lv et al. [LZN02] is used to extract the vertical vanishing point. The results obtained are very encouraging and close to the ground truth.

Effect of Principal Point on Camera Parameters: The proposed method assumes the princi-



(a)Error in f (b)Absolute error in θ_x, θ_y (c) Absolute Error in θ_z

Figure 8.12: Intrinsic and extrinsic camera parameters when the principal points is not exactly at the center of the image.

pal point is located at the center of an image. The image is then transformed so that the principal point lies at $(0, 0)$. Although this is a very reasonable assumption for currently available cameras, we analyze the effect of deviation from this assumption on both intrinsic and extrinsic camera parameters.

A random Gaussian noise of $\mu = 0$ and $\sigma = 9$ pixel was introduced to a point cloud containing 250 points. The error curves for the obtained focal length f, θ_x, θ_y and θ_z are shown in Figure 8.12. The error curves for all the estimated parameters are near linear. For a displacement of 9 pixels off the image center, the relative error in f is close to 0.03% (cf. Figure 8.12). Similarly, the absolute errors for θ_x, θ_y and θ_z are also very small, see Figure 8.12(b) and Figure 8.12(c), respectively. Thus a displaced principal point does not significantly affect the proposed method.

8.5 Conclusion

We have successfully demonstrated a novel approach to recover dynamic network geometry. Each camera, having a disjoint FoV, is assumed to undergo a general motion. Such a network could be, for instance, deployed for surveillance applications comprising of both stationary PTZ cameras

and cameras mounted on a roaming security or reconnaissance vehicles (e.g. [CM03]). Another application could be in an urban battlefield setting with soldiers carrying head mounted cameras.

Our contribution includes (i) computing the relative rotation matrix between N cameras using only vertical vanishing point, and (ii) calculating the $\mathbf{H}_{\mathbf{i,j}}^{\infty}$ for non-over lapping cameras and using it to obtain absolute rotation of each camera with respect to a common world coordinate system without overlapping FoV. We have successfully demonstrated the proposed method on several sequences and discussed possible degenerate configurations. The proposed network calibration technique is tested on synthetic as well as on real data. Encouraging results indicate the applicability of the proposed system.

CHAPTER 9
EUCLIDEAN PATH MODELING

We consider the problem of monitoring an area of interest, e.g. a building entrance, parking lot, port facility, an embassy, or an airport lobby, using stationary cameras. Our goal is to model the behavior of objects of interest, e.g. cars or pedestrians, with the intent to cover as large areas as possible by generally deploying non-overlapping cameras. In path modeling [GSR98, JH95, JJS04] for surveillance, the goal is to build a system that, once given an acceptable set of trajectories of objects in a scene, is able to learn the routes or paths most commonly taken by objects in order to classify incoming trajectories as conforming to the model or as unusual and anomalous.

The definition of an unusual behavior might be different for different applications. For example, a person walking in a region not used by most people, a car following a zigzag path or a person running in a region where most people simply walk. A *path* or *route* can be defined as any established line of travel or access. This is the region that is most used by the objects. *Trajectory* can be defined as a path followed by an object moving through the space. Most objects tend to follow a common trajectory while entering or exiting a scene due to presence of pavements, benches, or designated pathways. Our approach can model the usual trajectories of the object and perform measurements to indicate atypical trajectories that might call for further investigation through any higher level event recognition. Thus, given an unusual or anomalous behavior, we are able to distinguish it from acceptable ones. Moreover, as common pathways are detected by clustering the trajectories, we can efficiently assign detected trajectory to its associated path model. Hence, the vision system needs only to store the path label and the object labels instead of the whole trajectory set, resulting in a significant compression for storing surveillance data.

It is, however, known that due to perspective projection the measurements made from the images do not represent Euclidean information. Thus the obtained object trajectories and consequently the associated probabilities represent projectively distorted data, unless we have a calibrated camera. This is evident from simple observations: an object grows larger and moves faster as it approaches the camera center, or two objects moving in parallel directions seem to converge at a point in the image plane. Or, for example, a person walking at a distance from a camera will be in the field of view for a longer period of time compared to a person walking very close to the camera. Similarly, for a person walking towards a camera, the obtained trajectory contains a fewer number of overlapping data points and it is not possible to obtain accurate object motion from such a trajectory. The projective camera thus makes it difficult to characterize object characteristics and behaviors - in terms of their sizes, motion, length ratios and so on - unless camera is calibrated, in which case one can perform Euclidean measurements directly from images. For this purpose, we use the method described in Chapter 5.

In a nutshell, this chapter addresses a comprehensive set of problems for building a path modeling system, by proposing novel methods to use the calibrated camera to (i) perform metric rectification of the input sequence, (ii) register the sequence to the aerial imagery, (iii) obtain metric information about the objects from the rectified and registered images, and hence (iv) build Euclidean path models to monitor and characterize behavior of the objects by observing and performing measurements on trajectories. Remainder of chapter is organized accordingly.

9.1 Related Work

We divide the task of path modeling for surveillance in a single camera into three steps. The first step involves detecting and tracking objects in the video frames. Through this process, one can extract image plane trajectories of moving objects, which provide projectively distorted 2D representation of the true path in the 3D scene. In the second step, projective distortions are removed from the extracted trajectories to provide a Euclidean model of the path in the 3D space. Finally, a scene path model is built, whereby anomalous behaviors are detected by matching incoming trajectories to the model path for the area under surveillance. The system is able to log the behavior of an object from the moment it enters the camera's field of view until it exits, and enables the user to determine its conformity to the path model.

The first step of tracking is essentially a correspondence problem and is not the primary focus of this section; correspondence needs to be established between an object seen in the current frame and those seen in previous frames. Tracking is a widely studied problem in computer vision, and many suitable trackers exist for our purpose [CRM03, SM00, Ver99, KCM03, KS03]. We used the tracker presented by Javed et al. [JS02] to validate our method.

The second step, i.e. removal of the projective distortion, is very essential. As argued above, in order to obtain undistorted and real world information from any video sequence, the camera needs to be calibrated. Calibration is a necessary process in computer vision in order to obtain Euclidean information about the scene (up to a global scale), and to determine the rigid camera motion. We used the camera calibration methods based on pedestrians as described in Chapter 5.

Given a calibrated camera, object trajectories can be metric rectified. We can, thus, construct

a path model for the scene, incorporating various metric characteristics such as curvature and velocity. Although path modeling is a relatively new problem, we briefly survey some of the related work. Grimson et al. [GSR98] use a distributed system of cameras to cover a scene, and employ an adaptive tracker to detect moving objects. A set of parameters for each detected object are recorded, like the position, direction of motion, velocity, size, and aspect ratio of each connected region. Tracked patterns (e.g. aspect ratio of a tracked object) are used to classify objects or actions. Tracks are clustered using spatial features based on the vector quantization approach. Once these clusters are obtained the unusual activities are detected by matching incoming trajectories to these clusters. Thus, unusual activities are outliers in the clustered distributions. Boyd et al. [BMV99] demonstrate the use of network tomography for statistical tracking of activities in a video sequence. The method estimates the number of trips made from one region to another based on the inter-region boundary traffic counts accumulated over time. It does not track an object through the scene but only logs the event when an object crosses a boundary. The method only determines the mean traffic intensities based on the calculated statistics and no information is given about trajectories. Johnson et al. [JH95] use a neural network to model the trajectory distribution for event recognition and prediction. Recently, [MT04], and [TDG05] proposed methods for determining topology of a multi-camera network, and [WP05] used the 3D structure tensor for representing global patterns of local motion.

The most related work to ours is that of Makris and Ellis [ME02], where they develop a spatial model to represent the routes in an image. Once a trajectory of a moving object is obtained, it is matched with routes already existing in a database using a simple distance measure. If a match is found, the existing route is updated by a weight update function; otherwise a new route is created

for this new trajectory having some initial weight. Spatially proximal routes are merged together and a graph representation of the scene is generated. One limitation of this approach is that only spatial information is used for trajectory clustering and behavior recognition. The system cannot distinguish between a person walking and a person lingering around, or between a running and a walking person, since their models and measurements are not Euclidean. There exist no stopping criteria for merging of routes.

Our approach provides a Euclidean path modeling based on calibrated measurements. We then propose a multi-feature path modeling method that allows us to discriminate between trajectories with confidence. Innovative use of normalized-cuts makes possible to employ an unsupervised training phase for path modeling. Unlike existing methods, we not only look at the spatial information, but also the velocity and the curvature characteristics of trajectories. We test our system on real-world sequences with pedestrians passing through

9.2 Training Phase - Camera Calibration & Trajectory Rectification

Our framework is divided into two phases: the training phase and the testing phase. During training phase, our goal is to *first* used the calibrated camera to metric rectify the extracted object trajectories. *Second*, to cluster the input trajectories and build a model based on our features (Section 9.2.1). Once we have our path model, we can test the incoming trajectories and check for conforming behavior (as described in Section 9.3).

Trajectory And Image Rectification Once the camera is calibrated, the object trajectories obtained in the training phase can be metric rectified. As argued above, metric rectified data presents a more accurate picture of the original data. The line at infinity $\mathbf{l}_\infty$ intersects $\boldsymbol{\omega}$ at two complex

conjugate ideal points $\mathbf{I}$ and $\mathbf{J}$, called the *circular points* [HZ04]. The conic dual to the circular points is given by $\mathbf{C}_\infty^* = \mathbf{I}\mathbf{J}^\mathbf{T} + \mathbf{J}\mathbf{I}^\mathbf{T}$ where $\mathbf{C}_\infty^*$ is a degenerate conic invariant under similarity transformation. Under a point transformation, $\mathbf{C}_\infty^*$ transforms as:

$$\mathbf{C}_\infty^{*'} = (\mathbf{H_P H_A})\mathbf{C}_\infty^*(\mathbf{H_P H_A})^\mathbf{T} = \begin{bmatrix} \mathbf{K}\mathbf{K}^\mathbf{T} & \mathbf{K}\mathbf{K}^\mathbf{T}\mathbf{v} \\ \mathbf{v}^\mathbf{T}\mathbf{K}\mathbf{K}^\mathbf{T} & \mathbf{v}^\mathbf{T}\mathbf{K}\mathbf{K}^\mathbf{T}\mathbf{v} \end{bmatrix}$$

where $\mathbf{H_P}$ and $\mathbf{H_A}$ are respectively the projective and affine components of the projective transformation. It is clear that the affine ($\mathbf{K}$) and the projective ($\mathbf{v}$) components are determined directly from the image of $\mathbf{C}_\infty^*$. Once $\mathbf{C}_\infty^{*'}$ is identified, a suitable rectifying homography is obtained by using the SVD decomposition:

$$\mathbf{C}_\infty^{*'} = \mathbf{U} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{U}^\mathbf{T} \tag{9.1}$$

where $\mathbf{U}$ is the rectifying projectivity. A stratified solution is also proposed by Liebowitz [LZ98]. More results are provided in Section 9.5.1.

Fig. 9.1 depicts some results obtained by rectifying the obtained training trajectories from two of our three test sequences - each column represents a different sequence. Fig. 9.1(a) shows the training trajectories superimposed on the images plane. Fig. 9.1(c) is the rectified image, representing rectified trajectories, obtained by performing metric rectification on Fig. 9.1(a).

From here on, all references to 2-D image coordinates and trajectories imply *rectified* 2-D image coordinates and *rectified* trajectories, respectively. For simplicity and better visualization,

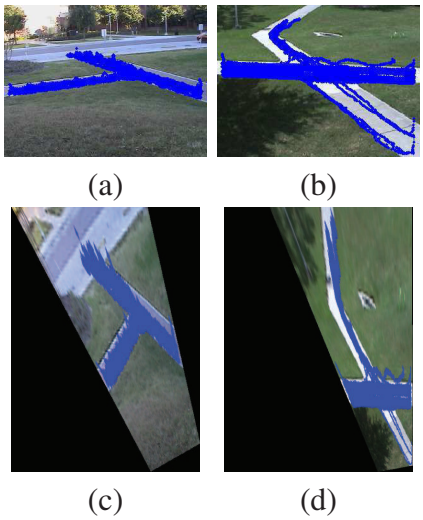


Figure 9.1: **Rectified Trajectories for two sequences (column wise)**:(c) represents reconstructed trajectories for **Seq** #2, while (d) represents **Seq** #3. Jagged dots at end points of the trajectories, in (d), are due to noisy tracking. See text for more details.

the results are still shown on un-rectified image plane in subsequent sections.

9.2.1 Model Building

Another important step during the training phase is to identify the different paths traversed by pedestrians in a scene. This section elaborates on how the extracted trajectories are used to create a path model.

Typical Setup: A typical setup consists of a single camera mounted on a wall or on a tripod looking at a certain location.For our training, we let people walk around the monitored scene and the object tracker gives the trajectories for the objects moving across the scene. Generally the trackers are able to uniquely label objects appearing in the sequence. Therefore, it is possible to maintain a history of the route taken by an object. For any object i tracked through n frames, the 2-D image coordinates for the trajectory obtained can be given as $\mathbf{T_i} = \{(\mathbf{x_1}, \mathbf{y_1}), (\mathbf{x_2}, \mathbf{y_2}), ..., (\mathbf{x_n}, \mathbf{y_n})\}$.

Note that the trajectories will be of varying lengths, depending on the location and velocity of the person. The trajectory of an object moving slowly will have more points (or pixels) compared to a fast moving object. For most tracking systems, it suffices to track the centroid of an object. But this might not be a good measurement for our system as we are dealing with physical pathways where the position of an object is very important. Thus, we track the feet of the objects for more precise measurements. The trajectories obtained through the tracker are sometimes noisy; therefore, trajectory smoothing is performed.

9.2.2 Trajectory Clustering

Once we have rectified trajectories from our training set, the next task is to cluster the trajectories into different paths. Clustering has to be based on some kind of similarity criteria. Perceptually, humans tend to group trajectories based on their spatial proximity. Since we are trying to create a path model, it is essential that we perform clustering using the spatial characteristics of the trajectories. Thus, we choose the Hausdorff distance as our similarity measure. For two trajectories $\mathbf{T_i}$ and $\mathbf{T_j}$, the Hausdorff distance, $\mathbf{D}(\mathbf{T_i}, \mathbf{T_j})$, is defined as:

$$\mathbf{D}(\mathbf{T_i}, \mathbf{T_j}) = \max\{\mathbf{d}(\mathbf{T_i}, \mathbf{T_j}), \mathbf{d}(\mathbf{T_j}, \mathbf{T_i}))\}, \tag{9.2}$$

where $d(\mathbf{T_i}, \mathbf{T_j}) = \underset{a \in \mathbf{T_i}}{\max} \; \underset{b \in \mathbf{T_j}}{\min} \|\mathbf{a} - \mathbf{b}\|$

The advantage of using Hausdorff distance is that it can compare two sets of different cardinality. Thus it allows us to compare two trajectories of different lengths. In order to cluster trajectories into different paths, we formulate a complete graph. Each node of the graph represents

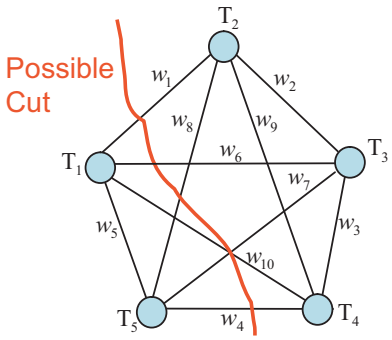
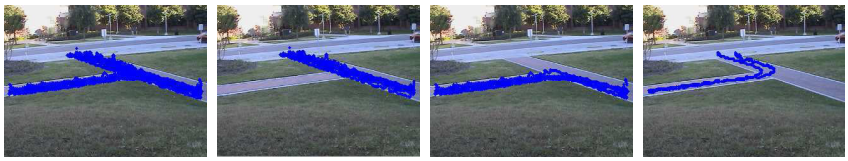


Figure 9.2: A complete graph of five nodes with Hausdorff distance as the edge weights. The red line may be a possible normalized-cut partitioning the graph into two subgraphs.



(a) (b) (c) (d)

Figure 9.3: Results of trajectory clustering using normalized-cuts. (a) all the trajectories in our training set **Seq #2** . After applying normalized-cuts, the clustered paths are shown in (b), (c) and (d).

a trajectory. The weight of each edge is determined by the Hausdorff distance between the two trajectories. The constructed complete graph needs to be partitioned. Each partition corresponds to a unique path, having one or more trajectories. To perform such a partition accurately and automatically, normalized-cuts [SM00] are used recursively to partition the graph. An example of graph formulation is given in Fig. 9.2.

Spatially proximal trajectories will have small weights because of lesser Hausdorff distance, and vice versa. This novel usage of normalized-cuts for trajectory clustering has certain advantages over other graph cut techniques. First, it avoids bias for partitioning out small sets of points.

Second, the problem is reduced to finding the eigenvectors of the system, which is very easy to compute. This technique makes it possible to perform recursive cuts by using special properties of the eigenvectors. We refer the reader to [SM00, SM98] for details on normalized-cuts. Fig. 9.3 shows the results obtained by clustering one of our data set.

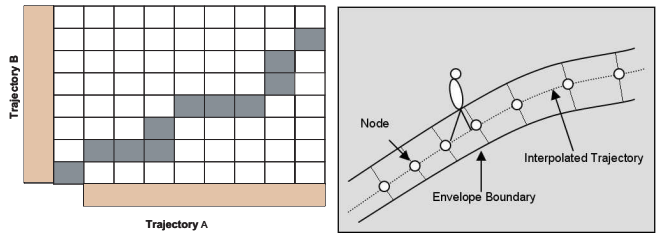
9.2.3 Envelope And Mean Path Construction

At this stage, trajectories have been clustered into different paths by applying normalized-cuts. Each path is represented by trajectories that make up that particular path. These trajectories, representing their corresponding paths, are used to create a path envelope and a mean path representation. An *envelope* can be defined as the spatial extent of a path (see Fig. 9.4). Applying the Dynamic Time Warping (DTW) [Keo02] algorithm, with column representing a trajectory **A** and the row representing a trajectory **B**, point-wise correspondences between the two trajectories is determined. Using DTW, distance at each instance is given by:

$$S(i,j) = \min\{S(i-1,j-1), S(i-1,j), S(i,j-1)\} + q(i,j) \tag{9.3}$$

where the distance measure is $q(i,j) = \frac{e^{\frac{(-\kappa(\mathbf{i},\mathbf{j}))}{\sigma_\kappa}} + e^{\frac{(-\overline{ij})}{\sigma_e}}}{2}$, $\overline{ij}$ represents the Euclidean distance, σ_κ represent the standard deviation in spatio-temporal curvature (explained later), and σ_e represents a suitable standard deviation parameter for the trajectory (in pixels). Thus, this distance measure merges trajectories based on the spatial as well as spatio-temporal curvature similarity. This algorithm is applied to the trajectories of all the obtained paths.

By pair-wise application of the above mentioned algorithm on each pair of trajectories from an obtained path, an envelope is created to represent the spatial extent of the path, and a mean



(a) (b)

Figure 9.4: (a) Standard construction for DTW algorithm for matching two trajectories **A** and **B**. (b) represents a typical scene where an object is traversing an existing path. An average trajectory and an envelope boundary are calculated for each set of clustered trajectories.

trajectory (using DTW) to represent all trajectories in the path. As shown in Fig. 9.5, for two trajectories, the point-wise matching between the two trajectories is carried out using the $S(i, j)$ measure defined above. Connecting the mid-points of the lines joining the matched corresponding points is taken as the mean path. And consequently, for these sample cases, the two trajectories, represented in green and red color, show the spatial extent of the path.

9.3 Test Phase: Scene Modeling And Verification

This section describes the *test phase*. A path model is developed that distinguishes between trajectories that are:

- Spatially unlike
- Spatially proximal, but of different speeds
- Spatially proximal but crooked
- or spatially proximal, but exceeding a maximum physical speed limit.

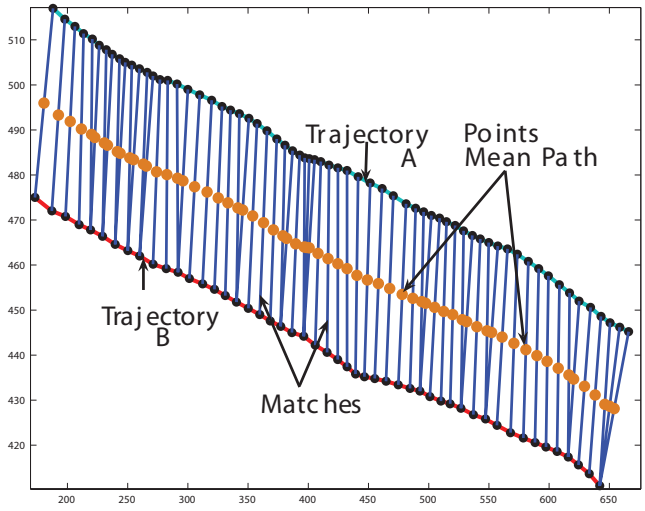


Figure 9.5: Dynamic Time Warping: An example of an average trajectory obtained by applying DTW on two sample trajectories. Blue lines connect corresponding matched points between the two trajectories.

To achieve these goals, first the usual paths are learned by applying normalized-cuts to cluster trajectories, as mentioned above. Once we have detected all paths in a scene, we apply our testing measure to verify the conformity of a candidate trajectory.

Validity of the candidate trajectory is tested based on its spatial, velocity and spatio-temporal curvature properties. Each of these tests serves a distinctive purpose. The usage of spatial properties for testing is to guarantee that the candidate trajectory is spatially close to our path i.e. to the envelope of our model. An anomalous trajectory can be discarded right away if it is considerably distant from the model. Using velocity characteristics allows us to distinguish between objects moving at different speeds e.g. a person walking compared to a person running, or recognize exceeding the expected physical speed limit. The spatio-temporal curvature measure makes it possible to distinguish between motion characteristics of our data and that of the candidate trajectory. For example, if our training data consists of pedestrians walking in straight line, then we can eas-

ily distinguish someone walking in a zigzag manner using our model, and hence classify it as an anomalous behavior.

Spatial Proximity: To verify spatial similarity, membership of the test trajectory is verified to the developed path model. All points on the candidate trajectory are compared to the envelope of the path model. The result of this process is a binary vector with 1 when a trajectory points is inside the envelope and 0 (zero) otherwise. This information is used to make a final decision for a candidate trajectory along with the spatio-temporal curvature measure. If all candidate trajectory points are outside the envelope, then this is an outright rejection.

Motion Similarity: The second step is essential to discriminate between trajectories of varying motion characteristics. The trajectory whose velocity is similar to the velocity characteristics of an existing route is considered similar. Velocity for a trajectory $\mathbf{T_i}(\mathbf{x_i}, \mathbf{y_i}, \mathbf{t_i})$, $i = 0, 1, , N-1$, is calculated as:

$$\mathbf{v'_i} = (\frac{\mathbf{x_{i+1}} - \mathbf{x_i}}{\mathbf{t_{i+1}} - \mathbf{t_i}} \frac{\mathbf{y_{i+1}} - \mathbf{y_i}}{\mathbf{t_{i+1}} - \mathbf{t_i}}), \mathbf{i} = \mathbf{0}, \mathbf{1}, \ldots, \mathbf{N-1} \tag{9.4}$$

Mean and the standard deviation of the motion characteristics of the training trajectories are computed. A Gaussian distribution is fitted to model the velocities of the trajectories in the path model. The Mahalanobis distance measure is used to decide if the test trajectory is anomalous.

$$\tau = \sqrt{(\mathbf{v'_i} - \mathbf{m'_p})^{\mathbf{T}} (\sum)^{-\mathbf{1}} (\mathbf{v'_i} - \mathbf{m'_p})} < \varphi \tag{9.5}$$

Where $\mathbf{v'_i}$ is velocity from the test trajectory, $\mathbf{m'_p}$ is the mean, φ a distance threshold, and $\sum$ is the covariance matrix of our path velocity distribution.

Spatio-Temporal Curvature Similarity: The third step allows us to capture the discontinuity in the velocity, acceleration and position of our trajectory. Thus we are able to discriminate between a person walking in a straight line and a person walking in an errant path. The velocity $\mathbf{v}_\mathbf{i}'$ and acceleration $\mathbf{v}_\mathbf{i}''$, first derivative of the velocity, is used to calculate the curvature of the trajectory. Curvature is defined as:

$$\kappa = \frac{\sqrt{\mathbf{y}''(\mathbf{t})^\mathbf{2} + \mathbf{x}''(\mathbf{t})^\mathbf{2} + (\mathbf{x}'(\mathbf{t})\mathbf{y}''(\mathbf{t}) - \mathbf{x}''(\mathbf{t})\mathbf{y}(\mathbf{t}))^\mathbf{2}}}{(\sqrt{\mathbf{x}'(\mathbf{t})^\mathbf{2} + \mathbf{y}'(\mathbf{t})^\mathbf{2} + \mathbf{1}})^\mathbf{3}} \tag{9.6}$$

Where x' and y' are the velocity components in x and y direction, respectively. Mean and standard deviation of κ's is determined to fit a Gaussian distribution for spatio-temporal characteristic. We compare the curvature of the test trajectory with our distribution using the Mahalanobis distance, bounded by a threshold. By using this measure we are able to detect irregular motion. For example, a drunkard walking in a zigzag path, or a person slowing down and making a u-turn.

True Physical Velocity: This measure is obtained by registering the ground-based surveillance cameras to aerial imagery. It is known that under projective imaging, a plane is mapped to the image plane by a perspective transformation. One way to uniquely identify this projective transformation is when the Euclidean world coordinates of four or more points are known. Thereafter, the images can be rectified to one that would have been obtained from a fronto-parallel view of the plane for a good registration to the aerial imagery. However, this imposes too many restrictions on the image rectification process as the knowledge of the world points is not always readily available, and the process can not be automated. To make this process automatic (i.e. without having to manually specify the Euclidean world coordinates of points), the estimated affine and the perspective

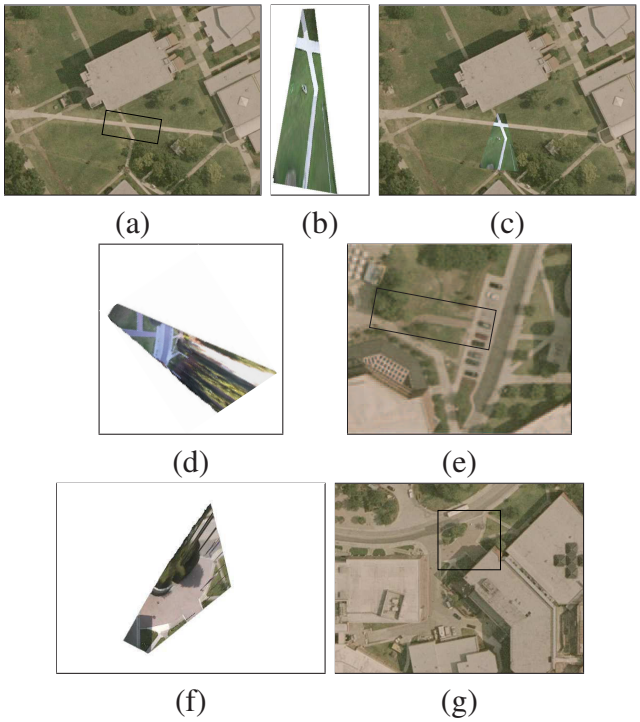


Figure 9.6: **Image Rectification and Registration:** (a) An image from **Seq # 3**, where as (b) is the metric rectified image for the same sequence. The metric rectified image is then registered to the satellite image as shown in (c). A rectified frame from **Seq # 2** and **Seq # 1** are shown in (d) and (f), respectively. The satellite images for these sequences are shown in (e) and (g), respectively. Since the satellite imagery (i.e. (e) and (g)) is different from the test sequences (due to new construction), the test images (i.e. (d) and (f) are not registered.

transform can be combined together to efficiently metric rectify the video sequence such that the only unknown transformation is the similarity transformation. We then use the method presented in [SS06, SGJ05] to perform image registration.

Fig. 9.6 shows an example of our automatic registration to aerial imagery. Once, a video sequence is registered to an aerial image, it is possible to retrieve metric information from the input video sequence, e.g. the true physical velocity. Generally, aerial images contain the world-to-image scale information, for instance in Fig. 9.9, where 140 pixels correspond to 40 yards. We use this estimated velocity to test if an object violates any established speed restrictions in a scene.

Given the spatial, and spatio-temporal measures computed as described above, we can examine the conformity of any incoming sequence. Thus, initially we detect non-conforming trajectories on the basis of spatial dissimilarity. In case the given trajectory is spatially similar to one of the path models, the similarity in the velocity feature of the trajectories in that path and the given trajectory is computed. If the motion features are also similar then a final check on spatio-temporal curvature is made. In addition to these similarity measures, we also determine the true physical speed to verify if a maximum permitted speed is violated. The trajectory is deemed to be anomalous if it fails to satisfy any one of the spatial, velocity or spatio-temporal curvature constraints.

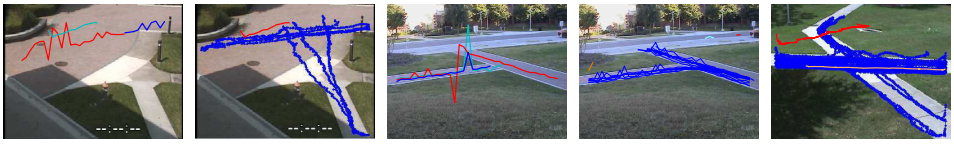
9.4 Handling Occlusions

For object detection and tracking, we use the method proposed by [JS02]. When an occlusion occurs the accurate position and velocity of the occluded object can not be determined. Few cases of occlusion are:

Inter-object occlusion occurs when one object blocks the view of other objects in the field of view of the camera. The background subtraction method gives a single region for occluding objects. If two initially non-occluding objects cause occlusion then this condition can be easily detected.

Occlusion of objects due to thin scene structures like poles or trees causes an object to break into two regions. Thus more than one extracted region can belong to the same object in such a scenario.

Occlusion of objects due to large structures causes the objects to disappear completely for a



(a) (b) (c) (d) (e)

Figure 9.7: Some cases of trajectories resulting from occlusion during the training phase and the test phase. (a) and (c) shows some trajectories obtaining due to occlusion not included in the training set for **Seq # 1** and **Seq # 2**, respectively. (b), (d) and (e) show some incomplete trajectories obtained due to occlusion which were rejected during the test phase.

certain amount of time, that is there is no foreground region representing such objects.

More details on how we handle these occlusions during the tracking process can be found in [JS02]. Although our tracking can handle occlusions to a great degree, not all cases can be handled correctly. As a result, we obtain incorrect trajectories, which affects our trajectory clustering method. During our training phase, two cases are considered:

1. When Inter-Object occlusion occurs: This kind of occlusion generates incomplete trajectories, i.e. a trajectory starts from one end of the image and ends before reaching the image boundary (possibly an exit point). We ignore this trajectory and do not use in our path building phase.

2. A new trajectory is generated not at the boundary of the image, but rather well inside the image plane. This generally occurs when scene structures causes an object to break, or when the tracker assigns new trajectories to objects emerging from occlusion. We also ignore this type of trajectory.

Some cases of trajectories resulting from occlusions are shown in Figure 9.7. Currently, occluding trajectories are not used in the training phase. Mainly because using partial or incomplete

trajectories would in general lead to an incorrect path model. However, some user-defined cases may be included if required.

During the testing phase, trajectories resulting from occlusion are not treated specially. If such a trajectory does satisfy the spatial proximity feature, it fails the motion and spatio-temporal features. This happens because there is no information regarding velocity and the curvature of the trajectory at the missing sections of the trajectory.

9.5 Results

The proposed system has been tested on multiple sequences with a variety of motion trajectories. The sequences have a resolution of 320×240 pixels and captured at multiple locations and each location contained multiple paths of travel. Three test sequences were used for evaluation purposes, named **Seq** $\#1$, **Seq** $\#2$, and **Seq** $\#3$. Our tracker is able to accurately establish correspondences over a variety of environmental conditions. Some test results and examples were provided throughout this chapter to clarify and illustrate the steps. Below, we present additional experimental evaluations.

9.5.1 Evaluating Registration To Aerial Imagery

Registration to aerial imagery gives a global view of the scene that is under observation, and allows for measuring physical quantities such as speed for determining conformity of incoming trajectories. The results obtained by rectifying the test sequences are shown in Fig. 9.6. A frame from the test sequence **Seq # 3** is rectified by using the line at infinity which is obtained as: $\mathbf{l}_\infty = \boldsymbol{\omega}\mathbf{v}_\mathbf{z}$. The obtained circular points are used to construct the conic $\mathbf{C}_\infty^{*'}$ in order to obtain the rectifying projec-

(a) (b) (c) (d) (e) (f)

Figure 9.8: Six test cases used to retrieve metric information. See text for more.

tivity, as described in Section 9.2.1. The rectified image is shown in Fig. 9.6(b), and the registered image is shown in Fig. 9.6(c). Similar computation for **Seq # 2** and **Seq # 1** produce the rectified images as shown in 9.6(d) and 9.6(f), respectively. Due to some newly constructed structures, the aerial imagery for **Seq # 2** and **Seq # 1** is somewhat different from the test sequences, hence the images are not perfectly registered.

Five cases are shown in Fig. 9.8 for computing physical speed. Fig. 9.8(a) shows a golf cart that takes only two seconds to move across the scene - the true speed obtained from the registered image is found to be 20.369 km/hr. The velocity of the bicycle, as shown in Fig. 9.8(b), is found to be 12.22 km/hr, whereas for three cases of pedestrians (i.e. Fig. 9.8(c)-(e)) the velocity is determined to be 4.58 km/hr, 3.66 km/hr, and 4.22 km/hr, respectively, which is very close to the average human walking speed. A case of a person riding a skate board is shown in Fig. 9.8(f) and the retrieved velocity is 9 km/hr.

Registration of multiple cameras to the aerial image is shown in Fig. 9.9. Three cameras were placed at three different locations along the path shown in the figure. Behavior of objects in the regions covered by the three cameras can be modeled by the proposed method and gives, in essence, the global behavior of the objects. Moreover, after metric rectification, the data obtained from multiple cameras can be used to obtain correct object correspondences across multiple non-overlapping

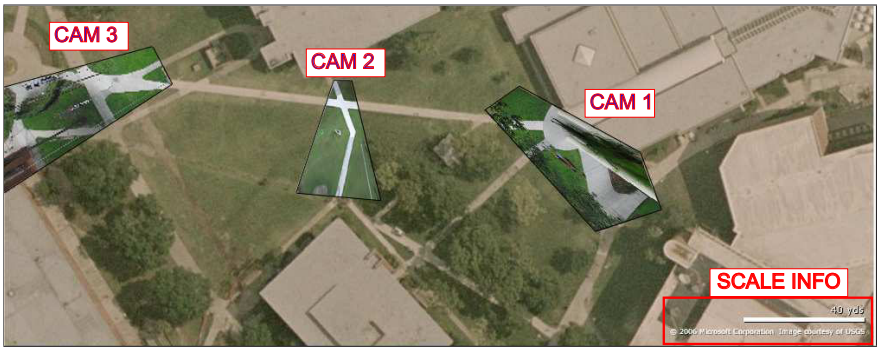


Figure 9.9: Multiple cameras registered to the corresponding satellite image: The input images have a few new structures compared to the old satellite image.

cameras i.e. the problem of object hand-over across multiple cameras (see [KCM03, JSS05]). For example, the true velocity of an object, or the height of an object can be extracted (once the camera is calibrated) and used as an additional feature for obtaining the correct correspondences across multiple non-overlapping camera.

9.5.2 Evaluating Path Modeling

As described above, during the training phase, normalized-cuts are applied to the trajectories in order to extract different paths in the scene. Once the different paths are determined, various characteristics are extracted form the trajectories in each path (Section 9.3). Three test sequences of varying length used:

Seq #1: This is a short sequence of 3730 frames with 15 different trajectories forming two unique paths. The clustered trajectories are shown in Fig. 9.10.

Trajectories obtained for the training sequence are depicted in Fig. 9.10(a)(b)(c), representing different behavior of the pedestrians. One test case is shown in Fig. 9.10(d). The training sequence only contained people walking in the scene. But the cyclist shown in (d) has motion character-

(a) (b) (c) (d)

Figure 9.10: (b)(c) show three clustered path for **Seq** #1 while (a) shows all the trajectories in the training phase. (d) demonstrates a test case where a bicyclist crosses the scene at a velocity greater than the pedestrians observed during the training phase.

(a) (b) (c)

Figure 9.11: Results obtained from **Seq** #2. Image (a),(b) and (c) show instances of a drunkard walking, a person running, and a person walking, respectively. Red trajectories denote unusual behavior while the black trajectories are the casual behavior.

istics different (containing faster movement) than the training cases, hence detected as abnormal behavior (displayed in red).

Seq #2: A real sequence of 9284 frames with 27 different trajectories forming 3 different paths after clustering. The length of the trajectories varies from 250 points to almost 800 points. The trajectories clustered into paths are shown in Fig. 9.3. The sequence contained pedestrians walking in either a straight line, or move left/right at the junction.

Three test cases are depicted in Fig. 9.11. A person walking in a zigzag fashion (Fig. 9.11(a)), and a person running (Fig. 9.11(c)) are flagged for an activity that is considered as an unusual behavior. Fig. 9.11(b) demonstrates a case where a person walks at a normal pace in conforming behavior.

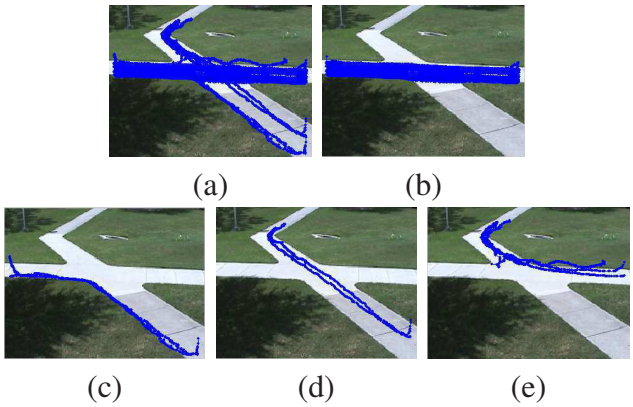


Figure 9.12: Results from the training sequence of **Seq** #3: (a) shows all the trajectories used in the training set. (b)-(d) are the 4 paths clustered from the input data.

Seq #3: The training sequence contains over 20 minutes of data forming over 100 trajectories of people walking around in the scene. The trajectories are clustered into 4 path models: horizontal movement, people coming from the upper part of the scene and going to the right, people coming from the upper region and coming to the lower right, and people coming from the left region and moving towards the lower part of the image. Trajectories clustered into different paths are shown in Fig. 9.12.

Some of the test cases are shown in Fig.9.13 (column wise). Two cases Fig.9.13(a,e) and Fig.9.13(b,f) contain people walking at normal pace - following the path model constructed in the training phase, hence flagged with a black trajectory i.e. acceptable behavior. Third column Fig.9.13(c,d) is flagged unacceptable as the person moves left, which is not contained in the model. Similarly, 4^{th} column contains a golf cart driven across the scene.

The system gives satisfactory results for all our experiments and is fairly efficient. Although some existing methods do incorporate model update, we believe this is what leads to a *model drift*. That is, after a number of updates the model can become general enough to accommodate any be-

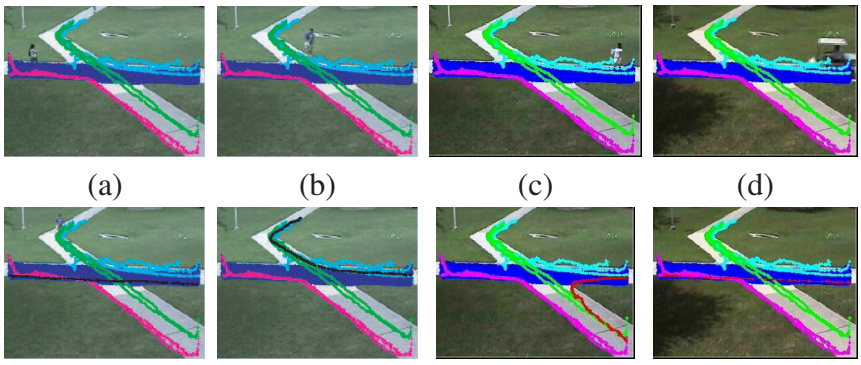


Figure 9.13: Results for **Seq** #3. Column 1 and 2 demonstrate normal behavior, while column 3 and 4 demonstrate two examples of unacceptable behaviors. See text for more details.

havior considering it as acceptable behavior. But certainly, the applicability of the proposed system lies in the spheres where there is a defined behavior, differentiable from certain other unacceptable behavior for, lets say, security reasons.

9.6 Conclusion

This chapter proposes a unified method for path modeling, detection and surveillance. The trajectory data is metric rectified to represent a truer picture of the data. Metric rectified observed scene is registered to aerial view to extract metric information from the video sequence, for example, the actual speed of an object. Normalized-cuts are then used to cluster metric rectified input training trajectories into various paths. We extract spatial, velocity and spatio-temporal curvature based features from the clustered paths and use it for unusual behavior detection. The proposed path modeling method has been extensively tested on a number of sequences and have demonstrated satisfactory results. Recognizing more complex events by attaching meanings to the trajectories is also one of our future goals.

CHAPTER 10
ESTIMATING GPS COORDINATES FROM IMAGES

In Chapter 4, we described how to calibrate a camera by presenting two different methods for estimating $\mathbf{l}_\infty$. As described below, in order to perform geo-temporal localization, we need to estimate the azimuth and the altitude angle of the sun. For this, it is necessary that the object bottom and top be visible in the image. However, if by some other technique the above mentioned two angles are readily available, then it is not necessary for the object to be visible.

In ICCV 2005 a contest was run on a collection of color images acquired by an already calibrated digital camera. The photographs were taken at various locations and often shared overlapping fields of view, or had certain objects in common. More importantly, the GPS locations for a subset of these images were provided in advance. The goal of the contest was to guess, as accurately as possible, the GPS locations of the unlabeled images. This chapter pushes the limits in the state of the art beyond what is currently known to be feasible from images in terms of geo-temporal localization solely based on computer vision techniques.

The cue that we use to geo-localize the camera and to determine the date of acquisition is the shadow trajectories of two stationary objects during the course of a day. Shadows have been used in multiple-view geometry in the past to provide information about the shape and the 3-D structure of the scene [BP98, CW06], or to recover camera intrinsic and extrinsic parameters [AB04, CS05]. Shadows are also recognized as useful tools for determining the time of the day. The use of shadow trajectory of a gnomon to measure time in a sundial is reported to as early as 1500 BC by Egyptians, which surprisingly requires sophisticated astronomical knowledge [Her67, III94, Wau73]. Determining the GPS coordinates and the date of the year from shadows in images

is a new concept that we introduce in this chapter.

In terms of applications, it is clear that the ability to determine geo-temporal information directly from visual cues, and without using any special instruments, opens new opportunities for the use of camera systems, or processing of visual data. Numerous applications may be envisioned, amongst which forensics, intelligence, security, and navigation are perhaps the most important ones. To demonstrate the power of the proposed method we downloaded images from online traffic surveillance webcams, and determined accurately the geo-locations and the date of acquisition.

10.1 The Geo-temporal Localization Step

After auto-calibration, we can determine the geo-location up to longitude ambiguity, and specify the day of the year when the images were taken up to, of course, year ambiguity. This is possible by using only three shadow points, compared to 5 required for the camera calibration. The key observation that allows us to achieve this is the fact that a calibrated camera performs as a direction tensor, capable of measuring direction of rays and hence angles, and that the latitude and the day of the year are determined simply by measuring angles in images.

Latitude: An overview of the proposed method is shown in Fig. 10.1. Let $\mathbf{s}_i$, $i = 1, 2, 3$ be the images of the shadow points of a stationary object recorded at different times during the course of a single day. Let $\mathbf{v}_i$ and $\mathbf{v'}_i$, $i = 1, 2, 3$ be the sun and the shadow vanishing points, respectively. For a calibrated camera, the following relations hold for the altitude angle ϕ_i and the azimuth angle

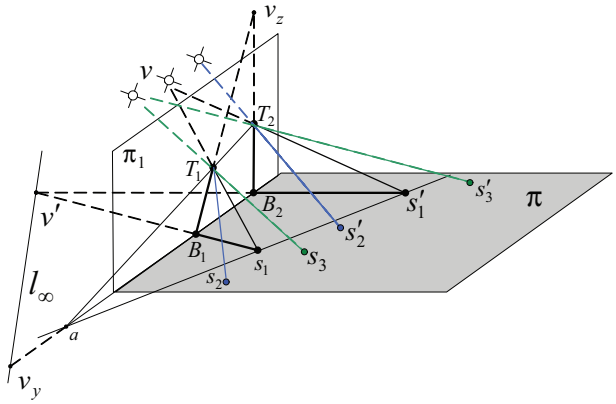


Figure 10.1: The setup used for estimating geo-temporal information.

θ_i of the sun orientations in the sky, all of which are measured directly in the image domain:

$$\cos\phi_i = \frac{\mathbf{v'}_i^T \boldsymbol{\omega} \mathbf{v}_i}{\sqrt{\mathbf{v'}_i^T \boldsymbol{\omega} \mathbf{v'}_i}\sqrt{\mathbf{v}_i^T \boldsymbol{\omega} \mathbf{v}_i}} \tag{10.1}$$

$$\sin\phi_i = \frac{\mathbf{v}_z^T \boldsymbol{\omega} \mathbf{v}_i}{\sqrt{\mathbf{v}_z^T \boldsymbol{\omega} \mathbf{v}_z}\sqrt{\mathbf{v}_i^T \boldsymbol{\omega} \mathbf{v}_i}} \tag{10.2}$$

$$\cos\theta_i = \frac{\mathbf{v}_y^T \boldsymbol{\omega} \mathbf{v'}}{\sqrt{\mathbf{v}_y^T \boldsymbol{\omega} \mathbf{v}_y}\sqrt{\mathbf{v'}^T \boldsymbol{\omega} \mathbf{v'}}} \tag{10.3}$$

$$\sin\theta_i = \frac{\mathbf{v}_x^T \boldsymbol{\omega} \mathbf{v'}}{\sqrt{\mathbf{v}_x^T \boldsymbol{\omega} \mathbf{v}_x}\sqrt{\mathbf{v'}^T \boldsymbol{\omega} \mathbf{v'}}} \tag{10.4}$$

Without loss of generality, we choose an arbitrary point on the horizon line as the vanishing point $\mathbf{v}_x$ along the x-axis, and the image point $\mathbf{b}$ of the footprint as the image of the world origin. The vanishing point $\mathbf{v}_y$ along the y-axis is then given by $\mathbf{v}_y \sim \boldsymbol{\omega}\mathbf{v}_x \times \boldsymbol{\omega}\mathbf{v}_z$. Now, let ψ_i be the angles measured clockwise that the shadow points make with the positive x-axis as shown in Fig. 10.1. We have

$$\cos\psi_i = \frac{\mathbf{v'}_i^T \boldsymbol{\omega} \mathbf{v}_x}{\sqrt{\mathbf{v'}_i^T \boldsymbol{\omega} \mathbf{v'}_i}\sqrt{\mathbf{v}_x^T \boldsymbol{\omega} \mathbf{v}_x}} \tag{10.5}$$

$$\sin\psi_i = \frac{\mathbf{v'}_i^T \boldsymbol{\omega} \mathbf{v}_y}{\sqrt{\mathbf{v'}_i^T \boldsymbol{\omega} \mathbf{v'}_i}\sqrt{\mathbf{v}_x^T \boldsymbol{\omega} \mathbf{v}_y}} \quad i = 1, 2, 3 \tag{10.6}$$

Next, we define the following ratios, which are readily derived from spherical coordinates, and also used in sundial construction [Her67, III94, Wau73]:

$$\rho_1 = \frac{\cos\phi_2 \cos\psi_2 - \cos\phi_1 \cos\psi_1}{\sin\phi_2 - \sin\phi_1} \tag{10.7}$$

$$\rho_2 = \frac{\cos\phi_2 \sin\psi_2 - \cos\phi_1 \sin\psi_1}{\sin\phi_2 - \sin\phi_1} \tag{10.8}$$

$$\rho_3 = \frac{\cos\phi_2 \cos\psi_2 - \cos\phi_3 \cos\psi_3}{\sin\phi_2 - \sin\phi_3} \tag{10.9}$$

$$\rho_4 = \frac{\cos\phi_2 \sin\psi_2 - \cos\phi_3 \sin\psi_3}{\sin\phi_2 - \sin\phi_3} \tag{10.10}$$

$$\tag{10.11}$$

For our problem, it is clear from (10.1)-(10.6) that these ratios are all determined directly in terms of image quantities. The angle measured at world origin between the positive y-axis and the ground plane's primary meridian (i.e. the north direction) is then given by

$$\alpha = \tan^{-1}\left(\frac{\rho_1 - \rho_3}{\rho_4 - \rho_2}\right) \tag{10.12}$$

from which we can determine the GPS latitude of the location where the pictures are taken as

$$\lambda = \tan^{-1}(\rho_1 \cos\alpha + \rho_2 \sin\alpha) \tag{10.13}$$

For n shadow points, we obtain a total of $\frac{n!}{(n-3)!3!}$ estimations of latitude(λ). In presence of noise, this leads to a very robust estimation of λ.

Day Number: Once the latitude is determined from (10.13), we can also determine the exact day when the images are taken. For this purpose, let δ denote the declination angle, i.e. the angle of the sun's rays to the equatorial plane (positive in the summer). Let also $\hbar$ denote the hour angle for a given image, i.e. the angle the earth needs to rotate to bring the meridian of that location to solar noon, where each hour time corresponds to $\frac{\pi}{12}$ radians, and the solar noon is when the sun is due south with maximum altitude. Then these angles are given in terms of the latitude λ, the sun's altitude ϕ and its azimuth θ by

$$\sin\hbar\cos\delta - \cos\phi\sin\theta = 0 \tag{10.14}$$

$$\cos\delta\cos\lambda\cos\hbar + \sin\delta\sin\lambda - \sin\phi = 0 \tag{10.15}$$

Again, note that the above system of equations depend only on image quantities defined in (10.1)-(10.6). Upon finding the declination and the hour angles by solving the above equations, the exact day of the year when the pictures are taken can be found by

$$N = \frac{365}{2\pi}\sin^{-1}\left(\frac{\delta}{\delta_m}\right) - N_o \tag{10.16}$$

where N is the day number of the date, with January 1^{st} taken as $N = 1$, and February assumed of 28 days, $\delta_m \simeq 0.408$ is the maximum absolute declination angle of earth in radians, and $N_o = 284$ corresponds to the number of days from the first equinox to January 1^{st}.

Longitude: Unfortunately, unlike latitude, the longitude cannot be determined directly from observing shadows. The longitude can only be determined either by spatial or temporal correlation. For instance, if we know that the pictures are taken in a particular state or a country or a region in the world, then we only need to perform a one-dimensional search along the latitude determined by (10.13) to find also the longitude and hence the GPS coordinates. Alternatively, the longitude may be determined by temporal correlation. For instance, suppose we have a few frames from a video stream of a live webcam with unknown location. Then they can be temporally correlated with our local time, in which case the difference in hour angles can be used to determine the longitude.

For this purpose, let $\hbar_l$ and γ_l be our own local hour angle and longitude at the time of receiving the live pictures. Then the GPS longitude of the location where the pictures are taken is given by

$$\gamma = \gamma_l + (\hbar - \hbar_l) \tag{10.17}$$

In the next section, we validate our method and evaluate the accuracy of both self-calibration and geo-temporal localization steps using synthetic and real data.

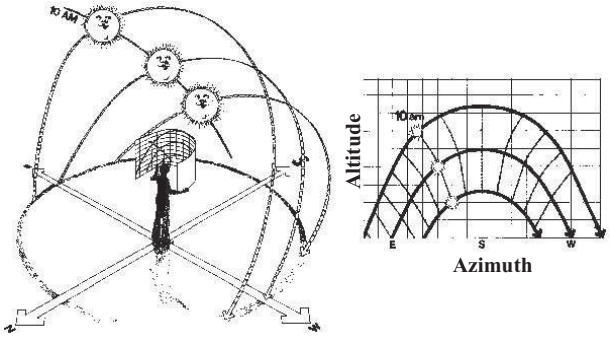


Figure 10.2: The Cylindrical of Sun Path Diagram (Mazria, Edward, The Passive Solar Energy Book). The shadow of an object throughout the course of a day follows a curve on the ground plane.

10.2 Using only two shadow points

For any location on the globe, the relationship between the location of the sun and the shadow is unique. This relationship can be graphically represented through sun-path diagrams. The exact position of the sun can be determined for any given time of the day using only the azimuth and altitude angle of that site. Figure 10.2 shows an example of vertical projection of sun-path as observed from earth. The vertical axis denotes the altitude and the horizontal axis denotes the azimuth angle. This plot is an earth base view of the sun's movement across the celestial sphere. The exact form of the curve depends on the location (latitude and longitude) and the time of the year. The question now is, can we estimate the GPS coordinates from just two points, whereas in previous sections we used three points?

The method presented in Section 10.1 requires azimuth and altitude angles, θ and ϕ respectively, of at least three shadow points. We also need to estimate the four ratios, i.e. (10.7)-(10.10), which depends on the angle, ψ. This angle ψ is measured between the shadow point $\mathbf{v}'$ and the $+ve\ x$-axis, as shown in Fig. 10.1. Therefore, we need to *first*, estimate the azimuth and altitude

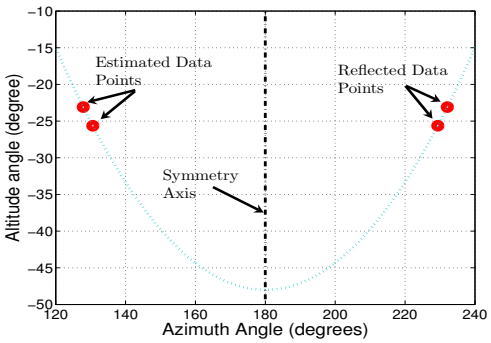


Figure 10.3: A $2^{nd} - degree$ polynomial fitted to the estimated altitude and azimuth angles.

angle of the sun for any time of the day, and *second*, estimate the vanishing point $\mathbf{v}'$ of the shadows cast at that particular time.

It becomes clear upon observing Fig. 10.2 that the sun-path curve is symmetric. The axis of symmetry is exactly at $180°$ azimuth angle. This corresponds to the solar noon, that is, when the sun is at its highest point. Now consider the case when we have only two images i.e. we have only two shadow points. This is shown in Fig. 10.3. The axis of symmetry is plotted by a vertical line at $\theta = 180°$. The two shadow points obtained from the images are plotted on the left of this axis. These two points are then reflected across the axis, as shown in the figure. The problem now reduces to fitting a polynomial curve to these four points. A polynomial of k^{th} degree is given as:

$$y = a_0 + a_1 x + \ldots a_k x^k \tag{10.18}$$

where the goal is to minimize the residual

$$R = \sum_{i=1}^{n} [y_i - (a_0 + a_1 x + \ldots a_k x^k)]^2$$

to fit the model as close to the data as possible. In matrix notation, the solution to the polynomial fit is given by:

$$\mathbf{y} = \mathbf{X}\mathbf{a} \tag{10.19}$$

where $\mathbf{y}$ contains the LHS of (10.18) evaluated for all data points, the matrix X contains the x values of the data points from the RHS of (10.18), and $\mathbf{a}$ contains the unknown parameters a_i [PFT88]. (10.19) can be solved as:

$$\mathbf{a} = (\mathbf{X^T X})^{-1}\mathbf{X^T y} \tag{10.20}$$

In our experiments, the polynomial that best fits the shadow data is that of degree 2. This is plotted in Fig. 10.3 as a dotted green curve. Once this curve is obtained, altitude ϕ_3 of any azimuth θ_3 of our choice can be estimated and vice versa.

Once (ϕ_3, θ_3) are obtained from the fitted shadow curve, the shadow point $\mathbf{v}'$ is obtained by solving the two equations:

$$\mathbf{v'^T l}_\infty = \mathbf{0} \tag{10.21}$$

$$\cos\theta_3 = \frac{\mathbf{v}_y^T \boldsymbol{\omega} \mathbf{v}'}{\sqrt{\mathbf{v}_y^T \boldsymbol{\omega} \mathbf{v}_y}\sqrt{\mathbf{v}'^T \boldsymbol{\omega} \mathbf{v}'}} \tag{10.22}$$

Once $\mathbf{v}'$ is obtained, (10.5) is used to estimate ψ to determine the four ratios i.e. (10.7)-(10.10). This enables use to used the method described in Section 10.1 to estimated the GPS coordinates.

10.3 Experimental Results

We rigorously test and validate our method on synthetic as well as real data. Results are described below.

Synthetic Data: Two vertical objects of different heights were randomly placed on the ground plane. Using the online available version of SunAngle Software [Gro], we generated altitude and azimuth angles for the sun corresponding to our own geo-location with latitude 28.51°. The data was generated for the 315^{th} day of the year i.e. the 11^{th} of November 2006 from `10:00am` to `2:00pm`. The solar declination angle for that time period is -17.49°. The vertical objects and the shadow points were projected by a synthetic camera with a focal length of $f = 1000$, the principal point at $(u_o, v_o) = (320, 240)$, unit aspect ratio, and zero skew.

Averaged results for latitude, solar declination angle, and the day of the year are shown in Figure 10.4. The error is found to be less than 0.9%. For a maximum noise level of 1.5 pixels, the estimated latitude is 28.21°, the declination angle is -17.932°, and the day of the year is found to be 314.52.

Real Data: Several experiments on two separate data sets are reported below for demonstrating the power of the proposed method. In the first set, 11 images were captured live from downtown Washington D.C. area, using one of the webcams available online at `http://trafficland.com/`. As shown in Figure 10.5, a lamp post and a traffic light were used as two objects casting shadows on the road. The shadow points are highlighted by colored circles in the figure.

Since we had more than the required minimum number of shadow locations over time, in order to make the estimation more robust to noise, we took all possible combinations of the available

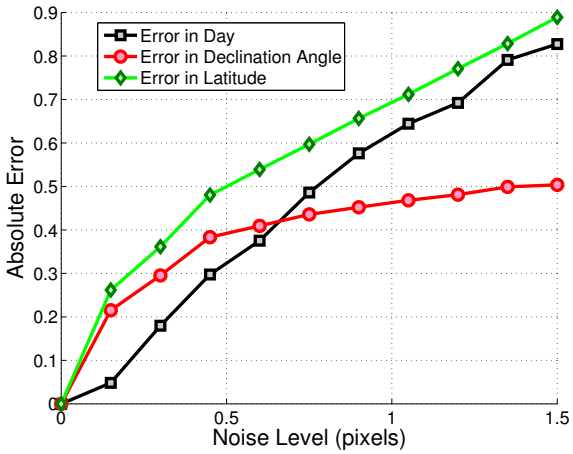


Figure 10.4: Performance averaged over 1000 independent trials: Result for average error in latitude, solar declination angle, and day of the year.

Figure 10.5: Few of the images taken from one of the live webcams in downtown Washington D.C. The two objects that cast shadows on the ground are shown in red and blue, respectively. Shadows move to the left of the images as time progresses.

points and averaged the results. For this first data set the images were captured on the 15^{th} November at latitude $38.53°$ and longitude $77.02°$. We estimated the latitude as $38.444°$, the day number as 316.293 and the solar declination angle as $-19.258°$ compared to the actual day of 319, and the declination angle of $-18.62°$. The small errors can be attributed to many factors e.g. noise, non-linear distortions and errors in the extracted features in low-resolution images of 320×240. Despite all these factors, the experiment indicates that the proposed method provides good results.

In order to evaluate the uncertainty associated with our estimation, we then divided this data set into 11 sets of 10-image combinations, i.e. in each combination we left one image out. We repeated

Table 10.1: Results for 11 sets of 10-image combination, with mean value and standard deviation.

	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_3$	$\mathcal{C}_4$	$\mathcal{C}_5$	$\mathcal{C}_6$	$\mathcal{C}_7$	$\mathcal{C}_8$	$\mathcal{C}_9$	$\mathcal{C}_{10}$	$\mathcal{C}_{11}$	Mean	STD
λ	33.73	35.70	37.03	36.1	35.72	38.21	39.23	45.78	41.84	40.88	41.96	38.743	3.57
δ	-14.47	-15.78	-15.93	-16.54	-17.25	-16	-16.70	-18.94	-15.87	-16.99	-16.24	-16.43	1.11
N	328.64	332.26	331.09	326.87	330.15	331.37	331.32	332.56	326.81	331.72	326.72	329.95	2.28

Figure 10.6: Few of the images in the second data set that were temporally correlated with our local time, taken also from one of the live webcams in Washington D.C. The objects that cast shadows on the ground are highlighted. Shadows move to the left of the images as time progresses.

the experiment for each combination and calculated the mean and the standard deviation of the estimated unknown parameters. Results are shown in Table 10.1. The low standard deviations can be interpreted as small uncertainty, indicating that our method is consistently providing reliable results.

A second data set is shown in Figure 10.6. The ground truth for this data set was as follows: longitude $77.02°$, latitude $38.53°$, day number of 331, and the declination of $-21.8°$. For this data set we assumed that the data was downloaded in real-time and hence was temporally correlated with our local time. We estimated the longitude as $78.761°$, the latitude as $37.791°$, the day number as 323.0653, and the declination angle as $-29.65°$.

10.4 Conclusion

This chapter describes a novel method based entirely on computer vision to determine the geo-location of the camera up to longitude ambiguity, without using any GPS or other instruments,

and by solely relying on imaged shadows as cues. We also describe situations where longitude ambiguity can be removed by either temporal or spatial cross-correlation. Moreover, we determine the date when the pictures are taken without using any prior information. The method is tested on synthetic as well as on real data, and the results are promising.

CHAPTER 11
APPLICATION TO MR ENVIRONMENT

A Mixed Reality (MR) system combines the real scene viewed by the user/agent and the virtual scene generated by the computer that augments the scene with some additional information. In order to successfully accomplish this task, the position and orientation of each user is tracked by the means of inertial sensors attached to the video see-through head mounted displays (HMDs) in a controlled MR environment. See [YWC05] for pose estimation in an augmented/mixed reality scenario. Figure 11.1(a)(b) shows images of such a scenario. A video see-through HMD consist of small mounted cameras that capture the surrounding environment. On the inside of the HMD, the captured video is played to the user in real-time possibly with some virtual information. While sufficient for indoors, this approach is not feasible for outdoor scenarios. The reason is that active tracking sensors (transmitter, receiver) systems are not portable and can only operate indoor under fixed and expensive setups. The cost involved is very high. Since HMDs contain mounted cameras, henceforth we simply use camera when referring to a HMD.

11.1 Estimating Relative Orientation

In order to successfully merge virtual information with real, each user's position and orientation has to be tracked continuously. For our experiments, we had two users wearing Canon Coastar video see-through Head-Mounted Displays HMDs walk in a family size room equipped with Polhemus magnetic tracker and an Intersense IS-900/PC hybrid acoustical/inertial tracker. In order to verify our method, described in Chapter 8, we compute the absolute rotation of each HMD w.r.t. the world co-ordinate system. We compared our results with the ground-truth from active sensors.

(a) (b) (c)

Figure 11.1: (a) shows a general setup of a MR environment. (b) is a picture taken of a user with an HMD mounted on his head. (c) Instances of the test data set. These images are taken from HMDs mounted on two users. See text for details.

Table 11.1: Error in degree for the angles calculated. See text for details.

Instance #	Error (θ_x)	Error (θ_y)	Error (θ_z)
1	2.13	0.747	1.9
2	2.09	0.868	2.25
3	1.735	0.17	2.34
4	2.18	0.133	2.47
5	1.35	0.228	2.57
6	2.15	0.148	2.66
7	2.047	0.48	2.74
8	0.808	0.39	2.76
9	0.32	3.71	1.38
10	1.78	2.51	1.79
11	3.82	0.9	2.49
12	4.8	3.35	2.16
13	1.87	1.36	1.25
14	0.16	2.72	3.55

Absolute orientation angles were obtained at each instance for each HMD. A long data sequence was used for testing and a few instances are shown in Figure 11.1(c). Table 11.1 presents the absolute error in degree (θ_x,θ_y,θ_z) for each instance. The results are encouraging and angles are very close to the ground truth. For our dataset, we found the mean error to be 2.06° degrees with standard deviation of 1.87°.

11.2 Conclusion

We have successfully demonstrated a novel approach to recover dynamic network topology for configuring a MR environment. Each camera or HMD, having a disjoint FoV, is assumed to undergo a general motion. Our contribution includes computing the relative rotation matrix between N cameras using only vertical vanishing point; and calculating the $\mathbf{H}_{\mathbf{i,j}}^{\infty}$ for non-overlapping cameras and using it to obtain absolute rotation of each camera with respect to a common world coordinate system in a MR environment. Thus, instead of expensive tracking and positioning systems that are currently being used in VR environments, the proposed method does the same task satisfactorily with inexpensive cameras. We successfully demonstrate the proposed method on several sequences.

CHAPTER 12
CONCLUSIONS

In this thesis, we have successfully demonstrated a novel approach to self-calibrate a dynamic camera network. Each camera, possibly having a disjoint FoV, can be permitted to undergo a general motion. Such a network could be, for instance, deployed for surveillance applications comprising of both stationary PTZ cameras and cameras mounted on a roaming security or reconnaissance vehicles (e.g. [CM03]). Another application could be in an urban battlefield setting with soldiers carrying head mounted cameras.

Our contribution includes (i) a global linear solution to self-calibrate a moving camera in the dynamic network using only the fundamental matrix, (ii) a camera calibration based on scene constraints (i.e. vanishing points and vanishing lines) by enforcing new constraints on the IAC, (iii) calibrating a PTZ camera from only two images, (iv) calibrate a camera observing shadow trajectories, (v) using only pedestrians for camera calibration, (vi) computing the relative rotation matrix between N cameras using only vertical vanishing point, and (vii) calculating the $\mathbf{H}_{\mathbf{i},\mathbf{j}}^{\infty}$ for non-overlapping cameras and using it to obtain absolute rotation of each camera with respect to a common world coordinate system without overlapping FoV. In addition, we demonstrated applications of our method (i) to configure a network of HMDs in a MR environment, (ii) to perform surveillance by constructing a path model based on behavior of the observed objects in a scene, and (iii) to estimate the GPS coordinates of the camera using only shadow trajectories of objects in the scene. We have successfully demonstrated the proposed method on several sequences and discussed possible degenerate configurations. The proposed camera calibration and network calibration technique are tested on synthetic as well as on real data. Encouraging results indicate the

applicability of the proposed system.

LIST OF REFERENCES

[AB04] M. Antone and M. Bosse. "Calibration of outdoor cameras from cast shadows." In *Proc. IEEE Int. Conf. Systems, Man and Cybernetics*, pp. 3040–3045, 2004.

[AHR01] L. De Agapito, E. Hayman, and I. Reid. "Self-calibration of rotating and zooming cameras." *Int. J. Comput. Vision*, **45**(2):107–127, 2001.

[AK71] Y. I. Abdel-Aziz and H. M. Karara. "Direct linear transformation into object space coordinates in close-range photogrammetry." 1971.

[AZH96] M. Armstrong, A. Zisserman, and R.I. Hartley. "Self-Calibration from Image Triplets." In *Proc. ECCV*, pp. 3–16, 1996.

[BA96] Michael J. Black and P. Anandan. "The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields." *Journal of Computer Vision and Image Understanding*, **63**(1):75–104, January 1996.

[BA03] P. Baker and Y. Aloimonos. "Calibration of a multicamera network." In *Proc. of IEEE Workshop on Omnidirectional Vision and Camera Networks*, 2003.

[BGK96] M. Bober, N. Georgis, and J.V. Kittler. "On Accurate and Robust Estimation of Fundamental Matrix." p. Poster Session 2, 1996.

[BMV99] Jeffrey E. Boyd, Jean Meloche, and Y. Vardi. "Statistical Tracking in Video Traffic Surveillanc." In *International Conference on Computer Vision (ICCV)*, 1999.

[BP98] J. Bouguet and P. Perona. "3D Photography on Your Desk." In *Proc. ICCV*, pp. 43–50, 1998.

[BR97] A. Basu and K. Ravi. "Active camera calibration using pan, tilt and roll." *IEEE Transactions on Systems, Man and Cybernetics, Part B*, **27**(3):559–566, January 1997.

[BZ99] C. Baillard and A. Zisserman. "Automatic Reconstruction of Piecewise Planar Models from Multiple Views." pp. II: 559–565, 1999.

[CBP05] C. Colombo, A.D. Bimbo, and F. Pernici. "Metric 3D Reconstruction and Texture Acquisition of Surfaces of Revolution from a Single Uncalibrated View." *IEEE Trans. Pattern Anal. Mach. Intell.*, **27**(1):99–114, 2005.

[CDR99] R. Cipolla, T. Drummond, and D. Robertson. "Camera calibration from vanishing points in images of architectural scenes." In *Proc. of BMVC*, pp. 382–391, 1999.

[CF04a] X. Cao and H. Foroosh. "Camera Calibration Without Metric Information Using 1D Objects." In *Proc. IEEE ICIP*, pp. 1349–1352, 2004.

[CF04b] Xiaochun Cao and Hassan Foroosh. "Simple Calibration Without Metric Information Using an Isoceles Trapezoid." In *Proc. Int. Conf. Pattern Recognition, ICPR'04*, Cambridge, UK, August 2004.

[CF06] X. Cao and H. Foroosh. "Camera Calibration and Light Source Orientation from Solar Shadows." *Journal of Computer Vision and Image Understanding (CVIU)*, **105**:60–72, 2006.

[CM03] J. Casper and R.R. Murphy. "Human-robot interactions during the robot-assisted urban search and rescue response at the World Trade Center." *IEEE Transactions on Systems, Man and Cybernetics, Part B*, **33**(3):367–385, June 2003.

[Cox74] H. S. M. Coxeter. *Projective Geometry*. University of Toronto Press, 1974.

[Cre85] Luigi Cremmona. *Elements of Projective Geometry*. Oxford University Press, 1885.

[CRM03] D. Comaniciu, V. Ramesh, and P. Meer. "Kernel-Based Object Tracking." *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **25**(5):564–575, 2003.

[CRZ99] A. Criminisi, I. Reid, and A. Zisserman. "A Plane Measuring Device." *Image and Vision Computing*, **17**(8):625–634, 1999.

[CRZ00] A. Criminisi, I. Reid, and A. Zisserman. "Single View Metrology." *Int. J. Comput. Vision*, **40**(2):123–148, 2000.

[CS05] X. Cao and M. Shah. "Camera Calibration and Light Source Estimation from Images with Shadows." In *Proc. IEEE CVPR*, pp. 918–923, 2005.

[CSS05] X. Cao, Y. Shen, M. Shah, and H. Foroosh. "Single View Compositing with Shadows." *The Visual Computer*, **21**(8):639–648, 2005.

[CT90] B. Caprile and V. Torre. "Using Vanishing Points for Camera Calibration." *Int. J. Comput. Vision*, **4**(2):127–140, 1990.

[CT98] Guan-Yu Chen and Wen-Hsiang Tsai. "An incremental-learning-by-navigation approach to vision-based autonomous land vehicle guidance in indoor environments using vertical line information and multiweighted generalized Hough transform technique." *IEEE Transactions on Systems, Man and Cybernetics, Part B*, **28**(5):740 – 748, Oct 1998.

[CT99] Robert Collins and Yanghai Tsin. "Calibration of an Outdoor Active Camera System." In *IEEE Computer Vision and Pattern Recognition (CVPR '99)*, pp. 528 – 534, June 1999.

[CT04] L. Cadman and T. Tjahjadi. "Efficient three-dimensional metric object modeling from uncalibrated image sequences." *IEEE Transactions on Systems, Man and Cybernetics, Part B*, **34**(2):856–876, April 2004.

[CW06] Yaron Caspi and Michael Werman. "Vertical Parallax from Moving Shadows." In *Proc. CVPR*, pp. 2309–2315, 2006.

[CZZ97] G. Csurka, C. Zeller, Z.Y. Zhang, and O.D. Faugeras. "Characterizing the Uncertainty of the Fundamental Matrix." **68**(1):18–36, October 1997.

[DDZ01] W.E. Dixon, D.M. Dawson, E. Zergeroglu, and A. Behal. "Adaptive tracking control of a wheeled mobile robot via an uncalibrated camera system." *IEEE Transactions on Systems, Man and Cybernetics, Part B*, **31**(3):341–352, June 2001.

[DF95] F. Devernay and O. Faugeras. "Automatic calibration and removal of distortion from scenes of structured environments." In *SPIE*, volume 2567, San Diego, CA, July 1995.

[Fau92] O. Faugeras. "What can be seen in three dimensions with an uncalibrated stereo rig?" In *Proc. ECCV*, pp. 563–578, 1992.

[FK03] J.M. Frahm and R. Koch. "Camera Calibration with Known Rotation." In *Proc. IEEE ICCV*, pp. 1418–1425, 2003.

[FL01] Olivier Faugeras and Quang-Tuan Luong. *The Geometry of Multiple Images*. Oxford University Press, 2001.

[FLM92] O. Faugeras, T. Luong, and S. Maybank. "Camera self-calibration: theory and experiments." In *Proc. of ECCV*, pp. 321–334, 1992.

[GL89] G.H. Golub and C.F. Van Loan. *Matrix Computations*. John Hopkins Press, 1989.

[GP00] P. Gurdjos and R Payrissat. "Recovering the vanishing self-polar triangle from a single view of a planar pattern." pp. 756–759, 2000.

[Gro] Christopher Gronbeck. "SunAngle software (www.susdesign.com/sunangle/).".

[GS03] P. Gurdjos and P. Sturm. "Methods and Geometry for Plane-Based Self-Calibration." In *Proc. IEEE CVPR*, pp. 491–496, 2003.

[GSR98] W.E.L. Grimson, C. Stauffer, R. Romano, and L. Lee. "Using Adaptive Tracking to Classify and Monitor Activities in a Site." In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 1998.

[HA97] A. Heyden and K. Astrom. "Euclidean reconstruction from image sequences with varying and unknown focal length and principal point." In *Proc. IEEE CVPR*, pp. 438–443, 1997.

[HA99] A. Heyden and K. Astrom. "Flexible Calibration: Minimal Cases for Auto-Calibration." In *Proc. IEEE ICCV*, pp. 350–355, 1999.

[Har] http://www.robots.ox.ac.uk/˜az/HZbook/HZfigures.html.

[Har92] R. I. Hartley. "Estimation of Relative Camear Positions for Uncalibrated Cameras." In *Proc. ECCV*, pp. 579–587, 1992.

[Har94] R. I. Hartley. "Self-calibration from multiple views with a rotating camera." In *Proc. ECCV*, pp. 471–478, 1994.

[Har97] R. I. Hartley. "Self-Calibration of Stationary Cameras." *Int. J. Comput. Vision*, **22**(1):5–23, 1997.

[Har98] R.I. Hartley. "Minimizing algebraic error in geometric estimation problems." In *Proc. of ICCV*, pp. 469–476, 1998.

[HB06] Adlane Habed and Boubakeur Boufama. "Camera self-calibration from bivariate polynomial equations and the coplanarity constraint." *to appear: Image and Vision Computing (IVC)*, 2006.

[Hei00] J. Heikkila. "Geometric camera calibration using circular control points." *IEEE Trans. Pattern Anal. Mach. Intell.*, **22**(10):1066–1077, 2000.

[Her67] A.P. Herbert. *Sundials Old and New*. Methuen & Co. Ltd, 1967.

[HHA99] R. I. Hartley, Eric Hayman, L. De Agapito, and I. Reid. "Camera calibration and the search for infinity." In *Proc. IEEE ICCV*, pp. 510–517, 1999.

[HHZ06] Weiming Hu, Min Hu, Xue Zhou, Jianguang Lou, Tieniu Tan, and Steve Maybank. "Principal Axis-Based Correspondence between Multiple Cameras for People Tracking." *IEEE Trans. Pattern Anal. Mach. Intell.*, **28**(4):663, 2006.

[HJL89] R.M. Haralick, H. Joo, C. Lee, X. Zhuang, V.G. Vaidya, and M.B. Kim. "Pose estimation from corresponding point data." *IEEE Transactions on Systems, Man and Cybernetics, Part B*, **19**(6):1426–1446, Nov 1989.

[HZ04] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[III94] Frederick W. Sawyer III. "A Three-Point Sundial Construction." *Bulletin of the British Sundial Society*, **94**(1):22–29, Feb 1994.

[JAS07] Imran Junejo, Nazim Ashraf, Yuping Shen, and Hassan Foroosh. "Robust Auto-Calibration Using Fundamental Matrices Induced by Pedestrians." In *IEEE International Conference on Image Processing (ICIP), 2007.*, 2007.

[Jay04] Christopher O. Jaynes. "Multi-view calibration from planar motion trajectories." *Image Vision Computing*, **22**(7):535–550, 2004.

[JCF06a] Imran Junejo, Xiaochun Cao, and Hassan Foroosh. "Calibrating Freely Moving Cameras." In *18th International Conference on Pattern Recognition (ICPR).*, 2006.

[JCF06b] Imran Junejo, Xiaochun Cao, and Hassan Foroosh. "Configuring Mixed Reality Environment." In *18th International Conference on Pattern Recognition (ICPR).*, 2006.

[JCF06c] Imran Junejo, Xiaochun Cao, and Hassan Foroosh. "Geometry of a Non-Overlapping Multi-Camera Network." In *5th IEEE International Conference on Advanced Video and Signal-based Surveillance (AVSS).*, 2006.

[JCF07] Imran Junejo, Xiaochun Cao, and Hassan Foroosh. "Auto-Configuration of a Dynamic Non-Overlapping Camera Network." *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, August*, **37**, 2007.

[JFa] Imran Junejo and Hassan Foroosh. "Euclidean Path Modeling for Video Surveillance." *at final stage of review at Journal of Image and Vision Computing (IVC).*

[JFb] Imran Junejo and Hassan Foroosh. "Geometrically Optimized PTZ Camera Calibration From Only Two Images." *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) - (Submitted).*

[JF06a] Imran Junejo and Hassan Foroosh. "Dissecting the Image of the Absolute Conic." In *5th IEEE International Conference on Advanced Video and Signal-based Surveillance (AVSS).*, 2006.

[JF06b] Imran Junejo and Hassan Foroosh. "Robust Auto-Calibration from Pedestrians." In *Proceedings of 5th IEEE International Conference on Advanced Video and Signal-based Surveillance (AVSS).*, 2006.

[JF07a] Imran Junejo and Hassan Foroosh. "Calibration of Rotating and Zooming Cameras by Direct Decomposition of Infinite Homography." In *Eleventh IEEE International Conference on Computer Vision (ICCV), 2007 (Submitted)*, 2007.

[JF07b] Imran Junejo and Hassan Foroosh. "Euclidean Path Modeling from Ground and Aerial Views." In *7th IEEE International Workshop on Visual Surveillance with CVPR.*, 2007.

[JF07c] Imran Junejo and Hassan Foroosh. "Using Calibrated Camera for Euclidean Path Modeling." In *IEEE International Conference on Image Processing (ICIP), 2007.*, 2007.

[JF07d] Imran Junejo and Hassan Foroosh. "Where and when are these pictures taken." In *Eleventh IEEE International Conference on Computer Vision (ICCV), 2007 (Submitted)*, 2007.

[JH95] Neil Johnson and David Hogg. "Learning the Distribution of Object Trajectories for Event Recognition." In *Proc. of British Machine Vision Conference (BMVC)*, 1995.

[JJS04] Imran Junejo, Omar Javed, and Mubarak Shah. "Multi Feature Path Modeling for Video Surveillance." In *17th conference of the International Conference on Pattern Recognition (ICPR)*, 2004.

[JRA03] Omar Javed, Zeeshan Rasheed, Orkun Alatas, and Mubarak Shah. "KNIGHTM: A Real Time Surveillance System for Multiple Overlapping and Non-Overlapping Cameras." In *The fourth International Conference on Multimedia and Expo (ICME)*, 2003. Baltimore, Maryland.

[JRS03] Omar Javed, Zeeshan Rasheed, Khurram Shafique, , and Mubarak Shah. "Tracking Across Multiple CamerasWith Disjoint Views." In *The Ninth IEEE International Conference on Computer Vision*, 2003.

[JS02] Omar Javed and Mubarak Shah. "Tracking and Object Classification for Automated Surveillance." In *the seventh European Conference on Computer Vision*, 2002.

[JSS05] Omar Javed, Khurram Shafique, and Mubarak Shah. "Appearance Modeling for Tracking in Multiple Non-overlapping Cameras." In *IEEE CVPR*, 2005.

[KCM03] J. Kang, I. Cohen, and G. Medioni. "Continuous tracking within and across camera streams." In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[Keo02] Eamonn Keogh. "Exact Indexing of Dynamic Time Warping." In *28th International Conference on Very Large Data Bases. Hong Kong*, pp. 406–417, 2002.

[KM05] Nils Krahnstoever and Paulo R. S. Mendonca. "Bayesian Autocalibration for Surveillance." In *Tenth IEEE International Conference on Computer Vision*, 2005.

[Kru13] E. Kruppa. *Zur ermittlung eines objektes aus zwei perspektiven mit innerer orientierung,122:19391948*. Sitz.-Ber. Akad. Wiss., Wien, math. naturw. Abt. IIa,, 1913.

[KS03] Sohaib Khan and Mubarak Shah. "Consistent Labeling of Tracked Objects in Multiple Cameras with Overlapping Fields of View." *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, **25**(10), 2003.

[KTA00] F. Kahl, B. Triggs, and K. Åström. "Critical Motions for Auto-Calibration When Some Intrinsic Parameters Can Vary." *J. Math. Imaging Vis.*, **13**(2):131–146, 2000.

[KZR03] J. Knight, A. Zisserman, and I. Reid. "Linear Auto-Calibration for Ground Plane Motion." In *Proc. IEEE CVPR*, pp. 503–510. IEEE, 2003. Madison, Wisconsin.

[LF96] Q.T. Luong and O.D. Faugeras. "The fundamental matrix: Theory, algorithms, and stability analysis." *Int. J. Comput. Vision*, **17**(1):43–75, 1996.

[Low04] David G. Lowe. "Distinctive image features from scale-invariant keypoints." *International Journal of Computer Vision*, **6**(2):91–110, 2004.

[LZ98] D. Liebowitz and A. Zisserman. "Metric Rectification for Perspective Images of Planes." In *Proc. IEEE CVPR*, pp. 482–488, 1998.

[LZ99] D. Liebowitz and A. Zisserman. "Combining Scene and Auto-Calibration Constraints." In *Proc. IEEE ICCV*, pp. 293–300, 1999.

[LZN02] Fengjun Lv, Tao Zhao, and Ramakant Nevatia. "Self-Calibration of a Camera from Video of a Walking Human." In *IEEE International Conference of Pattern Recognition*, 2002.

[ME02] Dimitrios Makris and Tim Ellis. "Path Detection in Video Surveillance." *Image and Vision Computing Journal (IVC)*, **20**(12):895–903, 2002.

[ME05] D. Makris and T. Ellis. "Learning semantic scene models from observing activity in visual surveillance." *IEEE Transactions on Systems, Man and Cybernetics, Part B*, **35**(3):397–408, June 2005.

[Men01] P. R. S. Mendonça. *Multiview Geometry: Profiles and Self-Calibration*. PhD thesis, University of Cambridge, Cambridge, UK, May 2001.

[MGP96] T. Moons, L.V. Gool, M. Proesmans, and E. Pauwels. "Affine reconstruction from perspective image pairs with a relative object-camera translation in between." *IEEE Trans. Pattern Anal. Mach. Intell.*, **18**(1):77–83, 1996.

[MK95] G. F. McLean and D. Kotturi. "Vanishing point detection by line clustering." *IEEE Trans. Pattern Anal. Mach. Intell.*, **17**(11):1090–1095, 1995.

[MK04] Y. Motai and A. Kak. "An interactive framework for acquiring vision models of 3-D objects from 2-D images." *IEEE Transactions on Systems, Man and Cybernetics, Part B*, **34**(1):566–578, Feb 2004.

[MT04] D. Makris and J. T.J. Ellis. "Bridging the Gaps between Cameras." In *IEEE Conference on Computer Vision and Pattern Recognition CVPR*, 2004.

[PFT88] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical Recipes in C*. Cambridge University Press, 1988.

[PKG99] M. Pollefeys, R. Koch, and L. V. Gool. "Self-Calibration and Metric Reconstruction in Spite of Varying and Unknown Internal Camera Parameters." *Int. J. Comput. Vision*, **32**(1):7–25, 1999.

[SGJ05] Y. Sheikh, A. Gritai, I. Junejo, R. Muise, A. Mahalanobis, and M. Shah. "Establishing a Common View from Multiple Moving Aerial Sensors." In *SPIE Symposium on Defense and Security.*, 2005.

[SGN03] R. Swaminathan, M.D. Grossberg, and S.K. Nayar. "A Perspective on Distortions." In *Proc. IEEE CVPR*, pp. 594–601, 2003.

[SH99] Y. Seo and K. Hong. "About the Self-Calibration of a Rotating and Zooming Camera: Theory and Practice." In *Proc. IEEE ICCV*, pp. 183–189, 1999.

[SH04] Yongduek Seo and Anders Heyden. "Auto-calibration by linear iteration using the DAC equation." *Image and Vision Computing (IVC)*, **22**(11):919–926, 2004.

[Shu99] J. A. Shufelt. "Performance Evaluation and Analysis of Vanishing Point Detection Techniques." *IEEE Trans. Pattern Anal. Mach. Intell.*, **21**(3):282–288, 1999.

[SK79] J. G. Semple and G. T. Kneebone. *Algebraic Projective Geometry*. Oxford Classic Texts in the Physical Sciences, 1979.

[SM98] J. Shi and J. Malik. "Motion Segmentation and Tracking Using Normalized Cuts." In *Proc. IEEE ICCV*, 1998.

[SM00] Jianbo Shi and Jitendra Malik. "Normalized Cuts and Image Segmentation." *IEEE Transactions on Pattern Analysis and Machine Intelligence(PAMI)*, 2000.

[SS06] Yaser Sheikh and Mubarak Shah. "Object Tracking Across Multiple Independently Moving Airborne Cameras." In *IEEE International Conference on Computer Vision, 2005.*, 2006.

[SSK05] Y. Shan, H.S. Sawhney, and R.T. Kumar. "Vehicle Identification between Non-Overlapping Cameras without Direct Feature Matching." pp. I: 378–385, 2005.

[Stu97a] P. Sturm. "Critical Motion Sequences for Monocular Self-Calibration and Uncalibrated Euclidean Reconstruction." In *Proc. IEEE CVPR*, pp. 1100–1105, 1997.

[Stu97b] Peter Sturm. "Self-calibration of a moving zoom-lens camera by pre-previous calibration." *Image and Vision Computing (IVC)*, **15**(8):583–589, 1997.

[Stu99] Peter Sturm. "Critical Motion Sequences for the Self-Calibration of Cameras and Stereo Systems with Variable Focal Length." In *British Machine Vision Conference, Nottingham, England*, pp. 63–72, Sep 1999.

[Tan96] J. Tani. "Model-based learning for mobile robot navigation from the dynamical systems perspective." *IEEE Transactions on Systems, Man and Cybernetics, Part B*, **26**(3):421–436, June 1996.

[TDG05] Kinh Tieu, Gerald Dalley, and W. Eric L. Grimson. "Inference of Non-Overlapping Camera Network Topology by Measuring Statistical Dependence." In *International Conference on Computer Vision*, 2005.

[TMH99] B. Triggs, P. McLauchlan, R. I. Hartley, and A. Fitzgibbon. "Bundle Adjustment — A Modern Synthesis." In *Vision Algorithms: Theory and Practice*, pp. 298–373, 1999.

[Tri97] B. Triggs. "Autocalibration and the Absolute Quadric." In *Proc. IEEE CVPR*, pp. 609–614, 1997.

[Tri98] B. Triggs. "Autocalibration from planar scenes." In *Proc. ECCV*, pp. 89–105, 1998.

[Tsa87] R.Y. Tsai. "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses." *IEEE J. of Robotics and Automation*, **3**(4):323–344, 1987.

[Ver99] Ramin Zabih Vera Kettnaker. "Bayesian Multi-Camera Surveillance." In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 1999.

[VMP04] R. Vidal, Y. Ma, and J. Piazzi. "A New GPCA Algorithm for Clustering Subspaces by Fitting, Differentiating and Dividing Polynomials." In *Proc. IEEE CVPR*, pp. 510–517, 2004.

[Wau73] A.E. Waugh. *Sundials: Their Theory and Construction*. Number ISBN 0-486-22947-5. Dover Publications, Inc., 1973.

[WKS04] Lei Wang, Sing Bing Kang, Heung-Yeung Shum, and Guangyou Xu. "Error Analysis of Pure Rotation-Based Self-Calibration." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **26**(2):275–280, 2004.

[WMC03] K.-Y. Wong, R.S.P. Mendonça, and R. Cipolla. "Camera Calibration from Surfaces of Revolution." *IEEE Trans. Pattern Anal. Mach. Intell.*, **25**(2):147–161, 2003.

[WP05] John Wright and Robert Pless. "Analysis of Persistent Motion Patterns Using the 3D Structure Tensor." In *Proceedings of the IEEE Workshop on Motion and Video Computing*, pp. 14–19, 2005.

[WS94] G. Willson, Reg. and A. Shafer, Steven. "What is the Center of the Image?" Technical Report 11, Nov 1994.

[YB96] B. Yamauchi and R. Beer. "Spatial learning for navigation in dynamic environments." *IEEE Transactions on Systems, Man and Cybernetics, Part B*, **26**(3):496–505, June 1996.

[YWC05] Ying Kin Yu, Kin Hong Wong, and M.M.Y.; Chang. "Pose estimation for augmented reality applications using genetic algorithm." *IEEE Transactions on Systems, Man and Cybernetics, Part B*, **35**(6):1295–1301, Dec 2005.

[ZAK05] Tao Zhao, M. Aggarwal, R. Kumar, and H. Sawhney. "Real-time wide area multi-camera stereo tracking." In *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2005.

[ZDF95] Zhengyou Zhang, Rachid Deriche, Olivier D. Faugeras, and Quang-Tuan Luong. "A Robust Technique for Matching two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry." *Artificial Intelligence*, **78**(1-2):87–119, 1995.

[ZH94] Xinhua Zhuang and Yan Huang. "Robust 3-D-3-D pose estimation." *IEEE Transactions on Systems, Man and Cybernetics, Part B*, **16**(8):818–824, Aug 1994.

[Zha00] Z. Zhang. "A Flexible New Technique for Camera Calibration." *IEEE Trans. Pattern Anal. Mach. Intell.*, **22**(11):1330–1334, 2000.

[Zha02] Z. Zhang. "Camera Calibration with One-Dimensional Objects." In *Proc. ECCV*, pp. 161–174, 2002.

[ZLA98] A. Zisserman, D. Liebowitz, and M. Armstrong. "Resolving Ambiguities in Auto-Calibration." *Phil. Trans. Royal Soc. London A*, **356**(1740):1193–1211, 1998.