Robust Camera Pose Recovery Using Stochastic Geometry

by

Matthew E. Antone

Bachelor of Science, Electrical Engineering
Massachusetts Institute of Technology, 1996

Master of Engineering, Electrical Engineering and Computer Science
Massachusetts Institute of Technology, 1996

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
at the
Massachusetts Institute of Technology

February 2001

©2001 Massachusetts Institute of Technology
All Rights Reserved

Signature of Author: . . . . . .
Department of Electrical Engineering and Computer Science
February 13, 2001

Certified by: . . . . . .
Seth Teller
Associate Professor of Computer Science and Engineering
Thesis Supervisor

Accepted by: . . . . . .
Arthur C. Smith
Chairman, Committee on Graduate Students
Department of Electrical Engineering and Computer Science

Robust Camera Pose Recovery Using Stochastic Geometry

by

Matthew E. Antone

Submitted to the Department of Electrical Engineering and Computer Science
on February 13, 2001
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

Abstract

The objective of three-dimensional (3-D) machine vision is to infer geometric properties (shape, dimensions) and photometric attributes (color, texture, reflectance) from a set of two-dimensional images. Such vision tasks rely on accurate camera *calibration*, that is, estimates of the camera's *intrinsic parameters*, such as focal length, principal point, and radial lens distortion, and *extrinsic parameters*—orientation, position, and scale relative to a fixed frame of reference.

This thesis introduces methods for automatic recovery of precise extrinsic camera pose among a large set of images, assuming that accurate intrinsic parameters and rough estimates of extrinsic parameters are available. Although the motivating application is metric 3-D reconstruction of urban environments from pose-annotated hemispherical imagery, few domain-specific restrictions are required or imposed.

Orientation is recovered independently of position via the detection and optimal alignment of translation-invariant image features (*vanishing points*). Known orientations constrain pair-wise epipolar geometry, reducing the determination of local translational offsets to a simple linear form. Local motion estimates are assembled into a global constraint set to determine camera positions with respect to a common coordinate frame. The final result is an assignment of accurate pose, along with its uncertainty, to every relevant camera.

The methods developed in this work are notable in several respects. First, they scale linearly in space and time usage with the number of input images. Second, they robustly and automatically achieve global optimality even with poorly known initial pose. Third, they represent every geometric quantity stochastically, as a sample from a probability distribution defined over an appropriate parameter space; explicit measures of uncertainty are produced alongside every parameter estimate and propagated throughout. Finally, they estimate feature correspondence probabilistically, and therefore neither require nor produce explicit correspondence.

A theoretical framework for the appropriate geometric models and reasoning tasks is presented, followed by descriptions of methods for optimal recovery of camera pose within this framework. Performance is analyzed in simulation and on large sets (tens of thousands) of real images of urban environments. Experiments show that the proposed methods can produce pose estimates consistent to roughly 0.1° (2 milliradians) of orientation and 5 centimeters of position, as well as epipolar registration within 4 pixels, across wide-baseline data sets spanning several hundred meters, outperforming bundle adjustment via manual correspondence for the same input data.

Thesis Supervisor: Seth Teller
Title: Associate Professor of Computer Science and Engineering

Acknowledgements

This work would not have been possible without the help and influence of many others.

First and foremost, I would like to thank Seth Teller for being an absolutely fantastic thesis supervisor and mentor. He devoted many hours, stayed many nights, and offered invaluable words of guidance, encouragement, praise, and criticism. Seth never failed to give prompt, thorough feedback on ideas and written works, or to quickly devise efficient algorithms for just about any task.

I'd also like to thank several other members of the department. My thesis committee, Sanjoy Mitter and Berthold Horn, came through more than once on very short notice. My academic advisor, Alan Willsky, nudged me in the right directions and answered all my questions about the graduate program. Dimitri Bertsekas allowed me to spread some knowledge, and my students, in turn, helped me keep perspective. Sandy Wells and Alan Edelman set aside time to discuss my work on several occasions. I extend a special thanks to Marilyn Pierce, for putting up with my constant requests and for pushing the deadlines back as far as they would go.

It was my distinct pleasure to work with a brilliant and multi-talented group of individuals. I learned a great deal from my colleagues in the City Project, who gave me their time, advice, and unquestioning support. Neel Master has an uncanny grasp of this complicated system and of the vast body of vision literature. Mike Bosse and Manish Jethwa provided sounding boards for ideas and brainstormed with me to solve many problems.

Several people outside my immediate circle were also quite helpful. Christopher Bingham at the Universitiy of Minnesota, John Kent at the University of Leeds, and Robert Collins at Carnegie Mellon University were kind enough to share code fragments for various numerical calculations involving spherical distributions. Frank Dellaert at CMU and Anand Rangarajan at Yale, experts on probabilistic correspondence, offered relevant information and advice. Tony Jebara gave me a crash course in Bayesian inference and discussed ideas about error models.

All the folks in the Graphics Group made this an enjoyable place to work. Bryt Bradley and Adel Hanna saved me more times and in more ways than I can count. My office mates over the years patiently withstood my guitar playing, and provided diversions—both technical and recreational—that kept me sane. Max Chen, Mok Oh, Manish Jethwa, and Matt Peters, aside from keeping me entertained and letting me in on their work, became my lifting partners at the gym (the M stands for Muscles!).

I'm especially grateful to Yuli Friedman, whose concern and true friendship kept me going, and who, more than anyone else, looked out for my well-being. He made sure that I slept once in a while, and selflessly provided the basic elements of survival: shelter, food, and TiVo.

Finally, to my friends, who have remained my friends even through my disappearance into the depths of thesitude:

Onward and upward.

About the Author

Matthew E. Antone received his Bachelor of Science degree in electrical engineering from the Massachusetts Institute of Technology in June, 1996. During his senior year he was inducted into the Tau Beta Pi, Eta Kappa Nu, and Sigma Xi honor societies. Matthew developed a Master's thesis at the MIT Media Laboratory and received his Master of Engineering degree in electrical engineering and computer science in September of 1996. He was an Interval Research Fellow in 1997, and taught a course in probabilistic systems analysis in 1999 and 2000. During his academic career Matthew also consulted for several Boston-area companies. He is a member of the IEEE and ACM, and his research interests include machine vision, signal and image processing, system theory, stochastic modeling, and optimization.

To my father, Adil, and my mother, Ann,
and to my brothers, Tim, John, and Chris,
for their patience, love, and understanding.

CONTENTS

1 Introduction **21**
- 1.1 The Vision Problem . . . 21
- 1.2 Image Formation . . . 23
  - 1.2.1 Pinhole Lens . . . 23
  - 1.2.2 Image Surfaces . . . 24
  - 1.2.3 Camera Parameters . . . 25
  - 1.2.4 Projection and Homographies . . . 26
  - 1.2.5 Sampling the Image . . . 27
- 1.3 Image Features . . . 28
  - 1.3.1 Image Space Tokens . . . 28
  - 1.3.2 Projective Tokens . . . 30
- 1.4 Multi-Camera 3-D Vision . . . 32
  - 1.4.1 Multi-View Geometry . . . 32
  - 1.4.2 Structure and Motion . . . 33
  - 1.4.3 Correspondence . . . 33
- 1.5 Thesis Overview . . . 34
  - 1.5.1 Objectives . . . 34
  - 1.5.2 Assumptions and Restrictions . . . 35
  - 1.5.3 Outline of Methods . . . 35
  - 1.5.4 System Properties . . . 36
  - 1.5.5 Dissertation Roadmap . . . 37

2 Context **39**
- 2.1 Related Work . . . 40
  - 2.1.1 Direct Pose Measurement . . . 40
  - 2.1.2 Interactive Systems . . . 41
  - 2.1.3 Controlled Calibration . . . 41
  - 2.1.4 Structure from Motion . . . 42
  - 2.1.5 Projective Reconstruction . . . 43

2.1.6 Correspondence Methods . . . 44
2.1.7 Stochastic Geometry . . . 45
2.2 The City Project . . . 46
2.2.1 Data Acquisition . . . 46
2.2.2 Hemispherical Image Registration . . . 47
2.2.3 Manual Pose Refinement . . . 48
2.2.4 Geometric Reconstruction . . . 48
2.2.5 Motivations for Automated Registration . . . 49
2.3 Contributions . . . 51
2.3.1 Uncertainty Models . . . 51
2.3.2 Hough Transform Methods . . . 51
2.3.3 Vanishing Point Estimation . . . 52
2.3.4 Rotational Registration . . . 52
2.3.5 Translational Registration . . . 52
2.3.6 End-to-End Results . . . 53

3 Stochastic Geometry **55**
3.1 Elementary Probability Theory . . . 56
3.1.1 Events . . . 56
3.1.2 Random Variables . . . 57
3.1.3 Conditioning . . . 58
3.1.4 Laws of Large Numbers . . . 58
3.2 Statistical Inference . . . 59
3.2.1 Maximum Likelihood Estimation . . . 60
3.2.2 Bayesian Estimation . . . 61
3.3 Projective Feature Uncertainty . . . 62
3.3.1 Past Work in Projective Uncertainty . . . 63
3.3.2 Bingham's Distribution . . . 64
3.3.3 Feature Representations . . . 65
3.4 Pose Uncertainty . . . 66
3.4.1 Orientation . . . 66
3.4.2 Position . . . 67
3.5 Uncertainty of Derived Quantities . . . 67
3.5.1 Gradient Pixels . . . 68
3.5.2 Duality of Projective Uncertainty . . . 69
3.5.3 Projective Data Fusion . . . 70
3.5.4 Cross Products . . . 71

4 Basic Algorithms **73**
4.1 Hough Transform . . . 73
4.1.1 Definition . . . 74
4.1.2 Properties . . . 74
4.1.3 Implementation . . . 75
4.1.4 Further Difficulties . . . 75
4.1.5 Parameterization of the Sphere . . . 76

4.1.6 Proximity Function . . . 77
4.1.7 Detecting Peaks . . . 78
4.1.8 Peak Uncertainty . . . 80
4.2 Expectation Maximization . . . 81
4.2.1 Mixture Models . . . 81
4.2.2 Derivation . . . 82
4.2.3 Expectation Step . . . 83
4.2.4 Maximization Step . . . 83
4.2.5 Robust Clustering . . . 84
4.2.6 Initialization . . . 84
4.3 Markov Chain Monte Carlo . . . 84
4.3.1 Parameters as States . . . 85
4.3.2 Metropolis Algorithm . . . 85
4.3.3 Simulated Annealing . . . 86
4.4 Adjacency Computation . . . 86
4.4.1 Using Camera Pose . . . 87
4.4.2 Approximate Determination of Adjacency . . . 87

5 Orientation Recovery 89
5.1 Vanishing Points . . . 90
5.1.1 Geometry . . . 90
5.1.2 Translation Invariance . . . 91
5.2 Related Vanishing Point Methods . . . 92
5.2.1 Rotational Pose from Vanishing Points . . . 93
5.2.2 Vanishing Point Estimation . . . 93
5.3 Single-Camera Formulation . . . 94
5.3.1 Mixture Model for Vanishing Points . . . 95
5.3.2 E-Step . . . 95
5.3.3 M-Step . . . 96
5.3.4 Initialization . . . 96
5.4 Two-Camera Registration . . . 97
5.4.1 Classical Pair Registration . . . 97
5.4.2 Stochastic Pair Registration . . . 98
5.4.3 Two-Camera Vanishing Point Correspondence . . . 99
5.4.4 Correspondence Ambiguity . . . 99
5.5 Multi-Camera Registration . . . 101
5.5.1 EM for Multi-Camera Registration . . . 102
5.5.2 EM for Multi-Camera Correspondence . . . 103
5.5.3 Construction and Initialization . . . 104
5.5.4 Merging and Separating Clusters . . . 105

6 Position Recovery 107
6.1 Two-Camera Translation Geometry . . . 108
6.1.1 Epipolar Geometry with Known Orientation . . . 108
6.1.2 Geometric Constraints on Correspondence . . . 109

6.2 Inference of Translation Direction . . . 110
6.2.1 Motion Direction from Known Correspondence . . . 111
6.2.2 Motion Direction from Probabilistic Correspondence . . . 112
6.2.3 Weight Variables . . . 113
6.2.4 Obtaining a Prior Distribution . . . 114
6.3 Monte Carlo Expectation Maximization . . . 117
6.3.1 Structure from Motion without Correspondence . . . 117
6.3.2 Counting Correspondence Sets . . . 118
6.3.3 Previous Expectation Maximization Methods . . . 119
6.3.4 Sampling the Posterior Distribution . . . 119
6.3.5 Match Perturbations . . . 121
6.3.6 Efficient Sampling . . . 121
6.3.7 Comparison to Consensus Sampling . . . 123
6.4 Multi-Camera Method . . . 123
6.4.1 Constraint Equations . . . 123
6.4.2 Constraints with Uncertainty . . . 125
6.4.3 Uncertainty in Final Positions . . . 126
6.5 Metric Registration . . . 126
6.5.1 Absolute Orientation . . . 127
6.5.2 Transforming Uncertainty . . . 128
6.6 Summary of Position Recovery . . . 129

7 Experiments 131
7.1 Simulation . . . 132
7.1.1 Projective Data Fusion Performance . . . 132
7.1.2 Single-Camera Vanishing Point Estimation . . . 134
7.1.3 Two-Camera Rotational Pose . . . 136
7.1.4 Multi-Camera Orientation . . . 136
7.1.5 Two-Camera Translational Pose . . . 138
7.1.6 Global Registration . . . 141
7.2 Real Data . . . 142
7.2.1 TechSquare Data Set (81 Nodes) . . . 143
7.2.2 GreenBuilding Data Set (30 nodes) . . . 147
7.2.3 AmesCourt Data Set (100 nodes) . . . 150
7.2.4 Omnidirectional Images . . . 152

8 Conclusions 153
8.1 Discussion . . . 153
8.1.1 System Benefits . . . 154
8.1.2 System Limitations . . . 154
8.2 Future Work . . . 156
8.2.1 Optimizations . . . 156
8.2.2 Improvements to Uncertainty Models . . . 156
8.2.3 Internal Calibration . . . 158
8.2.4 Exclusive Use of Line Features . . . 159

8.2.5 Domain-Specific Constraints . . . 161
8.2.6 Correspondence Propagation . . . 162
8.2.7 Extension to Real-Time . . . 163
8.2.8 Topological Analysis . . . 164
8.3 Summary . . . 165

A Quaternions **167**
A.1 Definition . . . 167
A.2 Conversions . . . 168
A.3 Transformations . . . 168
A.4 Comparison of Relative Rotations . . . 169
A.5 Optimal Deterministic Rotation . . . 169
A.6 Incorporation of Uncertainty . . . 170

B Bingham's Distribution **173**
B.1 Formulation . . . 173
B.2 Parameter Computation . . . 174
B.3 Confidence Limits . . . 175

FIGURES

1-1 Projection Models 23
1-2 Planar Pinhole Projection 24
1-3 Imaging Coordinate Systems 26
1-4 Ray Reprojection 27
1-5 Hierarchy of Image Features 28
1-6 Euclidean Lines and Points 29
1-7 Examples of Image Features 30
1-8 Projective Image Features 31
1-9 Structure, Motion, and Correspondence 32
1-10 Pose Recovery System Overview 36

2-1 The City Project 46
2-2 Argus Data Acquisition 47
2-3 Creation of Image Mosaics 48
2-4 Semi-Automatic Pose Refinement 49
2-5 Geometric Reconstruction 50
2-6 Correspondence Ambiguities 50

3-1 Approximations to Spherical Uncertainty 63
3-2 Bingham's Distribution on the Sphere 65
3-3 Projective Transformations 67
3-4 Gradient Pixel Uncertainty 68

4-1 Contours for Line Detection 75
4-2 Discretization of the Sphere 77
4-3 Local Maxima 79
4-4 Spatial Adjacency in Cubic Discretization 80
4-5 Camera Adjacency 88

5-1 Rotational Registration 90
5-2 Geometry of Vanishing Points 91

5-3 Vanishing Point Estimation . . . 94
5-4 Pair Couplets . . . 100
5-5 Two Solutions to Optimal Rotation . . . 100
5-6 Vanishing Point Correspondence Ambiguity . . . 101
5-7 Global Orientation Recovery . . . 102

6-1 Translational Registration . . . 108
6-2 Pair Translation Geometry . . . 109
6-3 Line Constraints . . . 111
6-4 Direction Constraints . . . 112
6-5 Correspondence Ambiguity . . . 113
6-6 Augmented Match Matrix . . . 114
6-7 Hough Transform for Baseline Estimation . . . 115
6-8 False Hough Transform Peaks . . . 116
6-9 Row and Column Swaps . . . 121
6-10 Split and Merge Perturbations . . . 122
6-11 Assembling Translation Directions . . . 124
6-12 Topological Degeneracies . . . 125
6-13 Metric Registration Process . . . 127

7-1 Deviation of Estimated Axes . . . 133
7-2 Distribution Coherence for Data Fusion . . . 133
7-3 Percentage of Vanishing Points Detected . . . 134
7-4 EM Vanishing Point Error . . . 135
7-5 Comparison of Two-Camera Rotation Methods . . . 136
7-6 Multi-Camera Orientation Performance . . . 137
7-7 Baseline Estimate Accuracy . . . 138
7-8 MCEM State Matrix Evolution . . . 139
7-9 MCEM Baseline Comparison . . . 140
7-10 Global Registration Results . . . 141
7-11 TechSquare Node Configuration . . . 143
7-12 TechSquare Epipolar Geometry Comparison I . . . 145
7-13 TechSquare Epipolar Geometry Comparison II . . . 146
7-14 GreenBuilding Node Configuration . . . 147
7-15 GreenBuilding Data Corrections . . . 148
7-16 GreenBuilding Epipolar Geometry Comparison . . . 149
7-17 GreenBuilding Epipolar Geometry . . . 149
7-18 AmesCourt Node Configuration . . . 150
7-19 AmesCourt Epipolar Geometry . . . 151
7-20 Hough Transform Peak Coherence . . . 152

8-1 Limitations of Adjacency Computation . . . 155
8-2 Earth Curvature . . . 155
8-3 Internal Parameters from Vanishing Points . . . 158
8-4 Two Dimensional Structure from Motion . . . 160

8-5 Camera Triplet Geometry 161
8-6 Three-Way Match Constraint Surface 162
8-7 Plane Orientations 163
8-8 Topological Holes 164

TABLES

7-1 TechSquare Data Size 143
7-2 TechSquare Computation Times 144
7-3 TechSquare Error Assessment 144
7-4 GreenBuilding Data Size 147
7-5 GreenBuilding Computation Times 148
7-6 GreenBuilding Error Assessment 148
7-7 AmesCourt Data Size 150
7-8 AmesCourt Computation Times 151
7-9 AmesCourt Error Assessment 151

CHAPTER 1

Introduction

In recent years, the study of 3-D machine vision has received a great deal of attention from both research and industrial communities. Vision systems process one or more input images from one or more visual sensors and attempt to recover interesting properties of the scenes viewed by the sensors, such as geometry (e.g. shape, distance, and dimensions) and photometric properties (e.g. color, texture, and reflectance). Applications of such technology are numerous, including physical simulation, virtual reality, cinematic special effects, architecture, and urban planning.

Accurate camera calibration is crucial to every 3-D vision task. This thesis concerns the recovery of camera pose, or *registration*, which is the estimation of world-relative positions and orientations of the cameras used to acquire a set of images. In the chapters that follow, a working system, quite general and applicable to real-world data, is proposed and demonstrated. Existing pose recovery methods are numerous, to say the least, so both similarities to and distinctions from these methods are discussed.

The remainder of this chapter reviews the fundamentals of conventional 3-D vision, beginning with the mechanics of basic image projection and formation of features, and followed by the inference of 3-D information from these features. The chapter proceeds with a presentation of the thesis and an overview of its core concepts, and concludes with an outline of the dissertation.

1.1 The Vision Problem

Machine vision encompasses a wide variety of tasks, but its general goal is enabling machines to "see", much as living organisms do—that is, to interpret in some sense the visual stimuli with which they are presented. In fact many analogies may be drawn between biological and artificial visual systems. Inputs to such systems primarily consist of cameras (the "eyes"), but may also include a variety of secondary devices such as inertial sensors, tactile feedback systems, or sonar (other "senses"). The visual system itself, usually consisting of a processor and a set of algorithms

(the "brain"), interprets these sensory inputs and constructs meaningful information from them.

To date, the "general vision problem" remains unsolved. One reason is that the problem is somewhat ill-posed; "sight" is a subjective term, and is applied rather loosely to specialized systems in an application-dependent context. Some define it as the low-level processing of visual input for such tasks as feature extraction, motion tracking, and target recognition, while others apply the term to higher-level tasks such as three-dimensional (3-D) reconstruction, autonomous navigation, or gesture interpretation. In truth, biological vision does not consist simply of raw processing of neural impulses from the retina, nor is it specialized to a single high-level task. Instead, it involves complex interactions among a tightly integrated collection of specialized neural subsystems interconnected by higher-level brain function and sensory feedback. Other systems such as memory and proprioception are intimately combined with processing in the visual cortex to perform vision-related tasks.

Artificial vision systems thus cannot be developed in isolation. They must consist of a set of generalized, interconnected tools, and must blur the distinction between the ability to see (e.g. "a small white region on a large red region") and the ability to understand (e.g. "a light reflected on the surface of an apple"). However, formulation of a general system is a daunting proposition: to enable true sight is in some sense to enable true cognition, somewhat of a holy grail to modern research. For this reason, the field of machine vision is closely related to those of artificial intelligence and machine learning.

In order to formulate more practical vision systems, application domains can be restricted to specific tasks. However, "black box" systems continue to elude researchers even for restricted tasks, because these tasks themselves consist of tightly-coupled subproblems, and are severely underconstrained without the external knowledge normally supplied by learning, context, and non-visual sensory input. Domain specificity is thus a double-edged sword: on the one hand it reduces complexity, but on the other it excludes critical context-dependent information. Such information can be supplied in the form of explicit constraints; the difficulty lies in drawing the line between that which is explicitly specified by a given model and that which the system is to infer. Too much constraint becomes overly restrictive, limiting application of the system to very specialized situations; overconstrained models often do not possess sufficient parameters to extract information of interest. Too little constraint makes problems inherently underspecified or intractably complex.

Another reason that vision problems remain unsolved is that modern machines are ill-suited for biological vision tasks: an artificial brain differs vastly from a real brain, even that of an insect. Despite attempts to study and simulate biological "hardware" (which itself is not fully understood), computers remain fundamentally different, ideal for precise, quantitative computation but not well-suited, as living beings are, for qualitative reasoning and inference. Mechanisms exist which simulate learning and experience, but must be painstakingly parameterized and are highly complex, typically applied to contrived situations. Such systems lack the sophistication and processing power required for practical solution of any but the most simple vision problems.

In spite of this seemingly bleak outlook, machines do have certain advantages and are better suited to many tasks than are biological vision systems. For example, humans recognize three-dimensional objects and navigate through unknown environments by forming qualitative descriptions of spatial relationships and of their surroundings; however, they cannot reproduce accurate measurements of the objects or environmental maps without the aid of external devices. Computers have at least a hope of autonomously performing such tasks, which as a consequence are active subjects of research and have led to new mathematical theory and methods very different from

those presumably employed by natural vision systems.

1.2 Image Formation

Images, the primary inputs to vision systems, are formed by the projection of light rays from the three-dimensional world (or *scene*) through optical elements (*lenses*) and onto a two-dimensional surface. The aggregate light is captured either by photo-reactive chemical film elements or by discrete electronic imaging elements such as charge-coupled devices (*CCDs*) that react to a small range of frequencies in the electromagnetic spectrum. Color images are typically captured by several sets of sensors, which are analogous to the sets of cones in the retina, each sensitive to a different frequency band—for example red, green, and blue for long, medium, and short wavelengths in the visible range.

1.2.1 Pinhole Lens

A *camera* is any abstract imaging device that consists of optical elements and an image surface. Many different projections of a given scene can be obtained by varying the optics of the camera. Perhaps the simplest projection model is the so-called *pinhole* model (Figure 1-1 and Figure 1-2), in which all light rays striking the film surface pass through a single point known as the *focal point* or *optical center*. An ideal pinhole lens has infinite depth of field, so that scene objects at arbitrary distances from the lens remain in focus; the model provides a good approximation to typical lenses.

Pinhole projection involves simple geometric relationships and governing equations; it does not require complicated ray tracing, as do nonlinear optical elements. Due to its simplicity, and the fact that it approximates many real lenses, this projection model is widely used in vision and graphics applications and will also be used throughout this thesis. Significant nonlinearities and other deviations from the ideal model are assumed to have been corrected prior to processing.

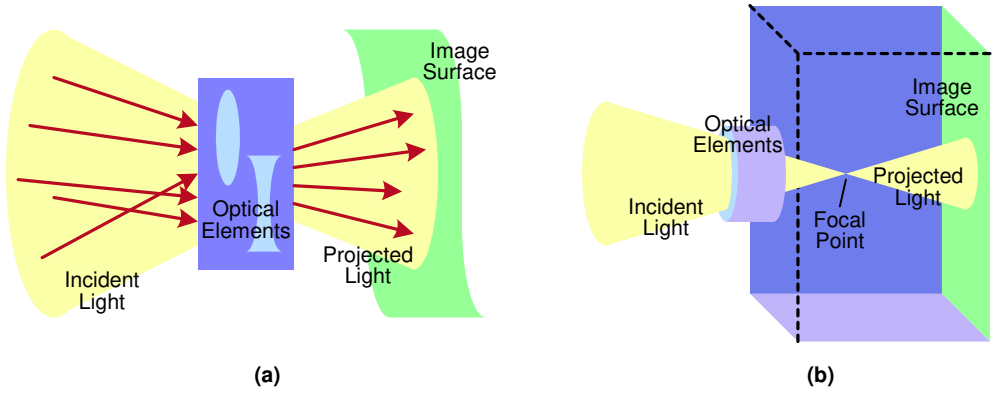


Figure 1-1: Projection Models
(a) A generic projection model transfers light to the image surface through arbitrary optics.
(b) In the pinhole model, all incident light passes through a single focal point.

1.2.2 Image Surfaces

Although images can be projected onto arbitrary surface geometry, by far the most widely used imaging surface is a simple two-dimensional plane: conventional cameras use the surface of a photosensitive, rectangular film plane, while digital cameras use a dense planar array of charge coupled devices. Images on such surfaces resemble perceived projections on the human retina, and thus qualitatively seem a natural representation of the world. Planar projection also happens to be mathematically convenient because it preserves properties such as collinearity and coplanarity: the dimension and form of linear subspaces (i.e. points, lines, and planes) remain invariant under such a projection [MZ92].

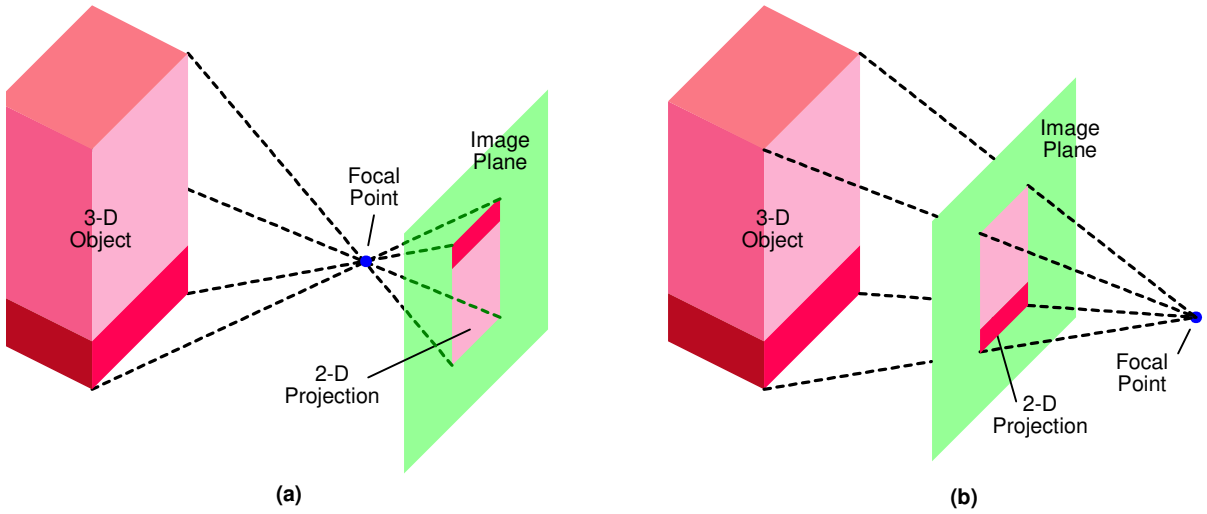


Figure 1-2: Planar Pinhole Projection
(a) Closer to physical reality in most cameras, the inverting model produces a reversed image. (b) The non-inverting model is preferred for its simplicity.

By its nature, a planar surface can only view light rays from a bounded volume, known as the *field of view* (*FOV*), defined by the extent and placement of the surface with respect to the focal point. By contrast, a *plenoptic surface* [MB95, GGSC96] fully enclosing the focal point can capture light rays from all directions. In nature, a very wide FOV serves to provide quick assessment of surroundings and aids in navigation (e.g. flight); full FOV has similar advantages in artificial vision applications because it eliminates certain motion ambiguities [FA98] and provides unbiased treatment of the projection space. Physical realization of such *omnidirectional* imaging devices is difficult, but these devices can be simulated, for example by a catadioptric camera [BN98], an array of outward-pointing conventional cameras [FAB+00], or a single conventional camera on a pan-tilt head [CMT98].

Arguably the most elegant and convenient theoretical representation of an omnidirectional surface is the unit sphere centered at the focal point, which exhibits perfect symmetry and uniform angular ray coverage. In practice, however, the sphere is not the simplest representation; parameterizations of its surface (for example by angular quantities such as elevation and azimuth) are complicated and can contain singularities.

The unit cube centered at the focal point provides an attractive and practical compromise. It is

convex and closed, thus capturing rays from all directions, yet also consists of planar surfaces, thus preserving the desirable properties of planar projection. Future sections will highlight this tradeoff between theoretical and practical representations of the projection surface.

Regardless of the specific form of the image surface, the projection of any scene under the pinhole model is obtained by intersection of the surface with all light rays passing through the focal point. The resulting image depends only on these rays, and not on the original 3-D structure. This is a *projective* dependence, meaning that all unobstructed 3-D points along a given ray produce identical image projections. More formally, if the focal point coincides with the origin, then for a given 3-D ray direction $\boldsymbol{r}$, all points $\boldsymbol{p}$ satisfying $\boldsymbol{p} = \alpha\boldsymbol{r}$ for any real, non-zero scalar α form an equivalence class, and are projectively indistinguishable. This topic is discussed further in §1.3.2.

1.2.3 Camera Parameters

Camera models can be arbitrarily complex, depending on the physical lens configuration and on how closely one wishes to approximate the true optical process. Although the pinhole model is relatively simple, it still has a variety of application-dependent parameterizations (e.g. [FvDFH90, BB95]). The parameters of any pinhole model can be divided into two sets: *internal* or *intrinsic* parameters describe lens characteristics and other properties that depend only on the camera itself, while *external* or *extrinsic* parameters describe the *pose* of the camera—that is, its position and orientation—with respect to some fixed global coordinate system. Since the primary concern of this work is recovery of external camera pose, only a minimal description of the internal parameters will be presented.

Figure 1-3 depicts the three primary coordinate systems used in pinhole projection. The first is the *world* or *scene* coordinate system, a 3-D Cartesian system with respect to which extrinsic camera pose and scene quantities are specified. The second is the camera coordinate system, whose origin is coincident with the camera's focal point. In planar projection, this system's $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{y}}$ axes are parallel to those of the image plane, while in spherical projection, the axes depend on the sphere parameterization used. The last is the image coordinate system, which describes the locations of points after they are projected.

A rigid transformation, consisting of a 3×1 translation $\boldsymbol{t}$ and orthonormal rotation $\boldsymbol{R}$, expresses points $\boldsymbol{p}^w$ in world space as points $\boldsymbol{p}^c$ in camera space. Its inverse specifies the orientation and position of the camera with respect to the scene coordinate system. Formally,

$$\boldsymbol{p}^c = \boldsymbol{R}^\top(\boldsymbol{p}^w - \boldsymbol{t}) \tag{1-1}$$

$$\boldsymbol{p}^w = \boldsymbol{R}\boldsymbol{p}^c + \boldsymbol{t} \tag{1-2}$$

where $\boldsymbol{t}$ is the position of the focal point, and the columns of $\boldsymbol{R}$ are the principal axes of the camera coordinate system, both expressed in scene coordinates. These two quantities thus summarize the external pose of the camera. It should be noted that there are several alternate representations for the rotation matrix $\boldsymbol{R}$, such as a Gibbs vector, angle and axis, or Euler angles; the primary representation used in this work is the unit quaternion, described in Appendix A.

The first-order internal calibration parameters (focal length, principal point, skew, etc) are determined by the extent and displacement of the image surface with respect to the optical center expressed in camera coordinates, and can be summarized by a 3×3 matrix $\boldsymbol{H}$. This matrix trans-

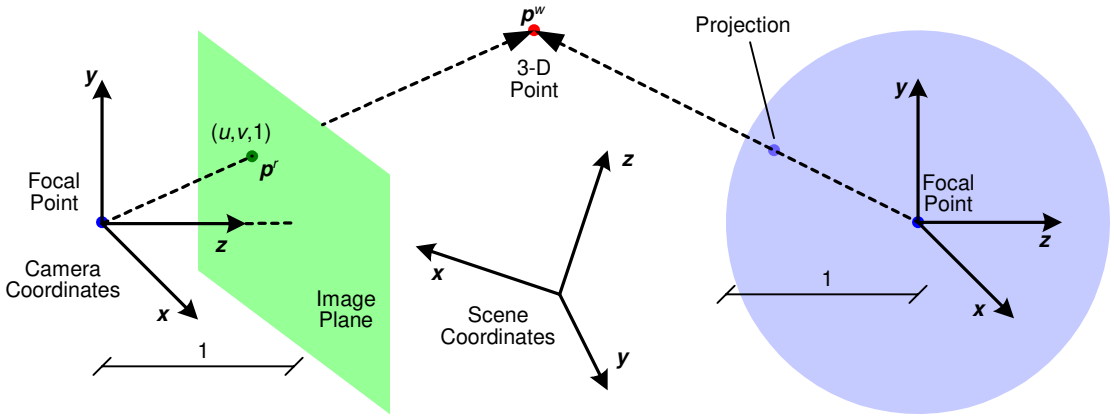


Figure 1-3: Imaging Coordinate Systems
The camera coordinate system has its origin at the focal point and its axes aligned with the image; the origin and axes are expressed relative to the scene's coordinate system. Planar projection of a 3-D point is shown at left, and spherical projection is shown at right.

forms points $\boldsymbol{p}^c$ in camera space to projective rays $\boldsymbol{p}^r$ by

$$\boldsymbol{p}^r = \boldsymbol{H}\boldsymbol{p}^c, \tag{1-3}$$

and the rays can then be intersected with the imaging surface to form the final image.

1.2.4 Projection and Homographies

In planar projection, the rays $\boldsymbol{p}^r$ intersect the image at unit distance from the focal point along the $-\hat{\boldsymbol{z}}$ direction, while in spherical projection the rays intersect the unit sphere centered at the focal point. The resulting points (for planar and spherical projections, respectively) are given by

$$\boldsymbol{p}^p = -\frac{\boldsymbol{p}^r}{\boldsymbol{p}^r \cdot \hat{\boldsymbol{z}}} = (u, v, 1) \tag{1-4}$$

$$\boldsymbol{p}^s = \frac{\boldsymbol{p}^r}{\|\boldsymbol{p}^r\|}. \tag{1-5}$$

Since all light incident on an image in the pinhole model passes through the focal point, the image depends only projectively on the scene; the incident light can thus be summarized by the set of all rays passing through the focal point, where each ray is essentially a 2-D entity. If the image surface geometry is known, the ray direction at any point on the image can be recovered. Therefore, as long as the focal point remains fixed with respect to the scene, images can be reprojected onto other surfaces, simulating other projection types; for example, a portion of a spherical image can be reprojected onto a plane, or a planar surface can be scaled and rotated (Figure 1-4).

Projective transformations representing invertible mappings between image surfaces are known as *homographies*. Planar homographies, which project rays onto planar surfaces, consist of transformation by a nonsingular 3×3 matrix $\boldsymbol{H}$ followed by normalization to unit depth, and form the basis for panoramic stitching and image mosaicing applications [SS97, CZ98, CMT98]. The

(a)

(b)

Figure 1-4: Ray Reprojection
Rays can be remapped in various ways. (a) An example of a linear plane-to-plane homography. (b) A sphere-to-plane homography obtained by unwrapping a hemispherical image.

projection of camera points to image points by (1-3) and (1-4) represents a linear homography of this kind.

A final note, although somewhat beyond the scope of this work, is that real lenses induce various higher-order distortions, some of which can be modeled as nonlinear transformations on the projected rays [BB95]. Distortions can be severe, especially in planar projection cameras whose lenses cover a large field of view. In a carefully calibrated system, lens aberrations of this type can be measured and image distortions corrected; however, correction is seldom perfect, and miscalibration contributes to estimation error in subsequent vision tasks.

1.2.5 Sampling the Image

Since projection surfaces are inherently two-dimensional, they can be parameterized by two variables (for example, (u, v) in (1-4)). The *radiance*, which encodes light color and intensity at a point (u, v) on the surface, is given by the continuous vector-valued function $\boldsymbol{C}(u, v) \in \mathbb{R}^3$. In real applications, images must be processed digitally as spatially-sampled representations of $\boldsymbol{C}(u, v)$. Each sample, or *pixel*, represents the aggregate light incident on a small image region; the radiance of rays in this region are averaged or "blurred", for example by

$$\hat{\boldsymbol{C}}(u_m, v_n) = \int_{\mathcal{R}} w(u, v)\boldsymbol{C}(u, v)\mathrm{d}u\mathrm{d}v \tag{1-6}$$

over a small region $\mathcal{R}$ around the sample (u_m, v_n), where $w(u, v)$ is a spatial weighting function (sometimes referred to as a *point spread function*). CCD arrays in digital cameras and digital scanners both produce images composed of color values on a regularly-spaced rectangular grid of pixels. Color components are also quantized to integer values when stored, but floating-point values are commonly used in image processing to minimize precision loss.

1.3 Image Features

When images are formed, the 3-D structure of the scene is lost; all that remains is a sampled set of projected rays. Taken alone, pixels simply represent a collection of spatially-indexed radiance values having only 2-D structure and no physical interpretation. If the original scene geometry is to be recovered from one or more images, certain higher-level quantities must be extracted that "summarize" the image pixels in a principled and meaningful way. Although some vision applications use the pixel values directly, structure is often more easily identified by using derived *features*, or *tokens*, such as contours, lines, and points; analogies can be drawn to mid-level brain function in biological vision.

Various types of features can be derived directly or indirectly from $\boldsymbol{C}(u, v)$, forming a somewhat hierarchical structure (Figure 1-5). Raw radiance at a single point is the most fundamental image-based quantity, from which other quantities such as *luminance* (monochromatic intensity) can be obtained. In color images, luminance is calculated by a projection of the radiance from $\mathbb{R}^3$ onto $\mathbb{R}$, for example by the linear mapping $L(u, v) = \boldsymbol{C}(u, v) \cdot \boldsymbol{m}$, where $\boldsymbol{m}$ is a constant vector.

The gradient of the image luminance, $\nabla L(u, v)$, forms the basis for extraction of many types of discrete image tokens. Its magnitude can be compared to a threshold value t and used to form a binary-valued "edge" function, such as

$$E(u, v) = \begin{cases} 0, & \|\nabla L(u, v)\| < t \\ 1, & \|\nabla L(u, v)\| \geq t \end{cases}, \tag{1-7}$$

from which line segments and corners can be inferred [Can86, Lim90].

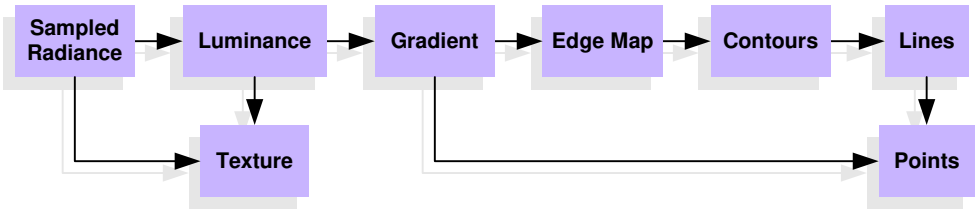


Figure 1-5: Hierarchy of Image Features
Image features can be directly or indirectly derived from raw image color. Certain features can only be obtained via sequences of various other feature extractions, and are thus subject to accumulation of error.

1.3.1 Image Space Tokens

Gradient-based features such as points and contours are often preferred over lower-level features such as texture and luminance in robust vision applications. Regions of large gradient magni-

tude arise from high-contrast texture and, more importantly, from 3-D depth discontinuities, thus capturing structural attributes of the underlying scene. In addition, they can be detected reliably under large variations in illumination, camera position, and other viewing conditions, because they represent *differences*—that is, relative changes rather than absolute values.

Points and straight lines are typically detected in the 2-D Euclidean image plane, or on a planar remapping of a curved imaging surface. This process is convenient for several reasons. First, planar projection preserves the dimension of linear subspaces; thus, points and lines detected in the image directly represent their 3-D counterparts and are likely to correspond to scene structure. Linear quantities are also amenable to a wide variety of powerful mathematical and statistical tools. Finally, the plane is easily parameterized (most often as a rectangular grid), facilitating pixel adjacency computation for such tasks as gradient evaluation and contour chaining.

Since a real image consists only of radiance samples, the gradient must be calculated using a discrete approximation, such as

$$\nabla L(u_m, v_n) \approx \frac{1}{2} \begin{pmatrix} L(u_{m+1}, v_n) - L(u_{m-1}, v_n) \\ L(u_m, v_{n+1}) - L(u_m, v_{n-1}) \end{pmatrix} \tag{1-8}$$

or application of a Sobel filter [Lim90]. Higher-level tokens such as lines are extracted by thresholding the gradient magnitude at each pixel to form a binary edge map as in (1-7), and then grouping collinear edge pixels into sets. The gradient values of a given set of collinear pixels are used to form a best-fit line segment [BHR86]. Point features can be extracted in various ways, such as by corner detection in the gradient image [HS88] or, as preferred in this work, by the apparent intersection of line segments (Figure 1-6). Examples of features in real images are shown in Figure 1-7.

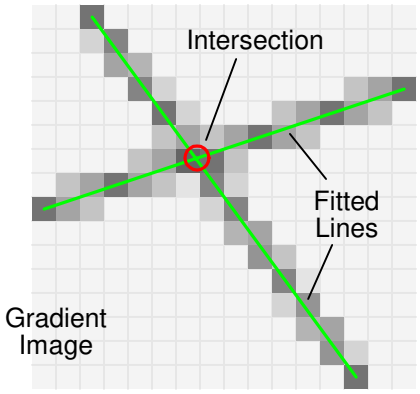


Figure 1-6: Euclidean Lines and Points
Line features are extracted from the image gradient by fitting to connected edge pixels. Point features are formed from the intersection of two or more lines.

A point in the Euclidean plane is most simply represented by its position (u, v). Lines, on the other hand, can be parameterized in various ways. A finite segment is often described by its two endpoint positions (u_1, v_1) and (u_2, v_2), or by its center position (u_c, v_c), length l, and angle θ from horizontal. Infinite lines can be described by relations such as

$$u \sin\theta + v \cos\theta + c = 0, \tag{1-9}$$

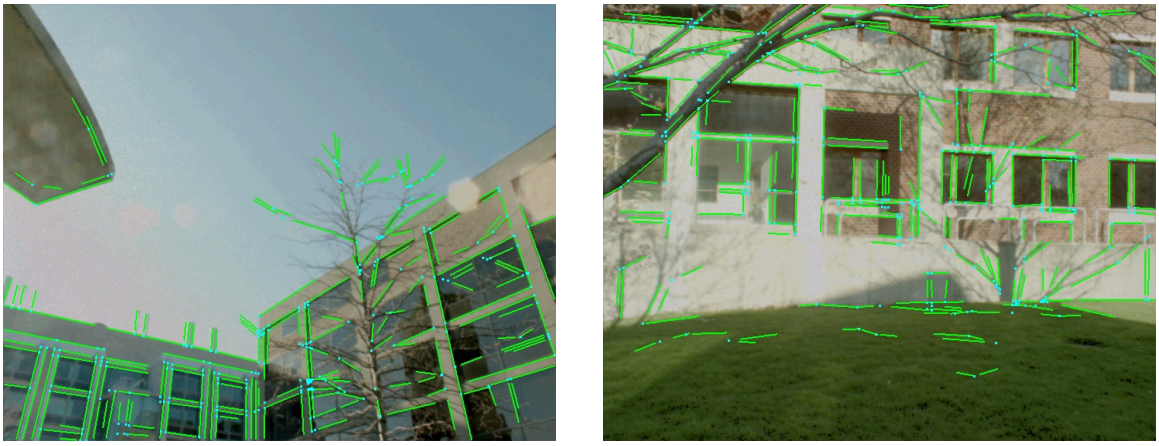


Figure 1-7: Examples of Image Features
Two sample images with line and point features superimposed. Features correspond both to geometry of interest (e.g. buildings) and to undesired "outliers" (e.g. trees).

where θ is the angle of the line to horizontal and c is the offset of the line from the origin, or by

$$au + bv + c = 0 \tag{1-10}$$

$$\boldsymbol{p} \cdot \boldsymbol{l} = 0 \tag{1-11}$$

where $\boldsymbol{p} = (u, v, 1)^\top$ and $\boldsymbol{l} = (a, b, c)^\top$.

1.3.2 Projective Tokens

Although the Euclidean image plane is a convenient space for the detection of low-level tokens, it can be poorly-suited for their representation. Light rays through the focal point are non-uniformly sampled over the projective space, and points at infinity—i.e. rays parallel or nearly parallel to the image plane—lead to instability and poor conditioning in inference tasks.

The *projective plane*, denoted by $\mathbb{P}^2$, is a closed topological manifold containing the set of all 3-D lines through the focal point. Points along any given 3-D line, except the focal point itself, define an equivalence class $\sim$:

$$\boldsymbol{p} \sim \boldsymbol{r} \quad \Leftrightarrow \quad \boldsymbol{p} = \alpha \boldsymbol{r}, \quad \alpha \neq 0 \tag{1-12}$$

where α is a real scalar value. Because of the relationship in (1-12), the projective plane is a *quotient space* on $\mathbb{R}^3$ (minus the focal point) and also on the surface of the unit sphere $\mathbb{S}^2$, sometimes referred to in the literature as the *Gaussian sphere* [Bar83]. The sphere's surface is an ideal space for representation of projective features, just as it is an ideal space for image projection: it is closed, compact, and symmetric, and it provides uniform treatment of rays from all directions.

It should be noted that as theoretical quantities, projective points on the sphere are *axial*, meaning that they exhibit antipodal symmetry, while real rays under the pinhole model are *directed*, meaning that antipodal points in a spherical image correspond to different physical light rays [Sto91]. However, antipodal points are often geometrically indistinguishable; therefore, this thesis treats projective features as axial quantities described on the unit hemisphere.

Points in the Euclidean image plane can be transformed to points on the sphere via a simple homography as presented in §1.2.4. A given point undergoes the transformation

$$(u, v) \quad \rightarrow \quad \boldsymbol{p} = (u, v, 1)^\top \quad \rightarrow \quad \frac{\boldsymbol{p}}{\|\boldsymbol{p}\|}; \tag{1-13}$$

it is first augmented to homogeneous (projective) coordinates, and then normalized to unit length. By contrast, points on a spherically-projected image need no transformation, thus emphasizing the utility of the unit sphere representation of the image surface.

All collinear points $\boldsymbol{p}_i$ in the image must satisfy the orthogonality constraint of (1-11), which implies two important facts. First, the line parameters $\boldsymbol{l}$ are unique only up to an arbitrary non-zero scale; $\boldsymbol{l}$ itself can thus be viewed as a projective point. Second, $\boldsymbol{p}$ and $\boldsymbol{l}$ are symmetrically related: the roles of the two quantities can be interchanged without altering the constraint. These two facts imply a simple *projective duality* between points and lines, which states that a line $\boldsymbol{l}$ can be represented as a unit direction $\frac{\boldsymbol{l}}{\|\boldsymbol{l}\|}$ on the sphere to which any projective point lying on the line is orthogonal; the set of all such points traces a great circle on the sphere.

Similarly, a given image point $\boldsymbol{p}$ can be viewed as a *pencil* of image lines which contain, and thus intersect at, that point. The parameterizations $\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots$ of such lines must satisfy (1-11), and thus the set of all line duals through $\boldsymbol{p}$ trace a great circle on the sphere orthogonal to $\boldsymbol{p}$. These relationships are depicted in Figure 1-8 and will be further emphasized in the discussion of projective inference and data fusion for the formation of line features and vanishing points.

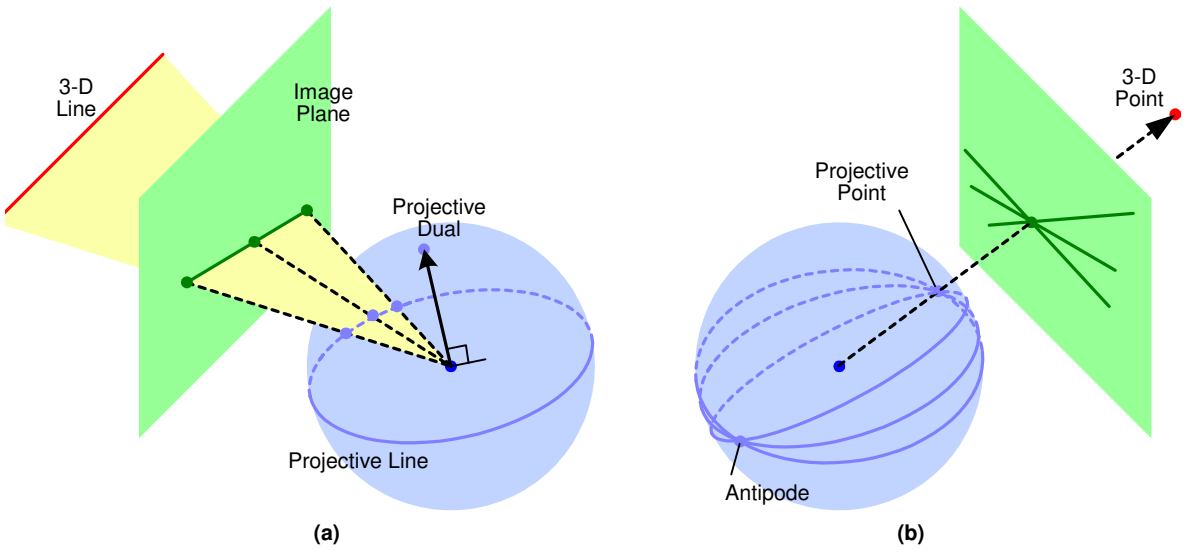


Figure 1-8: Projective Image Features
(a) A 3-D line can be represented by a 2-D line in planar projection or a great circle in spherical projection. Any point on the line must be orthogonal to the line's dual representation. (b) A 3-D point can be represented as a unit vector on the sphere, or as a pencil of lines passing through its projection.

1.4 Multi-Camera 3-D Vision

With suitable parameterizations for cameras and features in place, it is possible to pose a typical 3-D reconstruction problem: given a set of images and a set of extracted features, the problem is to infer the original 3-D scene structure. This is effectively the inverse of image projection and rendering, which deal with the formation of 2-D images from a 3-D scene. It is assumed here that the scene is rigid and static, i.e. that its shape and position are constant relative to a fixed global coordinate frame. The imaging configuration is modeled either as a single camera that moves with respect to the scene, or as a set of cameras at different positions and orientation that view the scene simultaneously.

1.4.1 Multi-View Geometry

There are three tightly coupled sets of parameters in 3-D vision. The first of these is the scene geometry, or the structure of the world expressed with respect to some fixed reference frame. Second is the camera geometry, or the set of internal and external camera parameters described in §1.2.3. The last is *correspondence*, which is an association between common geometric elements among different images, or between scene points and their projections. Figure 1-9 depicts the relationships between scene structure, camera geometry, and point feature correspondence.

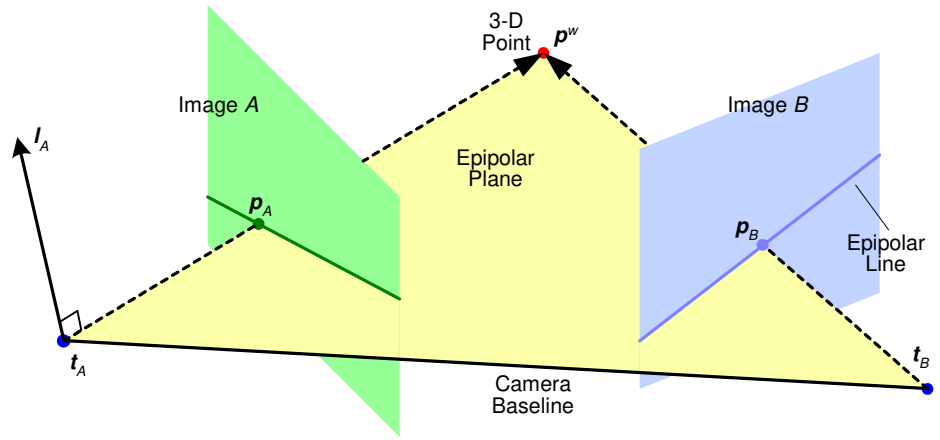


Figure 1-9: Structure, Motion, and Correspondence
A typical two-camera configuration depicting the interrelation between structure, motion, and correspondence. Projections of a scene-relative 3-D point must lie on the epipolar plane formed by the two camera centers and the point itself. Intersection of this plane with the images produces epipolar lines.

The plane that passes through both cameras' focal points and through a 3-D feature is called an *epipolar plane* and contains both 2-D observations of that feature. Various properties of epipolar geometry are often exploited to constrain structure and motion. For example, if the poses of cameras $\mathcal{A}$ and $\mathcal{B}$ are expressed by rotations and translations as described in §1.2.3, then for a given point feature observed by camera $\mathcal{B}$, the corresponding feature in image $\mathcal{A}$ must lie on a projective

epipolar line defined by

$$\boldsymbol{l}_{\mathcal{A}} = (\boldsymbol{R}_{\mathcal{A}}^{\top}\boldsymbol{R}_{\mathcal{B}}\boldsymbol{p}_{\mathcal{B}}^{r}) \times \boldsymbol{R}_{\mathcal{A}}^{\top}(\boldsymbol{t}_{\mathcal{B}} - \boldsymbol{t}_{\mathcal{A}}), \tag{1-14}$$

which is the cross product between the baseline (the vector connecting the optical centers of cameras $\mathcal{A}$ and $\mathcal{B}$) and the point feature, all expressed in the scene coordinate frame. The line $\boldsymbol{l}_{\mathcal{A}}$ is orthogonal to the epipolar plane, and represents the projection of the feature ray $\boldsymbol{p}_{\mathcal{B}}^{r}$ onto camera $\mathcal{A}$'s image surface.

1.4.2 Structure and Motion

Knowledge of any two of the three parameter sets (scene structure, camera pose, and correspondence) uniquely specifies the third. For example, in computer graphics, images are formed from precisely-specified scene geometry and camera parameters [FvDFH90], and in 3-D motion tracking, depth is recovered from known camera geometry and correspondence by direct triangulation [NRK98].

Knowledge or assumptions about a single parameter set is often sufficient for determination of the other two, though more work is required. In stereo vision [Fau93], known camera geometry is used to estimate both correspondence and depth. *Structure from motion* techniques, which attempt to recover scene structure and camera motion, rely on known correspondence. In the most general case, however, *none* of the three are known, and due to their strong coupling all parameters must be estimated simultaneously. This problem is severely underconstrained without additional knowledge, such as restrictions on the baseline, approximate camera motion, calibrated camera intrinsics, or a few correspondences or 3-D points.

Without prior 3-D scene information, it is impossible to recover the location and attitude of cameras with respect to global scene coordinates because the geometry is specified relative to an arbitrary frame of reference. Most often one of the cameras' local coordinate systems (e.g. that of camera $\mathcal{A}$) is therefore treated as the reference frame, and the other cameras are registered relative to this frame. The epipolar line equation in this case simplifies to

$$\begin{aligned} \boldsymbol{l}_{\mathcal{A}} &= (\boldsymbol{R}_{\mathcal{B}}\boldsymbol{p}_{\mathcal{B}}^{r}) \times \boldsymbol{t}_{\mathcal{B}} \\ 0 &= \boldsymbol{l}_{\mathcal{A}} - (\boldsymbol{R}_{\mathcal{B}}\boldsymbol{p}_{\mathcal{B}}^{r}) \times \boldsymbol{t}_{\mathcal{B}}, \end{aligned} \tag{1-15}$$

and (1-15) can be used as an error measure for estimates of the pose and structure. Epipolar geometry can also be summarized by a single matrix $\boldsymbol{F}$ known as the *fundamental matrix* [LF97], used in projective reconstruction techniques (§2.1.5), for which $\boldsymbol{p}_{\mathcal{A}}^{r}\boldsymbol{F}\boldsymbol{p}_{\mathcal{B}}^{r} = 0$. Another common error measure, defined in the Euclidean image plane, is the geometric distance between features extracted from the images and the projections of their corresponding 3-D points.

1.4.3 Correspondence

Many techniques have been devised to establish correspondence among image features arising from the same 3-D geometry. Tokens can be automatically tracked over time in image sequences or manually correlated, but the task of feature correspondence is generally coupled with recovery of structure and motion; establishing correspondence can thus be a circular problem.

All 3-D vision tasks implicitly require that cameras view overlapping geometry; otherwise images are completely uncorrelated and there is no correspondence to establish. The degree of overlap affects the efficiency and accuracy of any reconstruction algorithm. One difficulty that arises in feature correspondence methods is that not all 3-D scene geometry is viewed by all cameras. Thus, a given feature extracted from one image often has matches in only a few other images, or may have no matches at all. Missing features are caused by factors such as limited FOV, failure of 2-D feature extraction (for example due to quantization of distant objects), and obstruction (or *occlusion*) of geometry.

1.5 Thesis Overview

Having presented a brief outline of issues arising in classical 3-D vision, this chapter shifts focus to the current work, which, while building upon results from many areas of vision and mathematics, differs from past efforts in many respects. The enveloping ideas can be summarized by the following thesis statement:

> *Stochastic geometric, feature-based approaches combining parameter discretization and Bayesian inference techniques can be used to recover extrinsic camera pose among a large set of images. Such techniques provide a powerful probabilistic framework within which robust, accurate, and automatic estimates of camera orientation and position, as well as precise descriptions of their uncertainty, are obtainable without explicit feature correspondence.*

Several moderate assumptions are imposed by the underlying geometry and the application context. It will be shown, however, that a few of these assumptions are overly strict and can be relaxed, generalizing the techniques and allowing them to be used in a variety of situations. The remainder of this chapter outlines the thesis in more detail, discussing the specific goals of this work as well as providing an overview of the methods developed herein.

1.5.1 Objectives

The primary goal of this thesis is to develop and verify methods for extrinsic camera pose recovery that meet the criteria in the thesis statement without imposing overly restrictive assumptions. Modeling and estimation of uncertainty in all quantities using stochastic geometry and inference techniques is also an important objective.

It is crucial that all methods developed in this work rely on a solid theoretical foundation from which accurate, effective models for geometric uncertainty and parameter estimation can be derived. At the same time, the enveloping context and original motivation (described in §2.2) is a project whose goal is vision-based acquisition of urban scene models. The end-to-end concerns of the project motivate practical, efficient implementation of all algorithms and mathematical theory as well as analysis of their performance both under controlled conditions and on real-world data. This thesis thus attempts to strike a balance between pure theory and practical application.

1.5.2 Assumptions and Restrictions

Completely general systems for 3-D machine vision, as discussed earlier, have thus far eluded researchers' efforts; therefore, as with all other artificial vision systems to date, certain restrictions must be made on the problem so as to make pose recovery more tractable. These restrictions come in part from the goals of the larger project, and can be summarized as follows:

- ***Calibrated intrinsic parameters***. Since this work concerns only the recovery of external pose, it is assumed that the internal parameters are accurately known, and that a mapping from distorted image pixels to metric 3-D rays in camera coordinates is available. The parameters can be recovered by a variety of standard calibration techniques (for example [Tsa87]).

- ***Roughly-known extrinsic pose***. Approximate estimates of all camera positions and orientations must be available; these aid in disambiguating pose configurations, and also allow cameras to be efficiently partitioned into sets likely to view overlapping geometry.

- ***Hemispherical images***. Input images are assumed to be omnidirectional. Although the methods developed in this thesis can also be applied to planar images with more limited FOV, wide-angle imagery has many desirable properties, providing redundancy, disambiguating similar motions, and reducing bias in inference problems.

- ***Parallel lines***. Rotational pose recovery requires that at least two sets of parallel scene lines are viewed by each camera. Man-made environments often contain regular geometric structures that exhibit attributes such as coplanarity, collinearity, and parallelism, which more than satisfy this requirement.

- ***Sufficient image density***. As mentioned in §1.4.3, all 3-D vision systems implicitly assume that cameras view overlapping geometry. Since no scene structure is known or assumed *a priori*, a given camera's pose must be registered to geometry as viewed by other cameras rather than to ground-truth measurements. Cameras must therefore be placed so as to view common structure. More precisely, a significant portion of the geometry seen by any given camera must also be seen by at least one other camera.

1.5.3 Outline of Methods

This section presents an overview of the camera pose recovery system, illustrated in Figure 1-10.

The primary system inputs are extracted image features, accurately calibrated internal camera parameters, and approximate camera pose expressed in metric scene coordinates. The system consists of three main stages: recovery of rotational pose, recovery of translational pose, and global registration. First, the uncertainty of line features detected in the 2-D images is estimated using projective inference on the image gradient along the lines. Next, lines in each image are classified and assembled to construct position-invariant features (vanishing points) by exploiting projective duality and similar inference techniques. An expectation maximization (EM) algorithm based on a projective mixture model and initialized by a Hough transform allows for simultaneous estimation of multiple vanishing points in each image. Another EM formulation probabilistically

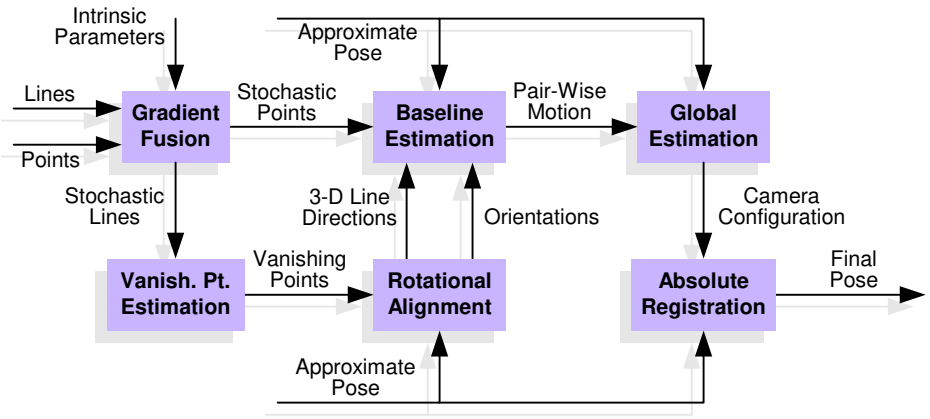


Figure 1-10: Pose Recovery System Overview
Images, extracted line and point features, and approximately calibrated cameras serve as system inputs. The system produces accurate external pose estimates as well as uncertainty in these estimates.

correlates vanishing points between images and recovers global orientation and its uncertainty for each image, *independently* of position.

Using the recovered orientation and extracted point features, whose uncertainty is derived from that of their constituent lines, translation directions are estimated between all adjacent pairs of cameras. Known orientations constrain local epipolar geometry and reduce pair-wise motion estimation to a simple linear form. Recovery of two-camera motion can then be formulated as a projective inference problem identical to those of line estimation and vanishing point detection. Geometric constraints drastically reduce the enormous number of possible point correspondence sets, but explicit correspondence is not established; rather, a Monte-Carlo expectation maximization (MCEM) algorithm, initialized by a unique Hough transform, samples from a probability distribution defined over all plausible correspondence sets.

Projective direction uncertainty is transformed to Euclidean positional uncertainty by employing a transformation between density parameters in the two spaces. The pair-wise direction estimates are then used as constraints in a global optimization that recovers consistent relative camera positions. Finally, all cameras are registered to metric coordinates via a rigid linear transformation.

1.5.4 System Properties

The techniques described above have several notable properties which represent incremental steps toward a general vision system.

- ***Robustness and accuracy***. Combinations of discrete and continuous parameter estimation techniques, as well as explicit compensation for outlier processes, provide a high degree of robustness. The registration system is fully automated, and converges on globally optimal solutions accurate to roughly 0.1° (2 milliradians) and 5 centimeters in data sets spanning hundreds of meters, even with significant error in the initial pose.

- ***Stochastic geometry***. Every geometric entity is treated as a sample from a probability distribution defined on the appropriate parameter space. Explicit models of geometric uncertainty

in all quantities are propagated through various novel inference and data fusion techniques, thus incorporating all possible error information and producing accurate measures of error in the final pose.

- ***Data fusion***. Redundancy in the data is exploited to produce more accurate parameter estimates. Fusing thousands of features into a few summarizing entities not only improves the accuracy of the resulting parameters, but also reduces the computational complexity and time required for subsequent tasks. Hundreds of thousands of line and point features are fused so that rotational pose can be recovered using only a few vanishing points, and translational pose using only a few pair-wise direction estimates.

- ***View insensitivity***. Features derived from the image gradient (lines and points) are largely insensitive to dissimilar illumination and viewpoint across images.

- ***Decoupled orientations***. Orientations of all cameras are decoupled from their positions; globally optimal rotational estimates are obtained using only adjacency information, and camera position is then determined using these estimates.

- ***Probabilistic correspondence***. Explicit correspondence among vanishing points and among point features is neither required nor produced. Instead, probabilistic correspondence is obtained via stochastic clustering algorithms.

- ***Computational efficiency***. Pose recovery is scalable and can operate on thousands of images. Rotational registration scales linearly in space and time with the number of images, line features per image, and vanishing points per image. Translational registration scales linearly with the number of images and quadratically with the number of point features used per image. Both techniques are also highly parallelizable.

1.5.5 Dissertation Roadmap

The remainder of this dissertation presents the underlying theory, methods, results, and evaluation of the thesis. The next three chapters provide some context and mathematical background. Chapter 2 discusses the motivating application framework of urban scene reconstruction, and provides a brief overview of the extensive body of research in extrinsic camera calibration and stochastic geometry. Relevant work on more specific topics is mentioned where applicable throughout the dissertation. Chapter 3 reviews fundamental probability theory and inference techniques, and presents models of uncertainty for image features and other geometric entities. Finally, Chapter 4 discusses the theory behind a few fundamental algorithms applied to this work, as well as novel extensions that improve their performance.

The two subsequent chapters present geometric constructions, theoretical methods and practical implementations for the recovery of external pose among a large set of cameras, which forms the main body of this work. Chapter 5 illustrates how rotational pose can be decoupled from translational pose and estimated using vanishing points, which are position-independent quantities. The resulting camera orientations, along with their uncertainty, constrain epipolar geometry and facilitate determination of camera positions, presented in Chapter 6.

The final two chapters assess the viability and performance of the uncertainty models and pose recovery methods developed in this work. In Chapter 7, a set of experiments on synthetic and real

data sets is described and results are presented. Chapter 8 offers a more detailed discussion of the results and summarizes the main contributions of this work, as well as suggesting directions for future research.

CHAPTER 2

Context

ALTHOUGH POSE RECOVERY is sometimes the sole objective of vision systems—for example, in certain robot navigation tasks—it seldom stands alone. Most often camera calibration is simply the means to an end, representing only a single component of a higher-level application. This component is a vital one, however, and is sometimes overlooked or neglected in system design.

Regardless of the application, ideal vision algorithms, including camera registration, possess a number of important attributes, such as:

- ***Automation***. Systems should operate autonomously, with little or no reliance on human intervention.
- ***Flexibility***. Operability should be maintained over a wide range of different viewing conditions and geometric configurations; for example, performance should not degrade with changes in illumination or camera baselines.
- ***Scalability***. Processor and memory usage should scale linearly or sub-linearly with the number of images and the number of features per image, and algorithms should be parallelized where possible for greater efficiency.
- ***Accuracy***. Systems should be formulated in a robust probabilistic framework that utilizes all available information and models all geometric uncertainty. This allows them to automatically detect and characterize failure, and to report the accuracy of results when successful.

Attainment of these characteristics is a significant step toward fully generalized 3-D vision. A great deal of effort has been expended on these problems, but despite the enormous body of extant work, no system has been developed that fully meets the above criteria. This thesis work is inspired in part by the lack of general, automatic pose recovery techniques.

The remainder of this chapter provides the research context that further motivates this work and within which new contributions are made. §2.1 briefly reviews previous work and the state of the

art in camera pose recovery, highlighting strengths and weaknesses of each class of techniques in light of the above ideal attributes. A project for reconstruction of large-scale urban environments, the primary framework for this thesis, is discussed in §2.2. Finally, the new techniques and theory developed as part of this work are outlined in §2.3.

2.1 Related Work

This thesis draws on results from many fields of science and mathematics. The contributions of hundreds of researchers both within and outside the field of machine vision serve as the foundation upon which this work is built; the enormity of the existing body of research thus precludes detailed treatment. A high-level overview of pertinent pose recovery techniques is presented in the following sections, and further detail is provided where appropriate throughout the dissertation.

2.1.1 Direct Pose Measurement

A somewhat generic formulation of 3-D reconstruction and its inherent difficulties was presented in §1.4. Some systems bypass the difficulties of traditional registration by using carefully planned camera paths and by directly measuring angles and distances. Such measurements can be obtained manually, though more sophisticated systems attach cameras to electronically-controlled mechanical lever arms and motors, which allow them to be positioned very precisely by programmatical specification of exact locations, viewing angles, and trajectories [IMG00].

This "brute-force" technique is commonly used by film crews for special effects, and obviously requires significant preparation and expensive, specialized equipment. The range of camera motion is restricted, subject to the equipment's physical limitations; large-scale applications that require significant camera separation thus necessitate painstaking manual surveying or *photogrammetry*. Other systems utilize a variety of auxiliary sensors such as global positioning systems (GPS), inertial devices, and range finders rigidly mounted with the camera on a freely-moving platform [BT00, SA00]. Such systems enjoy greater mobility and, with sufficient sensor redundancy, can fuse many separate noisy measurements into reliable pose estimates.

Interestingly, systems which incorporate sophisticated measurement devices often underestimate the importance of camera calibration by overestimating the capabilities of their equipment. Error in these systems is inevitable; for example, odometry and inertial sensors are subject to drift and other perturbations, and GPS, while ideally quite accurate, is only effective in practice when a sufficient number of satellites are in view, and has unpredictable behavior otherwise. Systems equipped with on-board 3-D range scanners, which measure distances to objects by timing the return of reflections (or "time of flight") of emitted light rays [LPC+00], do indeed recover accurate metric distances and scene geometry, but do so only *per viewpoint*; models acquired from multiple viewpoints—such as the various faces of a building or the profiles of a sculpture—must still be aligned with one another, and the pose recovery problem thus persists. Ironically, those systems designed to completely bypass camera calibration must still incorporate offline pose refinement algorithms.

2.1.2 Interactive Systems

As mentioned in §1.4, 3-D vision comprises several tightly-coupled problems (camera calibration, correspondence, and scene structure recovery); knowledge about any parameter set provides useful information about the others. One class of 3-D reconstruction techniques incorporates *a priori* geometric knowledge using semi-automated methods; that is, a human operator provides input through a user interface that strongly constrains the scene geometry and camera pose.

In [BB95] and [SHS98], 2-D features are used to impose linear 3-D constraints. Lines and points are identified and grouped into parallel, orthogonal, and coplanar sets, all via a manual process. The systems are then able to automatically recover full internal and external camera calibration as well as piecewise-planar scene structure, even from a single image. Multi-image reconstruction is also possible via manual correspondence of features across images.

Other systems utilize higher-level constraints, allowing the user to manipulate 3-D object primitives rather than individual 2-D points and lines [DTM96]. The primitives, which reside in scene space, are created by the user and projected onto each image. The user then aligns their corners with the appropriate image features, resulting in parameterized 3-D to 2-D correspondence. The system incorporates implicit constraints on each primitive (for example, orthogonality and parallelism in a rectangular solid) to estimate the free object parameters (e.g. scale, rotation, and translation) as well as the camera parameters.

Interactive systems bypass many difficulties inherent in typical 3-D vision. User interfaces are designed so that the most "difficult" work is done by the operator, whose visual system and perceptual knowledge impose powerful constraints; yet, if designed well, the interfaces are simple enough that constraints can be specified using only a few primitives. Sub-problems can often be decoupled (for example, internal from external camera parameters, or rotational from translational pose) in light of these constraints, greatly facilitating geometric recovery. As a result, models produced by these systems are accurate and visually convincing.

The drawbacks of such techniques, like the advantages, stem from user interaction. Because information is specified by a human operator, it is prone to error; 2-D feature locations are specified manually and can be less accurate than those of automatically detected features. More importantly, user-assisted techniques do not scale well: the task of manual feature identification and correspondence becomes cumbersome and eventually infeasible as the image set grows. These techniques can also lack stability and robustness, discussed in §2.2.5.

2.1.3 Controlled Calibration

Many vision applications, such as surveillance [Ste98] and 3-D motion tracking [NRK98], do not require cameras to move. In traditional *binocular stereo* [Hor86], two synchronized cameras with fixed relative pose are used to recover depth. This formulation can also be extended to three or more cameras [Sar00]. The basic paradigm differs little from that of the more general formulation sketched in §1.4.2: correspondence, usually determined densely over image luminance, is used to recover depth. Since the cameras are situated in a controlled configuration, however, relative pose can be calibrated in advance.

Calibration often involves imaging a target object with known geometry [Tsa87]. Detection of image features allows for 3-D to 2-D correspondence in each calibration image, making the estimation of relative camera pose and intrinsic parameters rather straightforward. Another cali-

bration technique involves the use of spotlights or other small, easily detectable entities that move through the scene [AP96, CDS00]. Measuring the points' positions as seen by all cameras in all images over time is equivalent to obtaining a set of corresponding features, so standard structure from motion techniques (§2.1.4) can be applied.

This type of calibration works well and is easily implemented. Spotlights and other targets are designed to differ significantly enough from the background that they can be automatically and reliably detected in the training images, and cameras can be calibrated very quickly as a result. However, these methods are restrictive in that they require controlled, offline calibration, and thus can only be applied when the relative camera geometry is fixed for the entire image sequence. In addition, the extensive physical hardware often required for multi-camera configurations introduces undesirable objects (e.g. scaffolding, brackets, and other cameras), even in non-training images [NRK98].

Finally, most stereo techniques operate directly on luminance or color, making them sensitive to image noise, changes in lighting, and wide camera baselines. Since detailed, accurate 3-D reconstruction is usually not the goal of these techniques, most authors do not address sensitivity issues or pose uncertainty.

2.1.4 Structure from Motion

Techniques more general than stereo, which attempt to recover geometry, camera pose, and correspondence for arbitrary image configurations, are collectively referred to as *structure from motion* (*SFM*) techniques. SFM spans a broad spectrum of theory and implementation [LMD95, PZ98, SK94, TML99]; this section highlights a few relevant examples and illustrates general similarities and differences between them.

SFM techniques involve correlation between features, whether dense (i.e. contiguous regions of image luminance or texture) or sparse (i.e. discrete contours and points). Dense region correlation is inherently sensitive to changes in illumination, although normalization techniques alleviate this problem somewhat [Coo98, LTM99]. Occlusion and drastic viewpoint changes also hamper these techniques, however, so sparse gradient-based feature correlation is preferred, since it is less sensitive to such phenomena (§1.3.1). In any case, here correspondence is assumed known; this issue is set aside for the moment and treated in more detail in §2.1.6.

Direct nonlinear constraint equations have been derived, for example by Horn [Hor86], to estimate structure and motion given a set of correspondences. Although the coupled nature of these equations often precludes analytic solution, certain restricted problems can be solved in closed form [Hor87, LH81, Har97]. Such problems are often described in terms of the minimum number of data points or constraints necessary to obtain a unique solution, which is perfectly accurate in the absence of noise. Noise is inevitable in real data, however, so parameters are typically overconstrained by incorporation of more data than is strictly required. This allows the parameters to be estimated, rather than obtained analytically, by minimization of an error function or by maximization of a probabilistic likelihood function.

Nonlinear optimization is performed using a variety of techniques. Some methods decouple parameters into more manageable subsets, sequentially optimizing over different subsets of parameters while assuming the rest are fixed at their most recently estimated values. *Bundle adjustment* is a common example that alternates between estimation of scene geometry and camera pose. Error functions are minimized by Newton-Raphson, Levenberg-Marquardt, and other gradient descent

algorithms that iteratively update the parameters by linearizing the error function around their current values [PTVF92]. Depending on the degree of nonlinearity in the problem, such algorithms can converge to local optima in the error landscape, and thus require sufficiently accurate initial parameter estimates to guarantee convergence to the correct global optimum.

Often, pose is recovered only between consecutive image pairs or triples in a sequence. This alleviates the problem of scale for long image sequences, but the weakness of purely local techniques is that they are prone to bias and error accumulation as the sequence progresses. Azarbayejani [AP95] addresses this issue by using an extended Kalman filter to update structure and motion using all available data, incrementally improving the estimates as new data is introduced.

Certain linearized versions of SFM problems have been formulated. One example is projective reconstrution, described in §2.1.5. Another is factorization, an elegant formulation that uses the singular value decomposition (SVD) of an observation matrix to separate structure parameters from motion parameters [PK94]. All features from all images are incorporated into a single, global optimization whose complexity scales linearly with the number of images and features, and which essentially converges in one step. The main drawback of factorization techniques is that they use a linear approximation to pinhole projection and are thus not applicable to scenes with large depth disparities (i.e. significant foreshortening). Nevertheless, factorization is simple and efficient, and is often used to initialize direct nonlinear optimization techniques [KH98].

Novel viewpoints on the SFM problem have recently emerged, most notably by Soatto, who poses the problem in the context of system dynamics and establishes rigorous theorems concerning characteristics such as reachability and observability of system states [Soa97, SB98]. Degeneracies and inherent ambiguity have an elegant interpretation in this context, and the methods seem promising. Thus far, however, they are mainly theoretical and have yet to be validated using real-world data; experimental results have been presented only for somewhat contrived situations.

2.1.5 Projective Reconstruction

Another class of techniques neither requires nor produces internal camera calibration. These techniques operate solely in the projective regime, recovering epipolar geometry from completely uncalibrated image sequences. Feature correspondence provides multi-linear constraints on structure and motion, leading to completely linear equation sets. Because of their generality and simple, elegant formulation, projective techniques have recently become popular alternatives to more traditional SFM approaches [LF97]. Much of the original development in this area is due to [Fau93, MZ92]; more recent theory is presented by [HZ00].

Perhaps the most serious drawback of purely projective techniques is that they recover structure and pose only up to an arbitrary projective transformation, meaning that metric reconstruction is unattainable. Liebowitz describes methods for imposing manually-specified constraints such as parallelism and orthogonality to recover a metric transformation on the recovered geometry [LCZ99], but does not address automatic application of these constraints. It has been argued that techniques which attempt to "fit" true rotations and translations to recovered projective transformations unnecessarily introduce error through extraneous parameters, unlike direct metric recovery of camera pose [Hor00]. Others use projective reconstruction to initialize direct, nonlinear SFM methods [MSKS00], which seems to negate some of the benefits of projective formulations.

2.1.6 Correspondence Methods

Nearly all registration algorithms rely on explicit knowledge of correspondence between features. The most common correspondence techniques involve low-level motion tracking. Sparse tracking methods first extract a set of prominent features (e.g. corners) from a given image, then find the best matches in subsequent images using characteristics such as spatial proximity, color, brightness, and gradient [SKS95]. Dense tracking methods estimate a motion field (*optical flow*) over the entire image using local constraints on the gradient, thus establishing local correspondence for all pixels [HS81, ZMI00].

Motion tracking generally requires small spatial and temporal separation (i.e. short camera baselines) so that the image brightness and viewpoint vary slowly over the sequence. *Pyramid* or *multiresolution* techniques [BA87] relax this restriction somewhat by first downsampling the images to obtain gross motion estimates, then gradually increasing the resolution to attain the desired accuracy. However, perspective distortion due to wide baselines can prevent establishment of reliable correspondence in any such approach.

Another limitation of tracking techniques is that error in feature positions tends to accumulate as the sequence progresses. This spatial drift prevents stable correspondence beyond one or two hundred frames [SKS95]. In addition, features move in and out of view due to occlusion and finite view fields, and most low-level algorithms, which typically operate on adjacent images, cannot determine whether newly detected features were previously seen [CV98].

Robust, accurate matching can thus be quite difficult in the presence of occlusion and outliers. Several robust statistical techniques have been developed to diminish the effects of outliers. Examples include RANSAC (random sampling consensus) [FB81, FZ98], MLESAC [TZ00], ROR (rejection of outliers by rotations) [ARS00], and LMS (least median of squares) [CC91, Ste98] algorithms, in which a large number of small match sets is chosen at random from the pool of all possible matches. Each set contains the minimal number of matches required to uniquely determine epipolar geometry, and thus essentially casts a vote for a particular motion estimate. The most consistent motion is chosen according to number of votes received, and the process repeats. These algorithms are quite robust even when outliers represent up to 40% of the data set, but perform rather poorly when this percentage is exceeded. In addition, explicit correspondence generated by these techniques cannot account for inherent match ambiguities (e.g. the aperture problem [YC92]).

Recently, techniques have emerged which recover pose and scene structure without explicit correspondence. Fua describes a technique that maximizes dense texture correlation across images, hypothesizing a deformable 3-D mesh and refining both its structure and the camera pose by minimizing luminance discrepancy between back projections of the original images onto the mesh [FL94]. The technique produces convincing results and detailed models, but like other texture-based approaches is ill-suited to situations in which the viewing conditions change significantly or geometry is occluded.

Several authors attempt to determine motion using probabilistic rather than explicit correspondence. Dellaert proposes the use of stochastic techniques to determine probabilistic correlation among discrete features by sampling from a distribution over all possible correspondence sets [DSTT00, Del00]. This method is theoretically sound and converges correctly even when absolutely nothing is initially known about parameters. However, it requires that the number of observed 3-D features is known, and that all features are visible in all images (i.e. that there are

no outliers and no occlusion), drastically limiting its applicability. Systems proposed by Rangarajan handle outliers and even non-rigid scene structure using mutual information [RCD99] and EM formulations [CR00a], but because of somewhat ad-hoc implementations, the probabilistic correspondence so obtained does not represent the true distribution.

Despite their shortcomings, such methods are promising because they circumvent many difficulties inherent in more traditional tracking and correlation methods. Probabilistic formulations are also more flexible and better suited to describing ambiguities in correspondence than are explicit assignment methods. These formulations provide the basis for a novel translation recovery algorithm, presented in §6.2.

2.1.7 Stochastic Geometry

It is important in any vision application to incorporate as much relevant information as possible. Uncertainty in parameter estimates should be determined, or at least bounded, and noise attributes of the raw input data should also be characterized and incorporated where possible to improve parameter estimates. Nearly all vision systems utilize some notion of system noise or error; even simple least-squares algorithms that specify no explicit error models assume implicitly that data is corrupted by additive Gaussian noise (this connection between probabilistic and "deterministic" methods is discussed in §3.2.1). Well-formulated algorithms impose explicit error models for both the data and the parameter estimates based on the structure of the problem at hand, while less rigorous methods treat error only in selected quantities.

One example highlighting the importance of uncertainty formulation is the severe instability of the classic eight-point algorithm [LH81] in the presence of noise. This issue was addressed by Hartley, who showed that normalizing the data stabilizes the algorithm [Har97], and has since been revisited in many contexts. Matei presents general *heteroscedastic* errors-in-variables models in which noise characteristics vary across parameters, rigorously demonstrating the precise corrections necessary for optimal solution of the eight-point algorithm as well as several other classic vision problems [MM00].

Considerable mathematical theory has been developed involving the effects of measurement errors (*errors-in-variables*) on parameter estimates [Ste85, AF84, MM00]. Similar formulations can be found in *total least squares* problems [GL80, Nie94, NFH00] and perturbation theory [DC87, Ste90, SS90]. Nearly all formulations assume the most convenient error model, which is additive Gaussian noise.

As will be discussed in §3.1.4, this additive Gaussian noise model is appropriate in many situations. There is a tendency, however, to overuse this model, applying it to problems in which it has no meaningful interpretation or violates assumptions about the underlying parameter space. For example, in [CZZF97] and [Zha98], recovery of the covariance of the fundamental matrix is described; however, its physical meaning (e.g. its units) is unclear.

Uncertainty in image features is usually treated in the Euclidean image plane, which is poorly suited for projective feature representations. The most notable treatment of uncertainty in projective features is due to Collins, who utilizes axial probability distributions on the sphere to represent measurement error and to perform inference tasks [CW90]. Other work has been presented by [NS94], who use an effective hybrid of projective and Euclidean probability distributions to describe the combination of rotational and translational pose uncertainty. Further discussion of these topics is deferred to §3.3.1.

2.2 The City Project

As mentioned previously, three-dimensional reconstruction has many useful applications. Working systems have recently been developed, both commercially and in research laboratories, which can reliably recover 3-D structure and texture of small-scale objects from a few images. However, there is relatively little research concerning reconstruction of large-scale environments using large-scale input [FZ98, GLT99].

One example of such research is the MIT City Project, whose goal is detailed, metric reconstruction of urban landscapes [CMT98, Tel98]. To this end, it utilizes a custom-built platform that acquires *controlled imagery* (images in roughly-known configurations) and employs a set of unique, scalable algorithms. The system comprises several components, shown in Figure 2-1. The sections that follow describe these components as they existed before development of this thesis in order to provide motivation for robust pose recovery.

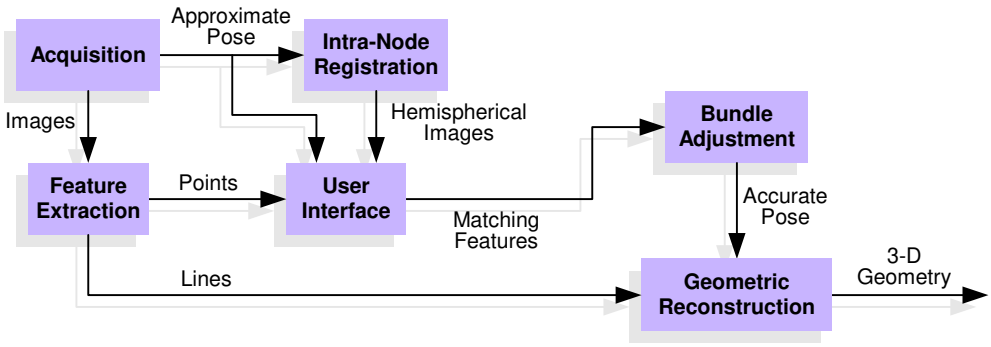


Figure 2-1: The City Project
Images and approximate camera pose are acquired and used to create a set of hemispherical images (nodes). A semi-automatic process externally registers the nodes, after which detailed scene geometry is estimated.

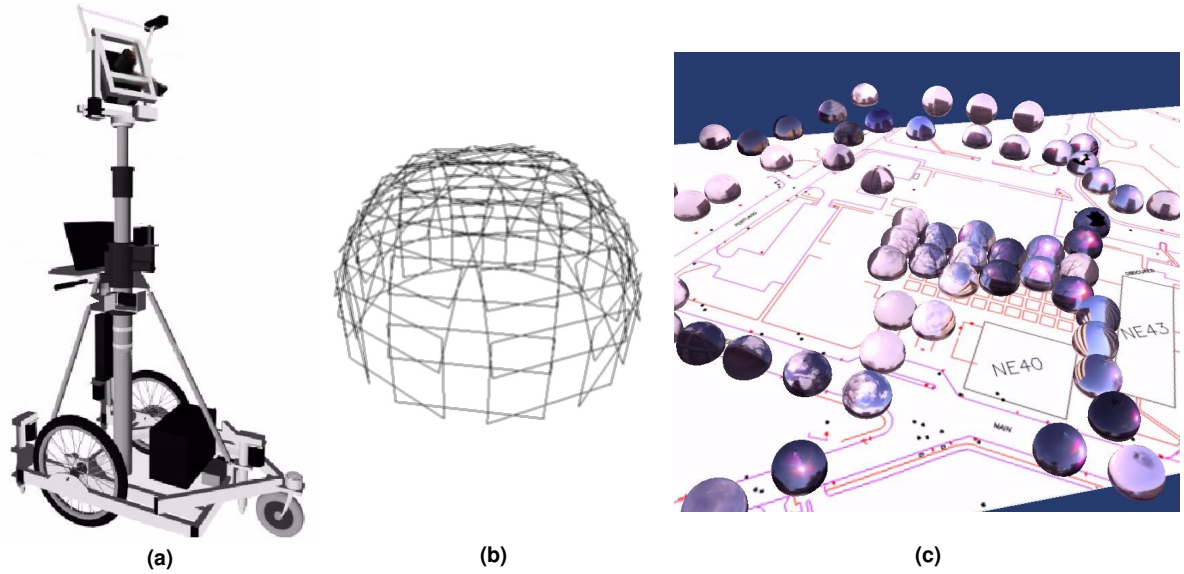
2.2.1 Data Acquisition

Raw data is acquired by a platform known as *Argus* [BT00], which is moved and controlled by a human operator. Argus is equipped with a single high-resolution digital camera mounted on an electronically controlled pan-tilt head that rotates the camera about its optical center to any specified orientation. An array of different devices, such as a global positioning system (GPS), inertial measurement unit (IMU), and odometers record various real-time measurements of the camera's pose.

The operator positions Argus, and at a given position the pan-tilt head rotates sequentially to a set of fixed orientations at which high-dynamic-range planar images are captured. The orientations are pre-calculated so that the resulting set of planar images (comprising a *node*) forms a partially-overlapping tiling of a hemispherical FOV, thus simulating an omnidirectional camera. Hundreds of nodes, each consisting of twenty to forty images, are acquired at various spatial positions (Figure 2-2).

Offline processes calibrate the camera's internal parameters and obtain rough estimates of external pose. Images of a calibration target with known geometry are acquired, and Tsai's method [Tsa87] is used to recover the first-order parameters (focal length, skew, and center of projection), as well as radial lens distortion coefficients. Images are undistorted (re-mapped) to square pixels according to these parameters. After actual field data is acquired, all sensor measurements are fused in an extended Kalman filter formulation [Ger99] to produce approximate external pose.

Input to subsequent vision algorithms thus consists of a large set of intrinsically corrected images accurate to within a few pixels and roughly-known pose. Although theoretically accurate to within approximately 2 degrees of heading and less than one meter of position [BT00], pose estimates are often corrupted by various sensor errors that can cause accuracy degradation of tens of degrees and several meters.



(a) (b) (c)

Figure 2-2: Argus Data Acquisition
(a) A prototype of the Argus acquisition platform, which collects image and pose data in hemispherical configurations by fusion of various sensor measurements. (b) A typical image configuration tiling the hemisphere. (c) Node locations registered with an actual ground map.

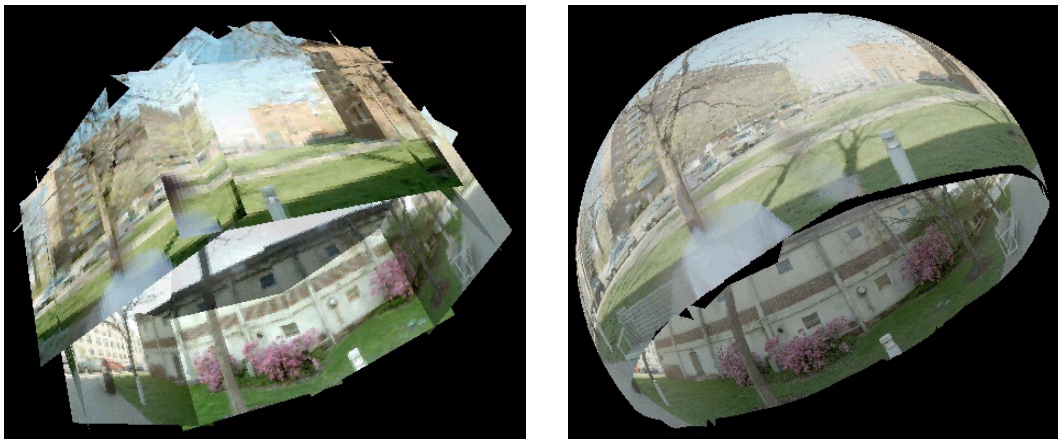
2.2.2 Hemispherical Image Registration

Measurement error, miscalibration, and inherent limitations on sensor precision contribute to slight misalignment between images in a given node. If node images are to be treated as a single omnidirectional image, they must be precisely registered so that a ray through any pixel corresponds to the correct projective ray through the camera's optical center.

After data acquisition, a mosaicing algorithm aligns the images within each node, maximizing dense correlation in image texture by iteratively adjusting rotations about the optical center and first-order camera intrinsics until image rays optimally coincide [CMT98, KMT00]. The effective result for a given node, shown in Figure 2-3, is a true hemispherical image for which the 3-D ray

through any pixel can easily be calculated. This fusion of many discrete rectangular images into a single entity greatly simplifies projective inference tasks and data management; hemispherical images also have other desirable properties, which were described briefly in §1.2.2 and §1.3.1.

It should be noted that the resulting mosaics are not perfect. Uncertainty in this process is not explicitly modeled, but error can certainly affect subsequent system stages and derived quantities.



(a) (b)

Figure 2-3: Creation of Image Mosaics
A partial tiling of rectangular images on the sphere serves to form a hemispherical image. (a) The original tiling. (b) Results of the mosaicing process.

2.2.3 Manual Pose Refinement

Once intra-node registration is complete, error in the external pose initially estimated by Argus must be corrected. Nodes are registered via a semi-automatic bundle adjustment process based on user-correlated point features.

First, line features are generated in each of the original planar images. A Canny edge detector [Can86] extracts edge pixels, which are then grouped into collinear sets by a clustering algorithm. Line parameters as in (1-10) are estimated in a linear least-squares formulation where edge pixels are weighted according their gradient magnitudes. Point features are derived from actual or hallucinated intersections between adjacent line segments.

Next, explicit correspondence between point features in different nodes is specified manually through a machine-assisted user interface. When enough correspondences have been identified, a standard bundle adjustment algorithm successively refines 3-D point feature positions and camera pose, minimizing the spatial discrepancy between extruded feature rays until convergence is reached. This technique does not utilize or produce uncertainty in the features or in the final pose estimates.

2.2.4 Geometric Reconstruction

Various algorithms can be applied to the images and newly registered cameras in order to infer three-dimensional structure. Although not the direct focus of this work, reconstruction of scene

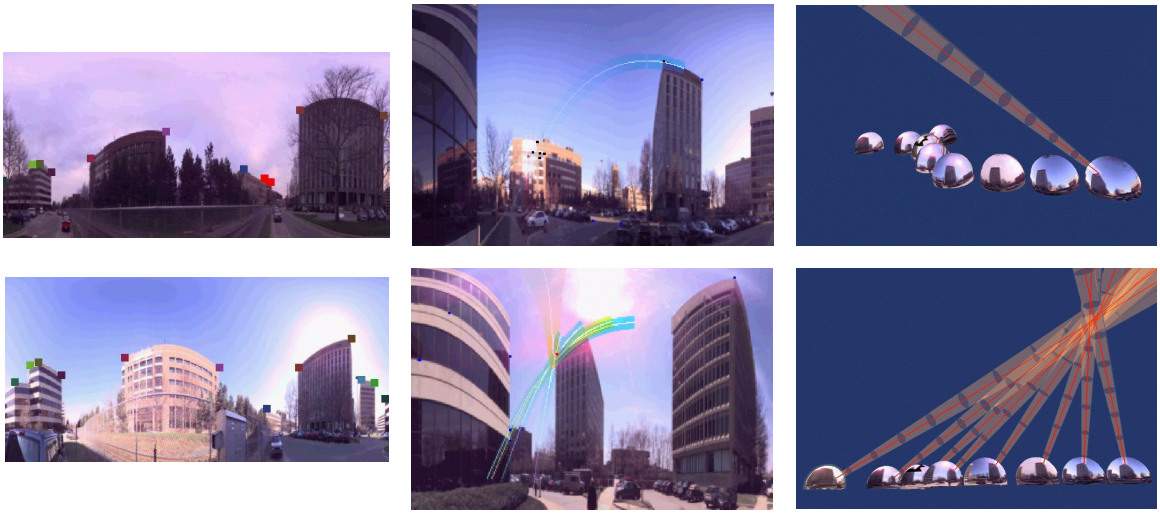


Figure 2-4: Semi-Automatic Pose Refinement
External pose is refined by a human-assisted process. (a) Examples of correlated point features. (b) Epipolar lines from single and multiple observations of the same building corner. (c) Uncertain 3-D rays extruded from single and multiple nodes.

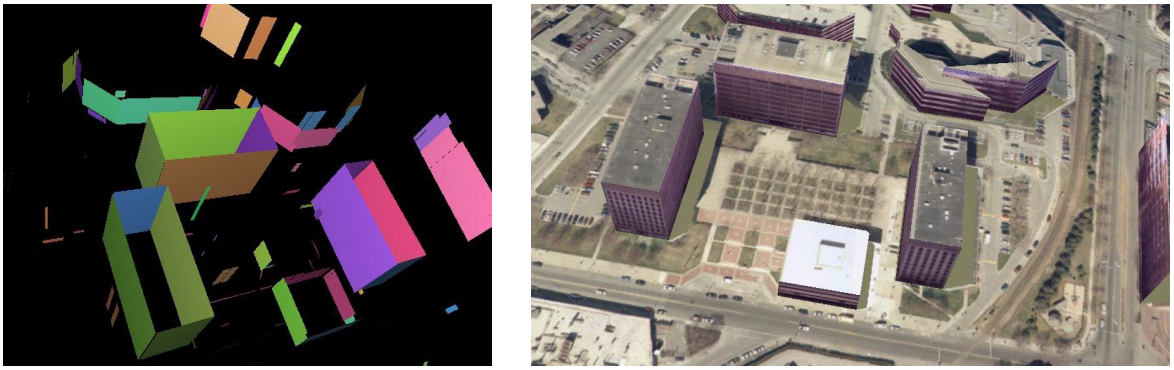
geometry motivates accurate camera pose recovery and estimation of uncertainty. Currently a space-sweep algorithm, which assumes the scene to be composed of planar vertical façades, recovers piecewise planar structure based on correlations between line features such as window edges [CT99]. Subsequent algorithms relax the planar assumption, estimating finer detail on the surface of the façades.

If accurate camera pose is known, a host of other techniques can also be used to extract 3-D structure. One such technique hypothesizes dense sets of small, oriented 3-D surface patches (*surfels*) [Mel99]. Others are based on the direct fusion of point and line features [CT97], and still others quantize space into discrete volume elements (*voxels*) and compute incidence of image rays on each voxel to determine model opacity and color.

2.2.5 Motivations for Automated Registration

While the pipeline described above is effective, improvements are clearly possible. Pose refinement, which is the only system component (aside from data acquisition) that requires human input, is the most obvious candidate for improvement.

Manual feature identification and assignment is a fairly simple process, especially when assisted by a computer, but, as mentioned in §2.1.2, it is tedious and quickly becomes infeasible as the number of acquired nodes grows. Correspondence is also subject to human error; for example, the operator often cannot distinguish between ambiguous matches (such as particular window corners of large buildings with dense window tiling), especially when there is significant occlusion (Figure 2-6). Because of scale issues and the difficulty involved in choosing consistent feature sets, operators typically specify the minimum number of matches necessary to produce a unique solution. In the presence of noise, which inevitably stems from image distortions, mosaic errors,



(a) (b)

Figure 2-5: Geometric Reconstruction
A plane sweep algorithm extracts vertical façades. (a) Hypothesized planar surfaces before thresholding. (b) The reconstructed scene with overlaid aerial texture.

and misdetection of features, using the minimum number produces inaccurate and unstable pose estimates.

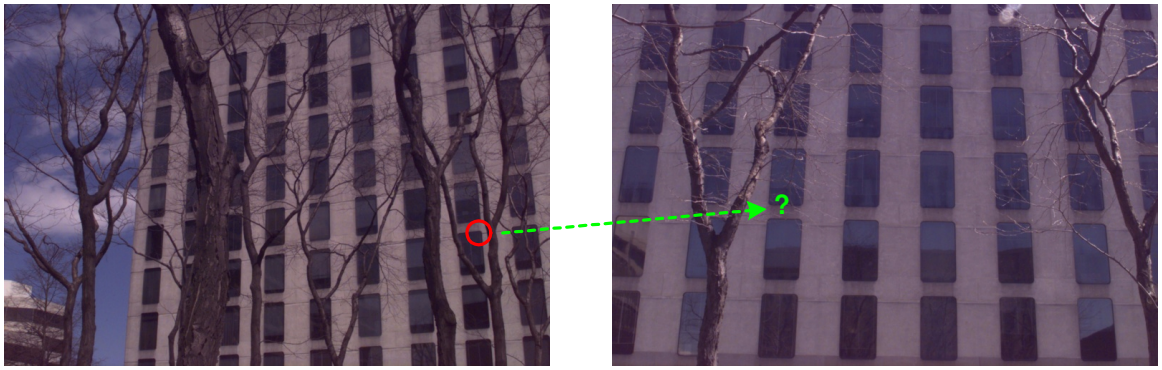


Figure 2-6: Correspondence Ambiguities
User-assisted feature correspondence can be difficult when there is regular geometry and significant occlusion. In this example, a particular window corner in one view does not have a clear match in another.

The failings of manual correspondence methods motivate fully automatic camera pose recovery. Instabilities caused by minimal feature sets can be overcome by exploiting the redundancy of image data, especially the large, dense sets typical of the City Project. Estimation theory and laws of large numbers (§3.1.4) imply that the certainty with which parameters are estimated increases with the number of observations; robust, statistically sound estimates of pose and structure can thus be obtained by utilizing as much relevant information as possible.

Geometric reconstruction algorithms rely heavily on external pose, so knowledge of pose uncertainty can improve accuracy in the resulting scene structure by giving large weight to reliable pose and discounting unreliable pose. Another motivation for this work is thus to provide meaningful uncertainty models and to increase the quality of existing error measures in the system.

As a final note, although pose recovery is motivated by and tailored to a specific system, the underlying methods in this thesis have been developed with more general systems in mind, and can thus be applied to a variety of situations.

2.3 Contributions

This work addresses many limitations of systems described in §2.1. Although existing algorithms and modeling techniques are utilized, several improvements and novel ideas have also emerged. The system developed in this work has practical benefit as well as theoretical rigor, and its performance is demonstrated on large sets of real and synthetic data. Care has also been taken to realize efficient implementations of all techniques.

Many of the system's unique properties were outlined briefly in the previous chapter. Several additional advantages stem from the context of the City Project. Knowledge of approximate initial pose allows for arbitrary camera configurations without extensive manual calibration, and provides precisely the type of prior information necessary to ensure convergence on globally optimal solutions. Hemispherical images, although not strictly required by these techniques, reduce bias in inference problems and eliminate motion ambiguities [YC92, Wil94, BKY97, FA98].

Specific contributions of this work are outlined below and described in more detail throughout the remainder of the dissertation.

2.3.1 Uncertainty Models

A unified treatment of uncertainty is developed and propagated through all stages of the camera registration process. Projective features, pose, and other derived quantities are treated as random variables, or samples from probability distributions defined on appropriate spaces, resulting in more accurate estimates of the final camera pose and its uncertainty. Such probabilistic treatment of all parameters is rather rare in existing vision systems, which often view data deterministically or use heuristically-obtained scalar weights.

This work makes extensive use of projective features. The notion of *dual uncertainty* is introduced, and the work of Collins is extended to unify inference of bipolar and equatorial distributions. Line features are treated as equatorial distributions on the sphere which are estimated by fusion of the image gradient. It is shown that projective uncertainty can be transformed into Euclidean uncertainty by manipulating the second-moment matrix of a spherical probability distribution.

New stochastic models of rotations are presented that describe random quaternions as distributions on a compact space. Correspondences between vanishing points are modeled as noisy samples from such distributions and fused in rotational inference problems, also producing uncertainty in the resulting rotation estimates.

2.3.2 Hough Transform Methods

The Hough transform, described in Chapter 4, is central to this work. Practical implementation of the transform itself is improved in several ways. A new peak detection algorithm finds statistically significant points of maximum incidence in the transform. The notion of a proximity function is

introduced and used to develop simple methods for anti-aliasing and incorporation of uncertainty in the contributing features.

The output of the transform is used only to initialize more accurate refinement techniques, thus reducing the need for overly sophisticated or complicated implementations. In vanishing point detection, the transform serves to determine the number and approximate location of prominent 3-D line directions. The entire projective plane is efficiently parameterized by a unit cube centered at the focal point, providing a closed and easily-discretized surface. In the estimation of two-camera translation, the Hough transform is used to find an approximate camera baseline direction by incorporating all possible point feature correspondences. This allows for robust initialization of accurate epipolar estimation techniques without requiring explicit correspondence. The transform is also shown to provide adequate prior distributions on parameters for Bayesian inference.

2.3.3 Vanishing Point Estimation

The estimation of 3-D line directions, or vanishing points, is crucial to rotational camera registration. Previous approaches either exclusively employ the Hough transform, which is robust but inherently inaccurate, or estimate one vanishing point at a time using a previously-acquired grouping of image lines into parallel sets. The approach developed here models vanishing points as a mixture of probability distributions on the sphere, using an expectation maximization algorithm (described in §4.2) to simultaneously estimate their 3-D directions and classify their constituent image lines. Initialization of this algorithm by a Hough transform is shown to be quite robust against outlier features and to provide accurate estimates of 3-D line directions.

2.3.4 Rotational Registration

A classical absolute orientation method is applied to two-camera registration using vanishing points as features, and is extended in three ways: first, feature uncertainty is incorporated into the estimation; second, uncertainty in the resulting rotation estimate is produced using a probability distribution on the 4-D hypersphere of unit quaternions; and third, the method is generalized to an arbitrary number of cameras.

The extension to multiple cameras also introduces new techniques for rotational bundle adjustment. A nested expectation maximization technique simultaneously performs probabilistic correspondence and registration of vanishing points in a mixture model formulation.

2.3.5 Translational Registration

A new geometric framework is illustrated for determining the optimal direction of motion between two cameras with known rotational pose. It is shown that this pair-wise motion problem has a simple and elegant geometric interpretation which allows inference techniques similar to those of line and vanishing point estimation to be employed. In addition, several types of novel geometric constraints are imposed as prior knowledge that reject or accept candidate point matches according to the likelihood of their physical occurrence. These constraints involve information obtained from approximate camera pose and previously estimated 3-D line directions.

A Hough transform technique is introduced which incorporates all possible matches between point features in a pair of cameras. The technique identifies the most likely pair-wise translational

direction, and a subsequent Markov chain Monte Carlo algorithm refines this direction by optimizing stochastically over all possible correspondence sets, thereby circumventing the need for explicit correspondence. The algorithm also handles outlier features and occlusion.

Finally, a method for incorporating all pair-wise direction constraints and their uncertainty into a global optimization is developed which utilizes linear least-squares optimization and produces estimates of uncertainty in the final camera positions.

2.3.6 End-to-End Results

Experiments are performed on simulated data sets to assess the performance of various stages of the pose recovery system, and to characterize and quantify its behavior with varying types and levels of input corruption. Results obtained from real large-scale data sets are also presented, and the consistency of pose estimates is assessed using a variety of different error measures [LLF98]. Examples include misalignment of corresponding 3-D feature rays in metric units, 2-D epipolar consistency in image pixels, and bounds on estimated uncertainty.

CHAPTER 3

Stochastic Geometry

DATA IS INVARIABLY SUBJECTED to various types of corruption, or *noise*, which can be viewed as the discrepancy between observed quantities and the "perfect" predictions of a given model. No real sensor can obtain perfect measurements, and no real computational device can represent numbers to infinite precision. In truth, what is often considered to be noise results from incomplete or inaccurate mathematical models, unaccounted parameters such as roundoff and quantization artifacts, and unmeasurable phenomena in the underlying physical process.

Noise corrupts virtually all stages of vision systems. For example, small vibrations and electromagnetic disturbances during image acquisition can cause fluctuation in pixel values and other sensor readings. Imprecise estimates of internal camera parameters, or unmodelled lens effects such as radial distortion, perturb measured projective rays from their true directions. Low-contrast image regions compromise the accuracy of edge detection techniques, and small or unreliable image features corrupt geometric inference tasks. Thus error in the system accumulates, beginning with the corruption of sensor measurements and compounded by imperfect models and imprecise computation.

Since noise originates from unmeasured (or unmeasurable) physical processes and quickly accumulates in subsequent manipulations, accounting for it explicitly is in general a hopeless task. This is not to say that noise should be ignored; although individual noise *values* may not be obtainable, it is often possible to characterize the general behavior of the noise *process*. Additional information such as redundant measurements or extension of the underlying models improves the accuracy of parameter estimates and produces tighter confidence bounds around them.

In addition, some measurements are inherently more reliable than others. It is sensible, therefore, to give stronger emphasis to reliable data than to unreliable data in estimation tasks. Models that accurately capture the behavior of noise processes allow for such emphasis and thus produce better estimates.

By definition, noise represents *uncertainty* in observed quantities. This uncertainty can be viewed as randomness, and uncertain quantities can then be treated as the outcomes of a *stochastic*

process, or samples from a probability distribution defined over the appropriate space of parameters. Such interpretations open problems to a large body of powerful mathematical theory.

As its name suggests, *stochastic geometry*, sometimes called geometric probability, concerns the study of uncertainty in geometric entities. It is a somewhat specialized branch of general probability theory, relying on the same basic axioms and methodologies but applying them to variables that represent spatial relationships, shapes, and geometric constraints [KM63, Sol78].

This chapter begins with a review of basic probability theory in §3.1 and statistical inference techniques in §3.2. These fundamental constructs are then applied to relevant entities such as projective image tokens (§3.3) and camera pose (§3.4) to formulate precise descriptions of uncertainty in these entities. §3.5 describes methods for fusion of many uncertain projective features into more accurate summarizing quantities and for obtaining other derived distributions.

3.1 Elementary Probability Theory

This section reviews the fundamental axioms and constructs of general probability theory, which naturally also apply to geometric quantities. Concepts and notation used throughout this dissertation are presented in the sections that follow.

3.1.1 Events

Although there are many different views on the precise definition of probability—the most notable being relative frequency and axiomatic—the notion of stochastic experiments and outcomes is more universally accepted: an *experiment* is any well-defined act that results in a well-defined *outcome* [Dra67]. For example, the act of measuring light intensity at a pixel with a CCD is a stochastic experiment whose outcome is the measurement value itself; the act of edge detection is an experiment whose outcome is a set of edge pixel locations.

For a given experiment, one must define a *sample space* Ω, i.e. the set of all possible outcomes of the experiment. An *event* on a sample space is a possibly empty collection of experimental outcomes; the event can be as fine-grained as an individual experimental outcome, or as coarse as the entire space. Choice of an appropriate sample space is perhaps the most crucial element in the formulation of a stochastic model.

Since an event simply represents a particular collection of experimental outcomes, it is often convenient to use set notation to describe combinations of events. For example, if $\mathcal{A}$ and $\mathcal{B}$ are two events defined on a sample space Ω, the event that $\mathcal{A}$ or $\mathcal{B}$ occurs is denoted by the union operator, $\mathcal{A} \cup \mathcal{B}$, and the event that both $\mathcal{A}$ and $\mathcal{B}$ occur is denoted by the intersection operator, $\mathcal{A} \cap \mathcal{B}$, defined in the usual way.

Finally, a *probability law* is a mapping from events in the sample space to the real numbers, $\Omega \rightarrow \mathbb{R}$. The probability of an event $\mathcal{A}$ is denoted as $P(\mathcal{A})$. According to the so-called axiomatic definition, every valid probability law satisfies three axioms, from which all other properties can be derived:

- ***Nonnegativity***. The probability of an event cannot be negative, i.e. $P(\mathcal{A}) \geq 0$.
- ***Normalization***. The probability of the entire sample space is unity, $P(\Omega) = 1$.

- ***Additivity***. If $\mathcal{A}$ and $\mathcal{B}$ are disjoint events, then $P(\mathcal{A} \cup \mathcal{B}) = P(\mathcal{A}) + P(\mathcal{B})$.

3.1.2 Random Variables

A *random variable* $\boldsymbol{x}$ is a real-valued quantity in $\mathbb{R}^n$ whose values represent particular events in Ω. With each random variable $\boldsymbol{x}$ is associated a probability density function, denoted as $p(\boldsymbol{x})$, which is a probability law that maps values of the random variable to the real numbers $\mathbb{R}$ and, of course, satisfies the basic axioms of §3.1.1. A density function is defined so that

$$\int_{\mathcal{S}} p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = P(\boldsymbol{x} \in \mathcal{S}). \tag{3-1}$$

In other words, the integral of the density function over any region $\mathcal{S}$ in the sample space is equal to the probability that $\boldsymbol{x}$ takes some value in $\mathcal{S}$. Note that by the normalization axiom, the probability in (3-1) approaches unity as the size of $\mathcal{S}$ increases; since $P(\boldsymbol{x} \in \Omega) = 1$, it must be true that $\int_{\Omega} p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = 1$ as well.

With this framework in place, one can perform quantitative manipulations of the variables and deduce useful properties. Define $\boldsymbol{g}(\boldsymbol{x})$ as any function of the random variable $\boldsymbol{x}$; then the expectation of $\boldsymbol{g}(\boldsymbol{x})$ with respect to $\boldsymbol{x}$ is defined by

$$E_{\boldsymbol{x}}[\boldsymbol{g}(\boldsymbol{x})] \equiv \langle \boldsymbol{g}(\boldsymbol{x}) \rangle_{\boldsymbol{x}} \equiv \int_{S} \boldsymbol{g}(\boldsymbol{x}) p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}. \tag{3-2}$$

When $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{x}$, this expectation is called the *mean* of $\boldsymbol{x}$, or $\boldsymbol{\mu}_{\boldsymbol{x}}$. The *second moment* of $\boldsymbol{x}$ is the expectation when $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{x}\boldsymbol{x}^{\top}$. Finally, the *covariance* $\boldsymbol{\Lambda}_{\boldsymbol{x}}$ of $\boldsymbol{x}$ is obtained from the expectation when $\boldsymbol{g}(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{\mu}_{\boldsymbol{x}})(\boldsymbol{x} - \boldsymbol{\mu}_{\boldsymbol{x}})^{\top}$. The mean and covariance are useful quantities that will be used extensively throughout this work; the mean represents the "average" value of the variable, and the covariance represents the "spread" of its probability distribution or the magnitude of its randomness around the mean. As the covariance approaches zero, the variable becomes deterministic; the covariance is thus useful for describing a variable's uncertainty.

Two commonly used density functions are also salient to this work. The *uniform* density has constant value c over a finite region $\mathcal{S}$, and is given by the inverse of the region's area

$$p(\boldsymbol{x}) = c = \frac{1}{\int_{\mathcal{S}} \mathrm{d}\boldsymbol{x}}, \tag{3-3}$$

inside $\mathcal{S}$ and zero elsewhere. The *normal* or *Gaussian* density is defined over all of $\mathbb{R}^n$ and is specified completely by its mean and covariance:

$$p(\boldsymbol{x}) = \mathcal{N}_n(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Lambda}) = \frac{1}{(2\pi)^{n/2} |\boldsymbol{\Lambda}|^{1/2}} \exp\left(-\frac{1}{2} (\boldsymbol{x} - \boldsymbol{\mu})^{\top} \boldsymbol{\Lambda}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) \right) \tag{3-4}$$

where $\boldsymbol{\mu}$ is the $n \times 1$ mean vector and $\boldsymbol{\Lambda}$ is the $n \times n$ covariance matrix of $\boldsymbol{x}$. The Gaussian density belongs to the exponential family of distributions and has numerous useful properties, which will be described in §3.1.4.

Probability densities can also be defined over multiple variables. The *joint density* of random variables $\boldsymbol{x}$ and $\boldsymbol{y}$ is denoted $p(\boldsymbol{x}, \boldsymbol{y})$, and by the normalization property, it must integrate to unity

over the new sample space of both $\boldsymbol{x}$ and $\boldsymbol{y}$. The marginal density of a single variable can be obtained from the joint density by simple integration:

$$p(\boldsymbol{x}) = \int p(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}\boldsymbol{y}. \tag{3-5}$$

This expression represents a projection of the multivariate density onto the sample space of one variable.

3.1.3 Conditioning

It is often useful to determine the probability of some event $\mathcal{A}$ in light of additional information provided by another event $\mathcal{B}$. The notation for such a quantity is $P(\mathcal{A}|\mathcal{B})$, read "the probability of $\mathcal{A}$ given $\mathcal{B}$" or "the probability of $\mathcal{A}$ conditioned on $\mathcal{B}$". Intuitively, knowledge of $\mathcal{B}$'s occurrence restricts the sample space to include only outcomes in $\mathcal{B}$. The conditional probability is thus equivalent to the event that both $\mathcal{A}$ and $\mathcal{B}$ have occurred, but must be divided by the probability that $\mathcal{B}$ occurred to satisfy the normalization axiom.

More formally, the conditional probability for events is defined as

$$P(\mathcal{A}|\mathcal{B}) = \frac{P(\mathcal{A} \cap \mathcal{B})}{P(\mathcal{B})} \tag{3-6}$$

assuming that $P(\mathcal{B})$ is non-zero. For random variables, the notation is as expected:

$$p(\boldsymbol{x}|\boldsymbol{y}) = \frac{p(\boldsymbol{x} \cap \boldsymbol{y})}{p(\boldsymbol{y})} \tag{3-7}$$

When $P(\mathcal{A}|\mathcal{B}) = P(\mathcal{A})$, events $\mathcal{A}$ and $\mathcal{B}$ are said to be *independent*, meaning that knowledge of one event gives no additional information about the other. Similarly, when $p(\boldsymbol{x}|\boldsymbol{y}) = p(\boldsymbol{x})$, the random variables $\boldsymbol{x}$ and $\boldsymbol{y}$ are independent. By (3-7), the independence of variables implies that their joint density can be expressed as the product of the individual densities:

$$p(\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{x})p(\boldsymbol{y}). \tag{3-8}$$

Simple manipulation of (3-7) also allows *inference* of $p(\boldsymbol{y}|\boldsymbol{x})$ from $p(\boldsymbol{x}|\boldsymbol{y})$ via *Bayes' Rule*,

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{y})p(\boldsymbol{y})}{p(\boldsymbol{x})}, \tag{3-9}$$

which forms the core of so-called Bayesian inference techniques, described further in §3.2.2.

3.1.4 Laws of Large Numbers

The Gaussian density has a number of distinguishing properties; for example, it exhibits ellipsoidal symmetry about its mean, its sample space is unbounded, and it can be characterized completely by its mean and covariance. In addition, this density lends itself well to inference problems because it has exponential form (see §3.2.1 below) and because of the remarkable *central limit theorem*.

Rigorous details are omitted here; at its core, however, the theorem states that if a random variable $\boldsymbol{S}_k$ is defined as the sum of k other variables $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k\}$, then the distribution of $\boldsymbol{S}_k$ approaches a Gaussian distribution $\mathcal{N}_n(\boldsymbol{S}_k; \boldsymbol{\mu}_{\boldsymbol{S}}, \boldsymbol{\Lambda}_{\boldsymbol{S}})$, *regardless of the distributions of the individual* $\boldsymbol{x}_i$ [Dra67]. If the $\boldsymbol{x}_i$ are independent and have mean $\boldsymbol{\mu}_i$ and covariance $\boldsymbol{\Lambda}_i$, then manipulations of expected values show that

$$\boldsymbol{\mu}_{\boldsymbol{S}} = \sum_{i=1}^{k} \boldsymbol{\mu}_i, \qquad \boldsymbol{\Lambda}_{\boldsymbol{S}} = \sum_{i=1}^{k} \boldsymbol{\Lambda}_i. \tag{3-10}$$

Thus, if uncertainty in a given measured quantity represents the aggregation of many unknown error processes (often the case in vision), then the uncertainty can be summarized fairly accurately by a Gaussian distribution.

Assume now that the $\boldsymbol{x}_i$ are independent and identically distributed (IID) samples; that is, they all have the same arbitrary probability density function with $\boldsymbol{\mu}_i = \boldsymbol{\mu}$ and $\boldsymbol{\Lambda}_i = \boldsymbol{\Lambda}$. Further, define the random variable $\boldsymbol{A}_k$ as the sample average, that is

$$\boldsymbol{A}_k = \frac{\boldsymbol{S}_k}{k} = \frac{1}{k} \sum_{i=1}^{k} \boldsymbol{x}_i. \tag{3-11}$$

The mean and covariance of $\boldsymbol{A}_k$ are

$$\boldsymbol{\mu}_k = \boldsymbol{\mu}, \qquad \boldsymbol{\Lambda}_k = \frac{1}{k} \boldsymbol{\Lambda}. \tag{3-12}$$

Note that as k increases, the covariance of the sample average decreases. Qualitatively, this result states that the sample average converges in probability to the true mean $\boldsymbol{\mu}$. This is stated more formally by the *weak law of large numbers*:

$$\lim_{k \to \infty} P(|\boldsymbol{A}_k - \boldsymbol{\mu}| > \varepsilon) = 0 \quad \text{for any } \varepsilon > 0. \tag{3-13}$$

As the number of samples increases, the probability that the average differs significantly from the true mean goes to zero. This result is important in data fusion applications, where many uncertain measurements are summarized by a few parameters: it states that the certainty of the parameters increases as the sample population grows.

Despite implications to the contrary, using a larger data set does not necessarily increase robustness or accuracy. The above arguments apply only to IID variables, that is observations which are all generated by the same random process, and not to *outliers*, or data generated by separate, perhaps unmodeled processes. If the number of outliers remains constant as the sample population grows—i.e. if their relative fraction decreases—then their collective effect will be overwhelmed; the same is true if outliers are explicitly modeled or removed from consideration; otherwise, they can skew or *bias* the parameter estimates.

3.2 Statistical Inference

One often wishes to characterize the process that generated a set of observed data points or measurements $\mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_k\}$, summarizing the measurements by a few parameters. If the

data is treated as a set of random samples from some probability distribution $p(\boldsymbol{x})$ described by unknown parameters $\boldsymbol{\Theta}$, a set of powerful statistical inference methods can be applied to extract meaningful information about the underlying process. The generating distribution, denoted as $p(\boldsymbol{x}_i|\boldsymbol{\Theta})$ to make explicit the dependence on the parameter values, is assumed to be of known form (e.g. Gaussian). Further, in the absence of knowledge about correlation among the data, all samples $\boldsymbol{x}_i$ are assumed to be drawn independently from this distribution.

Two general techniques for estimation of the generating density, namely maximum likelihood and Bayesian inference, are presented below. Both are used extensively in vision, learning, and many other contexts, and can often lead to similar results; however, each offers a somewhat unique perspective on the problem.

3.2.1 Maximum Likelihood Estimation

The method of *maximum likelihood* (ML) finds a particular set of parameters $\boldsymbol{\Theta}$ that best explains the data. As its name implies, the objective in ML estimation is maximization of a *likelihood function* $L(\boldsymbol{\Theta})$ that expresses the joint probability of all observations $\mathcal{X}$ as an explicit function of the parameters $\boldsymbol{\Theta}$. The likelihood of a single observation is given by $p(\boldsymbol{x}_i|\boldsymbol{\Theta})$. Since samples are assumed to be independent, the joint likelihood of all observations is simply the product of the individual likelihoods:

$$\begin{aligned} L(\boldsymbol{\Theta}) &= p(\mathcal{X}|\boldsymbol{\Theta}) \\ &= \prod_{i=1}^{k} p(\boldsymbol{x}_i|\boldsymbol{\Theta}). \end{aligned} \tag{3-14}$$

The problem is now to estimate the most likely value of $\boldsymbol{\Theta}$ given the data set—i.e. the value that maximizes (3-14)—which requires knowledge of the form of $p(\boldsymbol{x}_i|\boldsymbol{\Theta})$. In many situations the central limit theorem is applied, because nothing specific is known about underlying process; the individual samples are assumed to be drawn from a Gaussian distribution $\mathcal{N}_n(\boldsymbol{x}_i; \boldsymbol{\mu}_i(\boldsymbol{\Theta}), \boldsymbol{\Lambda}_i(\boldsymbol{\Theta}))$, where the mean and covariance are expressed as explicit functions of the parameters. If the distribution family is exponential, as is the case with Gaussian densities, it is convenient to maximize the logarithm of the likelihood function rather than the likelihood itself:

$$\begin{aligned} \text{argmax}_{\boldsymbol{\Theta}}[p(\mathcal{X}|\boldsymbol{\Theta})] &= \text{argmax}_{\boldsymbol{\Theta}}[\log p(\mathcal{X}|\boldsymbol{\Theta})] \\ &= \text{argmax}_{\boldsymbol{\Theta}} \left[\sum_{i=1}^{k} \log p(\boldsymbol{x}_i|\boldsymbol{\Theta}) \right]. \end{aligned} \tag{3-15}$$

The parameter values that maximize the expressions in (3-14) and (3-15) are equivalent because the likelihood function is non-negative and the logarithm is a monotonically increasing function.

If $p(\boldsymbol{x}_i|\boldsymbol{\Theta}) = \mathcal{N}_n(\boldsymbol{x}_i; \boldsymbol{\mu}_i, \boldsymbol{\Lambda}_i)$ as in (3-4), where the dependence on $\boldsymbol{\Theta}$ is made implicit for brevity, the expression in (3-15) becomes

$$\begin{aligned} & \text{argmax}_{\boldsymbol{\Theta}} \left[-\frac{n}{2} \sum_{i=1}^{k} \log(2\pi) - \frac{1}{2} \sum_{i=1}^{k} \log |\boldsymbol{\Lambda}_i| - \frac{1}{2} \sum_{i=1}^{k} (\boldsymbol{x}_i - \boldsymbol{\mu}_i)^\top \boldsymbol{\Lambda}_i^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_i) \right] \\ = \; & \text{argmin}_{\boldsymbol{\Theta}} \left[\sum_{i=1}^{k} \log |\boldsymbol{\Lambda}_i| + \sum_{i=1}^{k} (\boldsymbol{x}_i - \boldsymbol{\mu}_i)^\top \boldsymbol{\Lambda}_i^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_i) \right] \end{aligned} \tag{3-16}$$

which involves no exponentials. Estimation typically proceeds by equating to zero the partial derivatives of this new likelihood function with respect to the parameters and solving the resulting system of equations. Analytic solution is often possible for exponential distributions.

It is interesting to note a connection between the maximum likelihood formulation and standard least-squares minimization by considering two assumptions. The first is that the underlying distributions are Gaussian, so that the log-likelihood has the form in (3-16); the second is that the covariance in (3-16) is constant with respect to the parameters and is of the known form $\boldsymbol{\Lambda}_i(\boldsymbol{\Theta}) = \sigma_i \boldsymbol{I}$, where $\boldsymbol{I}$ denotes the $n \times n$ identity matrix. If both conditions are satisfied, then the formulation simplifies to

$$\begin{aligned} & \text{argmin}_{\boldsymbol{\Theta}} \left[\sum_{i=1}^{k} \frac{1}{\sigma_i} (\boldsymbol{x}_i - \boldsymbol{\mu}_i(\boldsymbol{\Theta}))^\top (\boldsymbol{x}_i - \boldsymbol{\mu}_i(\boldsymbol{\Theta})) \right] \\ = \; & \text{argmin}_{\boldsymbol{\Theta}} \left[\sum_{i=1}^{k} \frac{1}{\sigma_i} \|\boldsymbol{x}_i - \boldsymbol{\mu}_i(\boldsymbol{\Theta})\|^2 \right]. \end{aligned} \tag{3-17}$$

The expression in (3-17) is the optimal weighted least-squares estimator of a given set of a functional relationship between a given set of data points. Least-squares estimation can thus be viewed as a special case of maximum likelihood in which the underlying probability distributions are Gaussian.

3.2.2 Bayesian Estimation

A more general inference method, *Bayesian inference*, describes the *distribution* of the parameters rather than specific values, thus treating $\boldsymbol{\Theta}$ as a random variable with unknown probability density function. Bayesian techniques make use of prior information about the parameters, if available, and convert the prior density into a posterior density by using the measurements and exploiting Bayes' Rule.

The goal is to determine the density function $p(\boldsymbol{x}|\mathcal{X})$, that is the probability density of random variable $\boldsymbol{x}$ given the observed data. This density can be expressed as a marginal density of the joint distribution of the data and the parameters,

$$p(\boldsymbol{x}|\mathcal{X}) = \int p(\boldsymbol{x}, \boldsymbol{\Theta}|\mathcal{X}) \mathrm{d}\boldsymbol{\Theta}. \tag{3-18}$$

Using the definition of conditional probability given in (3-7), this expression can be rewritten as

$$p(\boldsymbol{x}|\mathcal{X}) = \int p(\boldsymbol{x}|\boldsymbol{\Theta}, \mathcal{X}) p(\boldsymbol{\Theta}|\mathcal{X}) \mathrm{d}\boldsymbol{\Theta}. \tag{3-19}$$

Given a set of parameter values, the distribution of $\boldsymbol{x}$ is completely specified and thus independent of the observations $\mathcal{X}$, so the density simplifies to

$$p(\boldsymbol{x}|\mathcal{X}) = \int p(\boldsymbol{x}|\boldsymbol{\Theta}) p(\boldsymbol{\Theta}|\mathcal{X}) \mathrm{d}\boldsymbol{\Theta}. \tag{3-20}$$

This integral can be interpreted as a weighted average of densities over all possible parameter values, distinguishing Bayesian inference from ML techniques which choose a single "best" set of parameters.

Using Bayes' Rule on the second factor gives

$$p(\boldsymbol{x}|\mathcal{X}) = \int p(\boldsymbol{x}|\boldsymbol{\Theta})\frac{p(\mathcal{X}|\boldsymbol{\Theta})p(\boldsymbol{\Theta})}{p(\mathcal{X})}\mathrm{d}\boldsymbol{\Theta}. \tag{3-21}$$

If the data are assumed to be independent, the conditional likelihood becomes

$$p(\mathcal{X}|\boldsymbol{\Theta}) = \prod_{i=1}^{k} p(\boldsymbol{x}_i|\boldsymbol{\Theta}), \tag{3-22}$$

just as in the ML case, and the marginal density of the data $p(\mathcal{X})$, which is effectively a constant factor that ensures the normalization axiom, is given by

$$\begin{aligned} p(\mathcal{X}) &= \int p(\mathcal{X}|\boldsymbol{\Theta}_0)p(\boldsymbol{\Theta}_0)\mathrm{d}\boldsymbol{\Theta}_0 \\ &= \int p(\boldsymbol{\Theta}_0)\left[\prod_{i=1}^{k} p(\boldsymbol{x}_i|\boldsymbol{\Theta}_0)\right]\mathrm{d}\boldsymbol{\Theta}_0. \end{aligned} \tag{3-23}$$

Evaluation of the integrals in (3-21) and (3-23) is generally feasible only for classes of density functions in which the posterior density has the same functional form as the prior [Bis95]. Members of exponential distribution families such as the Gaussian and Bingham distributions (§3.3.2) are examples of such classes, which are said to be *closed* with respect to Bayesian inference.

One of the main distinctions between Bayesian and ML techniques is the explicit use of the prior density $p(\boldsymbol{\Theta})$, which represents knowledge about the parameters in the absence of measurements. For example, if nothing is known in advance about the parameters except that they lie in a restricted region of the sample space, then $p(\boldsymbol{\Theta})$ may be defined as a uniform density over that region.

Despite the seeming dissimilarity between the two techniques, there is a simple relationship between maximum likelihood and Bayesian formulations. The likelihood function given in (3-14) is proportional to the posterior density $p(\boldsymbol{\Theta}|\mathcal{X})$; by definition, the peak of the likelihood function occurs at the ML estimate $\tilde{\boldsymbol{\Theta}}$. If this peak is relatively sharp, then the integral in (3-20) is dominated by the region around $\tilde{\boldsymbol{\Theta}}$, and can be approximated as

$$\begin{aligned} p(\boldsymbol{x}|\mathcal{X}) &\approx p(\boldsymbol{x}|\tilde{\boldsymbol{\Theta}})\int p(\boldsymbol{\Theta}|\mathcal{X})\mathrm{d}\boldsymbol{\Theta} \\ &= p(\boldsymbol{x}|\tilde{\boldsymbol{\Theta}}). \end{aligned} \tag{3-24}$$

As stated by the laws of large numbers, the covariance, and thus the spread of the posterior distribution, decreases as the number of pertinent observations increases. The Bayesian density thus becomes deterministic and approaches the ML solution when a sufficiently large set of observations is available.

3.3 Projective Feature Uncertainty

As discussed in §1.3.2, projective feature representations are often more convenient and better conditioned than Euclidean representations, and as a consequence are commonly used in vision

applications. Notions of magnitude and sign along projective rays are irrelevant, so the projective plane $\mathbb{P}^2$ can be mapped to the surface of the unit sphere $\mathbb{S}^2$, which is a closed, compact, symmetric space, and projective rays can be mapped to antipodally symmetric axial points on the sphere.

$\mathbb{S}^2$ is therefore a suitable sample space for projective random variables, and proper density functions, like the projective features themselves, must be defined only on this surface and must exhibit antipodal symmetry. Since the sphere is a closed space, the uniform distribution is well-defined; however, Euclidean uncertainty representations such as the Gaussian distribution cannot be applied.

3.3.1 Past Work in Projective Uncertainty

Despite the extensive use of projective quantities in vision applications, there is surprisingly little discussion in the literature concerning accurate uncertainty models and inference for such quantities. Gaussian random variables defined on an unbounded Euclidean space are almost exclusively preferred, as they have numerous desirable properties; by contrast, although the unit hypersphere is the most elegant parameter space for projective features, it is a highly nonlinear space in which all but the simplest inference tasks can become unwieldy. Iterative numerical solution of complicated equations is often required, unlike the straightforward, closed-form solutions for Gaussian random variables.

Many authors treat feature measurements as deterministic quantities additively corrupted by samples from a zero-mean trivariate Gaussian noise process $\boldsymbol{\eta}$ (Figure 3-1a), i.e. $\boldsymbol{x}_{meas} = \boldsymbol{x}_{true} + \boldsymbol{\eta}$. [MM00, HZ00]. Kanatani [Kan92] and Antone [AT00b] impose the additional constraint that the noise covariance be singular, that is have a zero eigenvalue corresponding to the eigenvector in the direction of the measurement. The remaining two eigenvectors are tangent to the sphere, defined in a local planar coordinate system (Figure 3-1b).

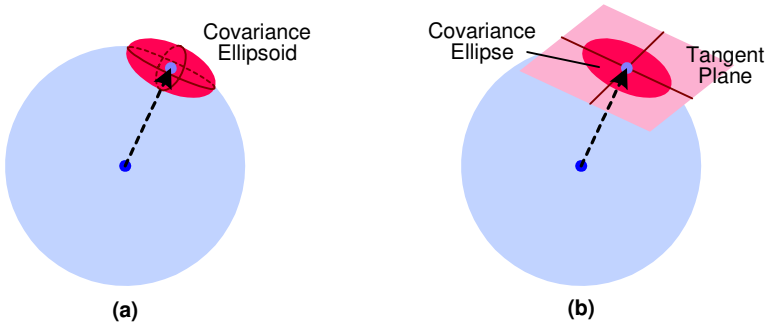


Figure 3-1: Approximations to Spherical Uncertainty
Additive Gaussian models are often used to describe noise in projective quantities, but are not defined on the appropriate parameter space. (a) A full 3-D model. (b) A tangent plane approximation.

Such additive models provide adequate descriptions of local behavior around the "mean" value. However, since they are effectively linearizations of spherical models, they become much less accurate further from the mean, especially as the covariance increases. Formulation of data fusion techniques is complicated by the fact that each measurement's uncertainty is defined in its own local coordinate system.

The statistics literature is rich with discussions of distributions on spheres. Examples include the Dimroth-Watson distribution [Wat83], which describes symmetric equatorial shapes, the Fisher distribution [Mar72, Ken82], which pertains only to directed (non-axial) quantities, and the angular central Gaussian distribution [Tyl87], which is invariant under any projective transformation but for which practical inference is all but impossible. Prentice [Pre84] has also shown that maximum likelihood estimates of spherical data can be obtained without explicit distributions.

Spherical uncertainty as applied to projective features, or "stochastic projective geometry", is treated extensively by Collins [CW90, Col93], who demonstrates that Bingham's distribution, described in the next section, is a suitable choice for projective inference problems that overcomes many shortcomings of other stochastic models.

3.3.2 Bingham's Distribution

Exponential distributions have many appealing properties [Ber79], not the least of which is their facility in inference tasks. Although the Gaussian density belongs to this family, it is a Euclidean probability measure and thus cannot be applied as an accurate uncertainty model for projective variables. However, if a zero-mean Gaussian density $p(\boldsymbol{x})$ (where $\boldsymbol{x} \in \mathbb{R}^3$) is conditioned on the event that $\|\boldsymbol{x}\| = 1$, the result is a flexible exponential density defined on the unit sphere.

This conditional density was first studied in depth by Bingham [Bin74], though it has been further analyzed (e.g. [JM79, Wat83]), and has also been extended to n dimensions. The distribution is parameterized by a symmetric $n \times n$ matrix $\boldsymbol{M}$, which can be diagonalized into the product $\boldsymbol{M} = \boldsymbol{U}\boldsymbol{\kappa}\boldsymbol{U}^\top$, where $\boldsymbol{U} \in \mathbb{R}^{n\times n}$ is a real unitary matrix whose columns $\boldsymbol{u}_i$ represent the principal directions of the distribution and $\boldsymbol{\kappa} \in \mathbb{R}^{n\times n}$ is a diagonal matrix of n concentration parameters κ_i. If $\mathcal{A}$ is the event that $\|\boldsymbol{x}\| = 1$, the density is given by

$$\begin{aligned} p(\boldsymbol{x}|\mathcal{A}) &= \frac{1}{c(\boldsymbol{\kappa})} \exp(\boldsymbol{x}^\top \boldsymbol{M} \boldsymbol{x}) \\ &= \frac{1}{c(\boldsymbol{\kappa})} \exp\left(\sum_{i=1}^{n} \kappa_i (\boldsymbol{u}_i^\top \boldsymbol{x})^2\right) \end{aligned} \tag{3-25}$$

where $c(\boldsymbol{\kappa})$ is a normalizing coefficient. This density will also be denoted $\mathcal{B}_n(\boldsymbol{x}; \boldsymbol{\kappa}, \boldsymbol{U})$, or simply $\mathcal{B}_n(\boldsymbol{x}; \boldsymbol{M})$, with the subscript n denoting the dimension of the space. The matrix $\boldsymbol{M}$ is analogous to the *information matrix* (inverse of the covariance) of a zero-mean Gaussian distribution [Riv84].

The Bingham density possesses a number of distinct characteristics. First, it is antipodally symmetric: the probability of any point $\boldsymbol{x}$ is identical to that of $-\boldsymbol{x}$. Second, it is closed under rotations. If $\boldsymbol{y} = \boldsymbol{R}\boldsymbol{x}$, where $\boldsymbol{R}$ is a rotation matrix and $\boldsymbol{x}$ has Bingham distribution $\mathcal{B}_n(\boldsymbol{x}; \boldsymbol{\kappa}, \boldsymbol{U})$, then $\boldsymbol{y}$ also has a Bingham distribution given by $\mathcal{B}_n(\boldsymbol{y}; \boldsymbol{\kappa}, \boldsymbol{R}\boldsymbol{U})$. Third, it is flexible, describing a wide variety of different density shapes (including uniform, bipolar, and equatorial) that depend only on the concentration parameters (Figure 3-2). Finally, the set of concentration parameters is unique only up to translation; in other words, the density is unchanged if a constant k is added to all parameters. By convention, the parameters (along with their corresponding modal directions $\boldsymbol{u}_i$) are ordered from smallest to largest, and shifted by an additive constant so that

$$\kappa_1 \leq \kappa_2 \leq \ldots \leq \kappa_n = 0.$$

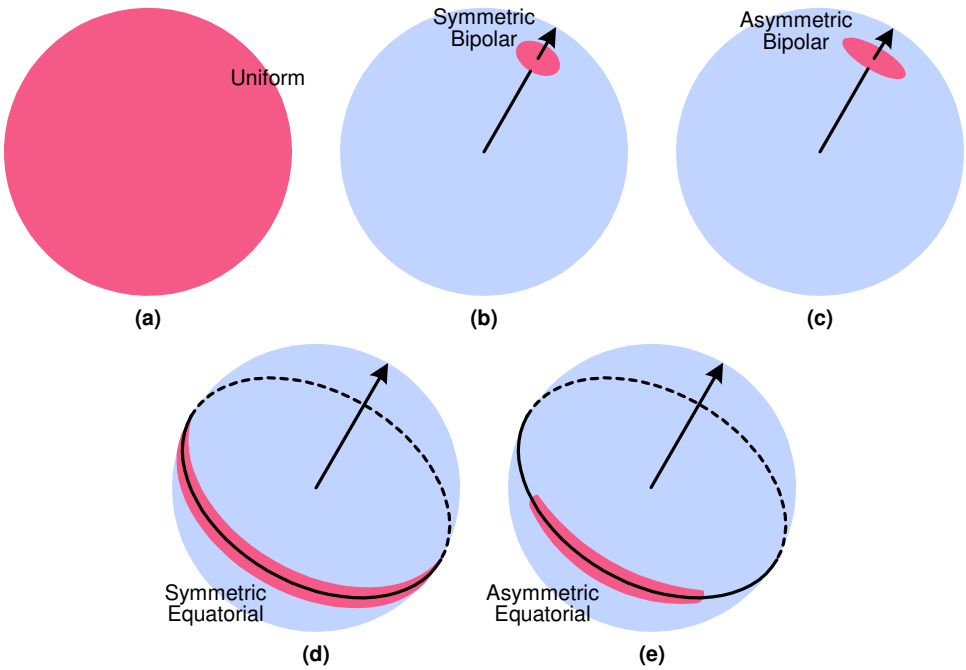


Figure 3-2: Bingham's Distribution on the Sphere
The shape of iso-density contours on Bingham's distribution depends on the concentration parameters. Examples are shown for $\mathcal{B}_3$. (a) A uniform distribution ($\kappa_1 = \kappa_2 = 0$). (b) A symmetric bipolar distribution ($\kappa_1 = \kappa_2 \ll 0$). (c) An asymmetric bipolar distribution ($\kappa_1 < \kappa_2 \ll 0$). (d) A symmetric equatorial distribution ($\kappa_1 \ll \kappa_2 = 0$). (e) An asymmetric equatorial distribution ($\kappa_1 \ll \kappa_2 < 0$).

Jupp and Mardia [JM79] have shown that the maximum likelihood estimates of Bingham parameters given a set of deterministic unit-length data points $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k\}$ is related to the sample second moment matrix

$$\boldsymbol{S}_{\boldsymbol{x}} = \frac{1}{k} \sum_{i=1}^{k} \boldsymbol{x}_i \boldsymbol{x}_i^\top. \tag{3-26}$$

If the matrix is diagonalized into $\boldsymbol{S}_{\boldsymbol{x}} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{-1}$, where $\boldsymbol{\Lambda}$ is a diagonal matrix of the eigenvalues, then $\boldsymbol{U} = \boldsymbol{V}$ (that is, the principal directions of the Bingham distribution are identically the eigenvectors of $\boldsymbol{S}_{\boldsymbol{x}}$), and the concentration matrix $\boldsymbol{\kappa}$ is a function of $\boldsymbol{\Lambda}$. It is thus possible to transform the Euclidean sample covariance into a spherical Bingham parameter matrix and vice versa. Note that the unit-length constraint on the data ensures that $\text{trace}(\boldsymbol{S}_{\boldsymbol{x}}) = 1$, i.e. that the eigenvalues sum to unity.

More information about the Bingham distribution can be found in Appendix B.

3.3.3 Feature Representations

The Bingham model $\mathcal{B}_3(\cdot)$ provides a natural representation for uncertainty in projective features. Points on the sphere can be treated as bipolar distributions (with $\kappa_1 \ll \kappa_2 \leq 0$), while lines

can be treated as equatorial distributions (with $\kappa_1 \leq \kappa_2 \ll 0$). The uniform distribution (with $\kappa_1 = \kappa_2 = 0$) describes completely uncertain entities.

Uncertainty is propagated through the feature extraction hierarchy described in §1.3, beginning with gradient pixels, each of which is modeled as a bipolar Bingham variable whose major axis passes through the pixel; determination of the variable's parameters is described in §3.5.1. Image lines are estimated by a separate technique from clusters of gradient pixels, and thus form equatorial distributions defined by the spatial extents of the clusters. §3.5.3 describes a general data fusion framework within which such equatorial distributions can be estimated. Line uncertainty also has a dual bipolar form, just as the line feature itself has a dual point representation on the sphere (§1.3.2). Point features in the image defined by line intersections are represented by bipolar distributions whose parameters depend on those of the constituent lines (§3.5.4).

3.4 Pose Uncertainty

One of the primary objectives of this work is to quantify uncertainty in estimated camera orientation and position. The following sections present the stochastic models used to describe pose uncertainty, and later chapters discuss its estimation.

3.4.1 Orientation

Three-dimensional orientation can be represented in many ways, such as (axis, angle) pairs, unitary matrices, and angular velocity. Because of the constrained and coupled nature of the parameters (e.g. orthonormality of rotation matrices), it is difficult to construct a simple, physically meaningful error model.

Some authors (e.g. [ZMI00]) treat orientation as a vector $\boldsymbol{\omega}$ whose magnitude $\|\boldsymbol{\omega}\|$ represents the angle of rotation and whose direction represents the axis. Additive Gaussian error models are used to quantify the uncertainty in the parameters; however, the Gaussian assumption is not valid because the sample space is finite, including only points inside the 3-D sphere of radius π. Singularity at the origin is also not addressed by this model.

Many formulations, including those presented in this thesis, represent rotations by unit quaternions, for which the appropriate sample space is the surface of the unit hypersphere in four dimensions (denoted by $\mathbb{S}^3$). Several authors therefore use 4×4 covariance matrices on quaternions [Pre86, Kan94, NS94] to describe uncertainty; this formulation is essentially a higher-dimensional analogue of the linearization used to represent directional uncertainty of projective features (§3.3.1) and has similar failings.

As seen in §3.3.2, the Bingham model is a sensible, flexible choice for describing uncertainty on hyperspherical surfaces. Therefore, rotational quantities are treated here as Bingham random variables parameterized by four concentration parameters $\boldsymbol{\kappa}$ (one of them zero by convention) and a 4×4 unitary matrix $\boldsymbol{U}$. This representation has also been used by Prentice for inference tasks concerning rotational quantities [Pre89].

3.4.2 Position

Random positions are defined on a Euclidean space $\mathbb{R}^3$, so are much simpler in form than are orientations. An ordinary trivariate Gaussian density $\mathcal{N}_3(\cdot; \boldsymbol{\mu}, \boldsymbol{\Lambda})$ can be used to describe absolute position or translational offsets. The mean $\boldsymbol{\mu}$ of the distribution represents the position itself, and the covariance matrix $\boldsymbol{\Lambda}$ represents its uncertainty.

3.5 Uncertainty of Derived Quantities

Image features and pose are often combined with or derived from other quantities in various ways—for example, cross products between two stochastic directions, or fusion of noisy measurements. This section describes the calculation of uncertainty in the resulting quantities.

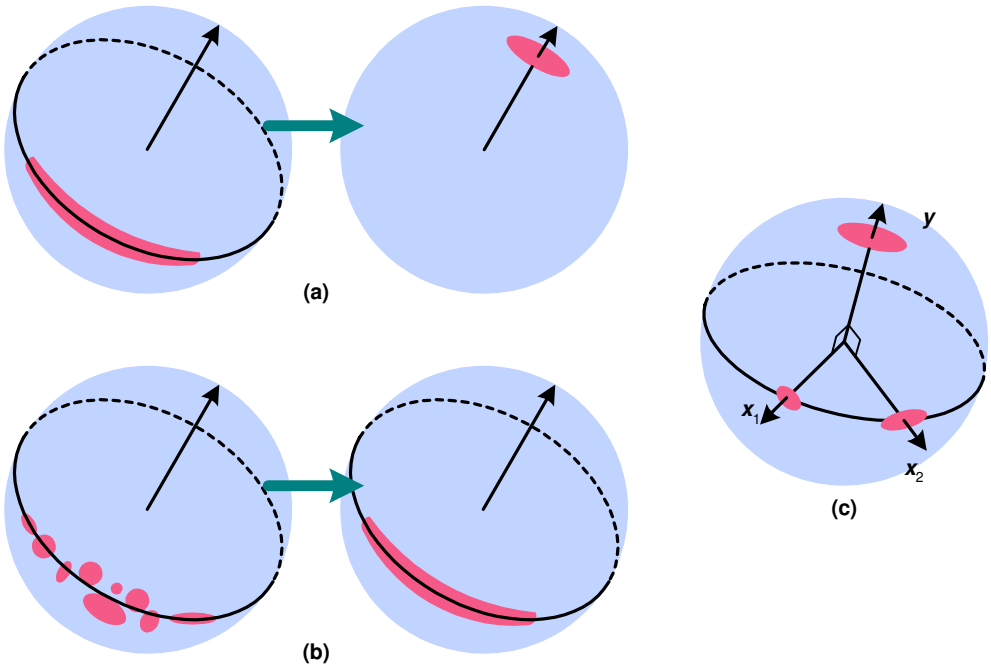


Figure 3-3: Projective Transformations
(a) The dual of an equatorial distribution is a bipolar distribution that preserves symmetry and asymmetry. (b) Many uncertain measurements can be fused into a single entity. (c) The cross product of stochastic directions can be formed by combining fusion and duality.

There exist general mechanisms for computing distributions of functions of random variables. For example, if $\boldsymbol{g}(\cdot)$ is an invertible function that maps $\mathbb{R}^n \rightarrow \mathbb{R}^n$ and $p_{\boldsymbol{x}}(\boldsymbol{x})$ is the known probability density of random variable $\boldsymbol{x}$, then the density of $\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{x})$ is given by

$$p(\boldsymbol{y}) = |\boldsymbol{J}_{\boldsymbol{g}^{-1}}(\boldsymbol{y})| p_{\boldsymbol{x}}(\boldsymbol{g}^{-1}(\boldsymbol{y})) \tag{3-27}$$

where $|\boldsymbol{J}_{\boldsymbol{g}^{-1}}(\boldsymbol{y})|$ is the determinant of the Jacobian of $\boldsymbol{g}^{-1}$ evaluated at $\boldsymbol{y}$. However, for more complicated mappings, such as the fusion of many noisy measurements into a single summarizing entity, density transformations are not so straightforward. The following sections describe computation of uncertainty in commonly-calculated quantities used throughout this work.

3.5.1 Gradient Pixels

As mentioned in §3.3.3, samples of gradient magnitude in the image can be represented as uncertain projective points. Since gradient pixels are used to estimate image lines and other quantities, it is important to quantify their reliability, which is assessed here by two qualitative characteristics. First is the gradient magnitude; sharp spatial changes in luminance signify drastic changes in depth or texture, either of which can provide strong structural cues. Second is pixel position; since the gradient is computed in planar rather than spherical images, pixels at different image positions provide different angular coverage of the sphere's surface. In particular, pixels close to the optical center subtend larger solid angles and are thus less reliable than pixels nearer to the image boundary.

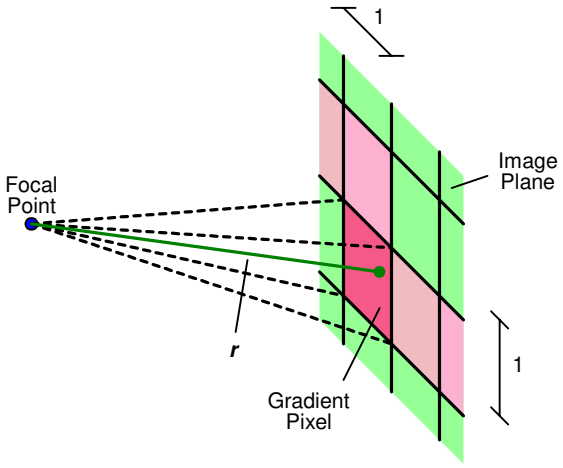


Figure 3-4: Gradient Pixel Uncertainty
The gradient in a square planar image pixel can be transformed into an uncertain quantity on the sphere by integration over all possible rays through the pixel.

The combination of these two characteristics can be used to obtain a set of Bingham parameters $\boldsymbol{M}$ for each gradient pixel at image position (u, v). Define w as the gradient magnitude $\|\nabla L(u, v)\| \in [0, 1]$, and define $\boldsymbol{r} = \boldsymbol{R}\boldsymbol{H}^{-1}\boldsymbol{p}$ where $\boldsymbol{p} = (u_0, v_0, 1)$, $\boldsymbol{R}$ is the camera's orientation matrix, and $\boldsymbol{H}$ is the projection matrix summarizing the camera's first-order internal parameters; this relation expresses the pixel as a ray in scene space. If the image pixel is assumed to be a unit square, then its second moment matrix can be calculated as:

$$\boldsymbol{S}_{\boldsymbol{x}} = \int_A \frac{\boldsymbol{r}\boldsymbol{r}^\top}{\|\boldsymbol{r}\|^2} \mathrm{d}u_0 \mathrm{d}v_0 \tag{3-28}$$

where A represents the square pixel region such that

$$-\frac{1}{2w} \leq (u_0, v_0) \leq \frac{1}{2w}.$$

Intuitively, each world ray $\boldsymbol{r}$ is normalized to unit length so that it lies on the unit sphere, and a scatter matrix is formed as the sum of squares of all such rays over the pixel. The pixel's effective

area is scaled down by the gradient magnitude, resulting in "tighter" distributions for more reliable pixels. The matrix can be used to compute the Bingham parameters, as described in §B.2.

Analytic evaluation of the integral in (3-28) is quite complicated, but under the reasonable assumption that the pixel subtends a very small angle on the sphere, linearization yields the following approximation:

$$\boldsymbol{S_x} \approx \frac{1}{\|\boldsymbol{r}\|^2}\boldsymbol{R}\boldsymbol{H}^{-1}\left(\boldsymbol{p}\boldsymbol{p}^\top + \frac{1}{12w^2}\mathrm{diag}(1,1,0)\right)(\boldsymbol{H}^{-1})^\top\boldsymbol{R}^\top. \tag{3-29}$$

This approximation does not immediately result in a valid scatter matrix because in general $\mathrm{trace}(\boldsymbol{S_x}) \neq 1$; thus, the resulting matrix is normalized by its trace.

3.5.2 Duality of Projective Uncertainty

It was shown in §1.3.2 that a duality exists between projective features; for example, a projective line spans a great circle on the unit sphere, but can also be represented as a projective point on the sphere normal to this great circle, and vice-versa. Uncertain line segments can be treated as equatorial Bingham distributions, but sometimes it is desirable to obtain a distribution on their dual point representation.

There is certainly a clear distinction between deterministic line features and their dual points, but the stochastic case is not so straightforward. The flexibility of the Bingham density allows for a continuum of distributional representations; for example, a very uncertain point feature may have a nearly equatorial distribution, or a very short line segment may have a nearly bipolar distribution. Therefore, the notion of a dual distribution is not a well-defined entity and is difficult to quantitatively describe.

Several qualitative statements, however, can be made concerning desirable properties of dual uncertainty. The dual distribution should be formulated such that deterministic duality of points and lines is a limiting case as uncertainty tends to zero. In addition, symmetry and asymmetry should be preserved in the expected way; for example, a symmetric bipolar variable should becomes a symmetric equatorial variable, and vice-versa (Figure 3-3a).

A simple dual distribution for a Bingham random variable $\mathcal{B}_3(\cdot;\boldsymbol{M})$ is proposed as $\mathcal{B}_3(\cdot;-\boldsymbol{M})$. This operation does not affect the modal directions encoded in $\boldsymbol{U}$, but because of the ordering and translation conventions in §3.3.2 it changes the concentration parameters κ_i and re-orders the $\boldsymbol{u}_i$. In particular,

$$\begin{array}{rccccc} & \kappa_1 & \leq & \kappa_2 & \leq & 0 \\ \rightarrow & -\kappa_1 & \geq & -\kappa_2 & \geq & 0 \\ \rightarrow & 0 & \leq & -\kappa_2 & \leq & -\kappa_1 \\ \rightarrow & \kappa_1 & \leq & \kappa_1 - \kappa_2 & \leq & 0 \\ \rightarrow & \tilde{\kappa}_1 & \leq & \tilde{\kappa}_2 & \leq & 0. \end{array}$$

The new concentration parameters $\tilde{\kappa}_i$ are obtained by negating, reversing, and shifting the old parameters, and the new orientation matrix $\tilde{\boldsymbol{U}}$ is a permutation of the old matrix. The resulting distribution possesses the desired qualitative properties mentioned above.

3.5.3 Projective Data Fusion

Data fusion is an important inference task in which a large number of uncertain measurements are summarized by a small number of representative quantities that are presumably more reliable (Figure 3-3b). This section describes methods for Bayesian inference from Bingham distributed observations.

It should first be noted that two of the assumptions in this thesis—namely hemispherical imagery and known internal camera calibration—imply that perfect measurements of projective rays are available. In reality, however, small unmodeled calibration errors perturb ray measurements slightly from their true directions. The noise in every measured or fused quantity can thus be decomposed into two components, one representing the initially unknown aggregation of unmodeled processes, and the other quantified by previous knowledge or inference. The task of data fusion, then, is to recover estimates of the aggregate unknown uncertainty, in the form of a Bingham parameter matrix $\boldsymbol{M}$, based on observations with known uncertainty.

As with previously described inference techniques, let $\mathcal{X}$ represent a set of samples $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_k\}$. In the deterministic case, i.e. when there is no measurement noise, a ML estimate of $\boldsymbol{M}$ can be obtained directly from the sample second moment matrix $\boldsymbol{S}_{\boldsymbol{x}}$, as discussed in §3.3.2. If uncertainty in the data is known only in the form of scalar weights w_i, then a weighted scatter matrix can be computed by

$$\boldsymbol{S}_{\boldsymbol{x}} = \frac{\sum_{i=1}^{k} w_i \boldsymbol{x}_i \boldsymbol{x}_i^\top}{\sum_{i=1}^{k} w_i} \tag{3-30}$$

and used to estimate $\boldsymbol{M}$.

A more general description of uncertainty assumes each measurement $\boldsymbol{x}_i$ to be corrupted by a Bingham process with parameter matrix $\boldsymbol{M}_i$. Unlike traditional errors-in-variables approaches (e.g. [MM00]), the underlying noise model is neither additive nor Gaussian; however, since Bingham's distribution is exponential, analogous errors-in-variables methods can be formulated on the sphere [Col93].

Let $\boldsymbol{\Theta}$ represent the random variable resulting from the fusion of all data in $\mathcal{X}$. Using Bayesian arguments from §3.2.2, the posterior density can be written as

$$p(\boldsymbol{\Theta}|\mathcal{X}) = \frac{1}{c} p(\boldsymbol{\Theta}) \prod_{i=1}^{k} p(\boldsymbol{x}_i|\boldsymbol{\Theta}) \tag{3-31}$$

where c is the normalizing marginal probability $p(\mathcal{X})$, which does not depend on $\boldsymbol{\Theta}$. The prior density is assumed to be of Bingham form, with parameter matrix $\boldsymbol{M}_0$. The problem is now to determine the form of the likelihood $p(\boldsymbol{x}_i|\boldsymbol{\Theta})$ given that each measurement is itself an uncertain sample from a Bingham distribution.

Let $\boldsymbol{M}_i(\boldsymbol{\Theta})$ represent the Bingham uncertainty in $\boldsymbol{x}_i$ as expressed from the reference frame of the variable $\boldsymbol{\Theta}$. The conditional likelihood is then given by

$$p(\boldsymbol{x}_i|\boldsymbol{\Theta}) = \frac{1}{c(\boldsymbol{\kappa_i})} \exp(\boldsymbol{x}_i^\top \boldsymbol{M}_i(\boldsymbol{\Theta}) \boldsymbol{x}_i). \tag{3-32}$$

The transformation required to express $\boldsymbol{M}_i$ as $\boldsymbol{M}_i(\boldsymbol{\Theta})$ is a rotation $\boldsymbol{R}(\boldsymbol{\Theta} \rightarrow \boldsymbol{x}_i)$ that takes $\boldsymbol{\Theta}$ to $\boldsymbol{x}_i$. Thus, the likelihood can be rewritten as

$$\begin{aligned} p(\boldsymbol{x}_i|\boldsymbol{\Theta}) &= \mathcal{B}_3(\boldsymbol{x}_i; \boldsymbol{M}_i(\boldsymbol{\Theta})) \\ &= \mathcal{B}_3(\boldsymbol{x}_i; \boldsymbol{R}(\boldsymbol{\Theta} \rightarrow \boldsymbol{x}_i)\boldsymbol{M}_i\boldsymbol{R}^\top(\boldsymbol{\Theta} \rightarrow \boldsymbol{x}_i)) \\ &= \frac{1}{c(\boldsymbol{\kappa_i})} \exp(\boldsymbol{x}_i^\top \boldsymbol{R}(\boldsymbol{\Theta} \rightarrow \boldsymbol{x}_i)\boldsymbol{M}_i\boldsymbol{R}^\top(\boldsymbol{\Theta} \rightarrow \boldsymbol{x}_i)\boldsymbol{x}_i) \\ &= \mathcal{B}_3(\boldsymbol{R}(\boldsymbol{\Theta} \rightarrow \boldsymbol{x}_i)\boldsymbol{x}_i; \boldsymbol{M}_i) \\ &= \mathcal{B}_3(\boldsymbol{\Theta}; \boldsymbol{M}_i) \\ &= \frac{1}{c(\boldsymbol{\kappa}_i)} \exp(\boldsymbol{\Theta}^\top \boldsymbol{M}_i \boldsymbol{\Theta}). \end{aligned} \tag{3-33}$$

Finally, substitution of (3-33) into the posterior density in (3-31) gives

$$\begin{aligned} p(\boldsymbol{\Theta}|\mathcal{X}) &= c \exp\left(\boldsymbol{\Theta}^\top \left[\sum_{i=0}^{k} \boldsymbol{M}_i\right] \boldsymbol{\Theta}\right) \\ &= c \exp(\boldsymbol{\Theta}^\top \boldsymbol{M}_{\boldsymbol{\Theta}} \boldsymbol{\Theta}) \end{aligned} \tag{3-34}$$

where c is a normalizing coefficient. This density has Bingham form $\mathcal{B}_3(\boldsymbol{\Theta}; \boldsymbol{M}_{\boldsymbol{\Theta}})$ and is easily calculated from the measurement uncertainty and prior distribution.

Several intuitive insights can be drawn concerning (3-34). First, if nothing about $\boldsymbol{\Theta}$ is initially known, then the prior density $\boldsymbol{M}_0$ can be chosen as a uniform density (i.e. $\boldsymbol{M}_0 = \boldsymbol{0}$, containing no information). Second, any completely uncertain measurement contributes nothing to the posterior distribution. Finally, the dual distribution of $p(\boldsymbol{\Theta}|\mathcal{X})$ can be obtained simply by negating the parameter matrix $\boldsymbol{M}_{\boldsymbol{\Theta}}$, which is equivalent to negating each measurement matrix $\boldsymbol{M}_i$. This is consistent with the notion of projective duality; for example, if $\boldsymbol{\Theta}$ represents a projective line obtained by the fusion of projective points $\boldsymbol{x}_i$, then the dual of this line is a point obtained by the fusion (i.e. intersection) of lines.

The above techniques are used to determine equatorial distributions on line features estimated from collinear gradient pixels, which themselves are random quantities (§3.5.1). They are also used in the estimation of vanishing points and motion directions, described in later chapters.

3.5.4 Cross Products

The cross product $\boldsymbol{y} = \boldsymbol{x}_1 \times \boldsymbol{x}_2$ between a pair of unit vectors is a common operation that determines the vector $\boldsymbol{y}$ orthogonal to both $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. Examples include the cross product between a pair of projective points, which represents the line through both points, and between a pair of projective lines, which represents their point of intersection. The reliability of the result depends on that of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, and also on the angle between them. Qualitatively, the magnitude or certainty of $\boldsymbol{y}$ is maximized when the two vectors are orthogonal, and minimized when the two vectors are parallel or anti-parallel.

The cross product of two stochastic projective points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ can thus be viewed as the dual of an equatorial distribution formed by the fusion of these two points (Figure 3-3c), and

the methods of the previous section can be applied. The Bingham parameter matrix of the cross product is simply

$$\boldsymbol{M_y} = -(\boldsymbol{M}_1 + \boldsymbol{M}_2) \tag{3-35}$$

where $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ are the parameter matrices of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, respectively. In the case of line intersections, the parameter matrix is given similarly by

$$\boldsymbol{M_y} = \boldsymbol{M}_1 + \boldsymbol{M}_2 \tag{3-36}$$

where here $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are equatorial random variables representing projective lines.

CHAPTER 4

Basic Algorithms

THE SYSTEM developed as part of this thesis employs many commonly-used algorithms. This chapter reviews fundamentals and derivations of the most important of these algorithms, and presents novel modifications tailored to specific system tasks. §4.1 describes the Hough transform quite generally as a powerful discretized voting mechanism, used here for rapid, robust initialization of more accurate continuous techniques. The expectation maximization (EM) algorithm, presented in §4.2, simultaneously classifies data and estimates parameters, and is applied to several inference tasks such as vanishing point estimation and rotational registration. In §4.3, the Markov chain Monte Carlo (MCMC) algorithm is discussed as an efficient means of sampling from probability distributions in high-dimensional spaces via stochastic perturbations. Finally, §4.4 describes a method used to determine adjacency between cameras given approximately known external pose. Each section outlines the problem domain, sketches a brief derivation if applicable, describes the mechanics of the algorithm, and mentions applications of the algorithm to this work.

4.1 Hough Transform

The Hough transform was originally formulated as a robust mechanism for detection of straight lines, circles, and ellipses in images [Hou62]. It has since seen many other uses in vision and image processing applications, such as the detection of vanishing points [Bar83], but has remained an image-to-parameter space transformation [KTT99].

This section presents the Hough transform more generally as a robust parameter estimation tool for largely overdetermined systems of equations (i.e. for objective minimization in a "primal" formulation). The transform can also be interpreted geometrically as a practical method for locating intersections of many hypersurfaces (i.e. for objective maximization in a "dual" formulation). Real implementations of the transform inherently limit its applicability and can prevent attainment of highly accurate solutions; however, all techniques in this work which utilize the transform do so

only to initialize other more accurate techniques that require reasonable initialization to find global optima. In stochastic terms, the Hough transform produces strong prior distributions on parameter estimates.

4.1.1 Definition

Consider the set of equations

$$f_i(\boldsymbol{x}_i, \boldsymbol{\Theta}) = 0 \qquad \forall i \tag{4-1}$$

$$g_j(\boldsymbol{\Theta}) = 0 \qquad \forall j \tag{4-2}$$

where $f_i(\cdot)$ and $g_j(\cdot)$ are known scalar-valued functions. These equations define an explicit relationship between observable quantities $\boldsymbol{x}_i \in \mathbb{R}^n$ and unknown parameters $\boldsymbol{\Theta} \in \mathbb{R}^m$. For a given value of $\boldsymbol{x}_i$, there exists a possibly empty set $\mathcal{S}_i$ of values of $\boldsymbol{\Theta}$ that satisfy (4-1). If $m \geq n$, then the solution set $\mathcal{S}_i$ forms a hypersurface of dimension $m - n$ in the parameter space $\mathbb{R}^m$. The formation of hypersurfaces for all i defines the *Hough transform* of the data $\mathcal{X}$, and also demonstrates the principle of duality. The additional constraints $g_j(\cdot)$ effectively restrict the solution space to a possibly nonlinear subspace of $\mathbb{R}^m$.

To illustrate more concretely, take the commonly-used example of line detection in planar images. As mentioned in §1.3.1, a straight line can be parameterized by an angle and an offset according to (1-9). The parameters to be estimated are thus $\boldsymbol{\Theta} = (\theta, c) \in \mathbb{R}^2$, and the parameter space can be restricted by

$$g(\boldsymbol{\Theta}) = \begin{cases} 0, & 0 \leq \theta \leq \pi, \quad -c_0 \leq c \leq c_0 \\ 1, & \text{otherwise} \end{cases} \tag{4-3}$$

where c_0 is related to the image dimensions (since the line must lie within the bounds of the image). Each observation $\boldsymbol{x}_i \in \mathbb{R}^2$ consists of the image coordinates (u_i, v_i) of a single pixel; the data-dependent constraints are then given by

$$f(\boldsymbol{x}_i, \boldsymbol{\Theta}) = u_i \sin\theta + v_i \cos\theta + c. \tag{4-4}$$

Thus, constraint "surfaces" are sinusoidal contours in the (θ, c) plane, shown in Figure 4-1.

4.1.2 Properties

If perfect measurements of the data are available, the common intersection of all hypersurfaces $\mathcal{S}_i$ defines the overall solution set, i.e. the values of $\boldsymbol{\Theta}$ that simultaneously satisfy all constraints. Since the analytic intersection of many geometric entities in a potentially high dimensional space is not a feasible task, it is convenient to define a non-negative *incidence function* $h(\boldsymbol{\Theta})$ for values of $\boldsymbol{\Theta}$ satisfying (4-2). The incidence function evaluated at a given point measures the number of hypersurfaces passing through that point; thus, solutions to (4-1) and (4-2) are given by $\operatorname{argmax}_{\boldsymbol{\Theta}}[h(\boldsymbol{\Theta})]$, the values of $\boldsymbol{\Theta}$ that maximize $h(\cdot)$; the parameter space thus becomes an *accumulation space*.

Hypersurfaces representing unstructured random outlier data intersect the true data surfaces and each other at random points and are therefore not self-consistent. As a result, they do not significantly impact the incidence function, and, in the absence of measurement noise, maximum values of $h(\cdot)$ thus remain invariant with respect to random outlier data.

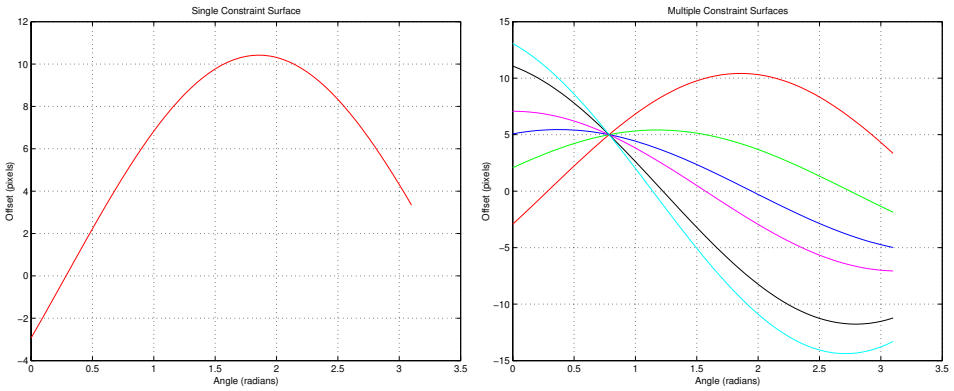


Figure 4-1: Contours for Line Detection
A single constraint on a 2-D image line is shown at left. When superimposed, multiple constraints all intersect at a common point in the parameter space, shown at right. This point represents the optimal solution—that is, the parameter values that satisfy all constraints.

4.1.3 Implementation

The above seems an elegant formulation for robust solution of arbitrary equation systems. However, three problems immediately arise in practice. First, implementation of a continuous accumulation space is impossible because it requires an infinite amount of storage. Second, $h(\cdot)$ is not an analytic function, so the task of finding its maxima is impossible because brute-force search of the parameter space requires infinite computation time. Third, data is invariably corrupted by noise, which disperses the intersections and thus obscures the true maxima of $h(\cdot)$.

All three problems can be addressed by sampling $h(\cdot)$ at discrete points in the parameter space. In practice, the accumulation space consists of a set of spatially-indexed cells, each of which contains a non-negative accumulation value and represents a small, contiguous spatial range of parameters. A simple algorithm for approximating $h(\cdot)$ is to initialize all cells to zero, and then to intersect them with the dual surface $\mathcal{S}_i$ of each data point $\boldsymbol{x}_i$, incrementing the value at each cell through which the surface passes.

Because the number of cells is finite, this implementation requires finite memory and computation time. Dispersion of surface intersections is partially ameliorated by the cells' finite size, which tends to "blur" proximal intersections somewhat.

4.1.4 Further Difficulties

Discretization overcomes several of the initially apparent difficulties with Hough transform approaches. However, the literature notes several other fundamental concerns, described below.

One difficulty is parameterization. Appropriate choice of the parameter space is crucial for Hough transform applications and in fact for any estimation technique. There are often several representations for quantities of interest. Singularities and high-dimensional parameter spaces can prevent efficient discretization, and highly nonlinear parameterizations complicate spatial indexing and notions of cell adjacency.

Reliably detecting maxima or *peaks* in the transform is also a difficult task. Intersections become dispersed, in part by the discretization and in part by noisy data, and as a result otherwise

sharp peaks spread over larger regions and decrease in magnitude. In addition, multi-modal data produces multiple peaks, in which case all "high" incidence regions in the parameter space must be found rather than a single point of maximum incidence. If the number of peaks is not known in advance, then discrimination between weakly-defined true peaks and strong outlier peaks becomes difficult.

There are also inherent limitations on the accuracy of Hough transform techniques [GH90]. Peaks can be detected only to within a given cell boundary, so accuracy is limited by cell size. In addition, discretization can split peaks over several cells, confusing their true positions. Sophisticated approaches such as curve fitting [NP88], hypothesis testing [PIK94], and multi-resolution transforms [Ati92] attempt to improve accuracy, but still cannot completely overcome the inherent limitations.

Finally, representation of uncertainty in the data and in recovered peaks is problematic and not treated in most Hough transform implementations, which intersect ideal, infinitesimally thin surfaces with the set of accumulation cells. Anti-aliasing methods [FvDFH90] in the space of cells can ameliorate the problem, but are somewhat ad-hoc and do not directly represent the data's uncertainty. Some authors approximate uniform uncertainty by "thick" surfaces that intersect a larger portion of the accumulation space [Shu99], but implementation for complex surfaces presents a significant challenge.

Because of its inherently imprecise nature and other limitations, the Hough transform is not the ideal tool for accurate parameter estimation tasks. Its appeal lies in its robustness and the simplicity and speed with which approximate solutions can be obtained. In this thesis, therefore, the Hough transform is used only to initialize other more accurate parameter estimation methods. Combining the robustness and efficiency of discretized voting methods with the precision of continuous-space inference results in a powerful parameter estimation framework.

This framework considerably simplifies practical implementation of the transform. Data uncertainty need not be treated in detail, and peak detection methods need not be extremely precise, since these issues are addressed by subsequent continuous parameter estimation. All formulations in this thesis also reside in low-dimensional spaces, alleviating memory and computational issues and allowing the use of simple rectangular-grid cell structures. Simple, efficient implementations can thus be used as long as the parameterization is well-conditioned and peaks can be detected to reasonable accuracy; the following sections discuss efficient parameterization and peak detection methods.

4.1.5 Parameterization of the Sphere

All Hough transform applications in this work are formulated in the projective plane $\mathbb{P}^2$. As mentioned in §1.3.2, $\mathbb{P}^2$ can be mapped to the surface of the unit sphere $\mathbb{S}^2$, which in the notation of (4-2) is equivalent to imposing the quadratic constraint function

$$g(\boldsymbol{\Theta}) = \boldsymbol{\Theta}^\top \boldsymbol{\Theta} - 1. \tag{4-5}$$

Parameterization and uniform discretization of the sphere's surface are generally difficult. Tesselations of the sphere are sometimes used [KN95], but the data structures required to account for cell adjacency can be complex. The Euclidean plane tangent to the sphere serves as a good local approximation [Kan92], but is ill-conditioned near points at infinity (i.e. projective points parallel to the plane).

Nevertheless, planar Euclidean surfaces have the advantage of simplicity: discretization is straightforward, cell adjacency is naturally indexed, and accumulation of linear projective features reduces to simple line clipping and drawing. Thus, if parameters are constrained to lie on a small known portion of the sphere's surface, then a finite, regularly-sampled plane tangent to the sphere (Figure 4-2a) provides a good local approximation without singularities.

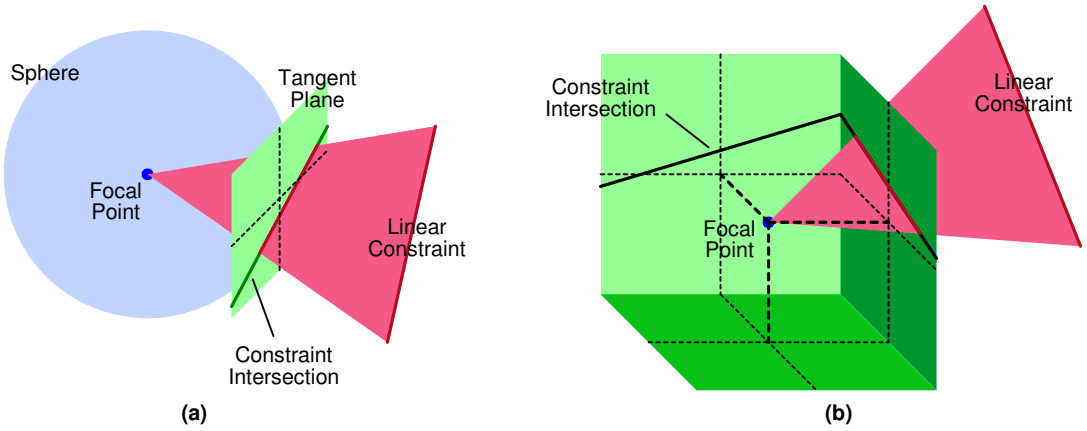


Figure 4-2: Discretization of the Sphere
Linear approximations can facilitate efficient sampling of spherical surfaces. (a) A tangent plane covers a small portion of the surface. (b) A cube covers the entire surface.

If such constraint on the parameters cannot be imposed, then the entire spherical surface must be parameterized (Figure 4-2b). §1.2.2 introduced the unit cube, a compromise between planar surfaces, which are simplest to implement, and spherical surfaces, which represent the ideal parameter space. Because projective quantities are axial, only half of the sphere's surface is required to describe all possible ray directions; thus only three cube faces are used rather than all six [TPG97]. The cube contains no singularities, spans the entire space, and provides a reasonable, efficient approximation to uniform sampling. It should again be emphasized that perfect sampling is not necessary; planar approximations suffice because the Hough transform is used only for initialization of other techniques.

4.1.6 Proximity Function

Most Hough transform implementations *scan convert* various primitives representing data hypersurfaces into the cells of the accumulation space. Essentially, this involves computing a geometric intersection per hypersurface; if primitives are easily parameterized (in the case of straight lines, for example), this process is quick and efficient, but remains infeasible for non-parametric shapes. The method also fails to incorporate data uncertainty, treating each $\mathcal{S}_i$ as an ideal, infinitesimally thin surface.

One alternative is to compute accumulation values *per cell* rather than *per hypersurface*, in effect reversing the order of loops in the transform. Each cell is visited only once, and the accumulation value is incremented for each $\mathcal{S}_i$ passing through that cell. Because the accumulation space is discrete, however, nearly all hypersurfaces will "miss" all cells; a cell's quantized spatial position will most likely not precisely satisfy any given constraint.

A novel solution that addresses this problem and also allows incorporation of data uncertainty is the introduction of a *proximity function* $\rho(\boldsymbol{x}_i, \boldsymbol{\Theta})$. Such a function tests any point $\boldsymbol{\Theta}$ in the parameter space and returns a nonnegative value indicating the point's proximity to a given hypersurface. Ideally $\rho(\cdot)$ produces very low values for points far from $\mathcal{S}_i$, and higher values (e.g. 1) for points lying precisely on $\mathcal{S}_i$.

Proximity functions can be implemented in several ways. Perhaps the simplest is to utilize the constraint equation (4-1), whose value is precisely zero when $\boldsymbol{\Theta}$ precisely meets the constraint. For example,

$$\rho(\boldsymbol{x}_i, \boldsymbol{\Theta}) = \begin{cases} 1 - \frac{1}{r}|f(\boldsymbol{x}_i, \boldsymbol{\Theta})|, & |f(\boldsymbol{x}_i, \boldsymbol{\Theta})| \leq r \\ 0, & \text{otherwise} \end{cases} \tag{4-6}$$

where r represents a tolerance value, or maximum "radius" from the hypersurface beyond which the proximity is considered zero. The expression in (4-6) is a linear mapping of constraint error to proximity; more complicated mappings (e.g. quadratic) can also be applied. In fact, the constraint error can be mapped onto a Gaussian distribution in $\mathbb{R}$ to model data uncertainty:

$$\rho(\boldsymbol{x}_i, \boldsymbol{\Theta}) = \mathcal{N}_1(f(\boldsymbol{x}_i, \boldsymbol{\Theta}); 0, \sigma_i^2), \tag{4-7}$$

where σ_i^2 is a scalar value representing the degree of uncertainty in $\boldsymbol{x}_i$. This formulation also allows for non-uniform weighting of observations in the transform.

Other proximity functions can incorporate the geometric distance of a point in the parameter space from a given constraint $\mathcal{S}_i$, if such distances are measurable (for example, if the normal vector to the hypersurface can be analytically evaluated at every point). Generally, however, (4-7) provides a simple and effective means for accumulating complex surfaces, smoothing the data, and incorporating uncertainty. This method, although more computationally expensive than the scan conversion approach, produces more accurate results.

4.1.7 Detecting Peaks

In typical applications of the Hough transform, maxima of $h(\cdot)$ are discrete points in the parameter space. These peaks are not easily detected, however, as they are often corrupted by data noise, outliers, and discretization artifacts. In addition, as mentioned in §4.1.4, it is often difficult to distinguish between true solutions and peaks that occur by random chance. This section presents a method for detecting most true peaks while rejecting most false peaks.

It is assumed here that the accumulation space is a two-dimensional planar surface parameterized by ordered pairs (u, v). Because of various corruptive influences, peaks are regions of "high" incidence, not necessarily maximum incidence. By its mathematical definition, a peak at (u_0, v_0) is a local maximum in the parameter space, occurring when the following conditions are satisfied:

$$\begin{aligned} \|\nabla h(u_0, v_0)\| &= 0 \\ \frac{\partial^2 h(u,v)}{\partial u^2}\frac{\partial^2 h(u,v)}{\partial v^2}\bigg|_{(u_0,v_0)} &> \left[\frac{\partial^2 h(u,v)}{\partial u \partial v}\bigg|_{(u_0,v_0)}\right]^2 \\ \frac{\partial^2 h(u,v)}{\partial u^2}\bigg|_{(u_0,v_0)} &< 0 \end{aligned}$$

These conditions can be inconclusive, however, requiring still higher-order derivatives in cases where equality rather than strict inequality is attained.

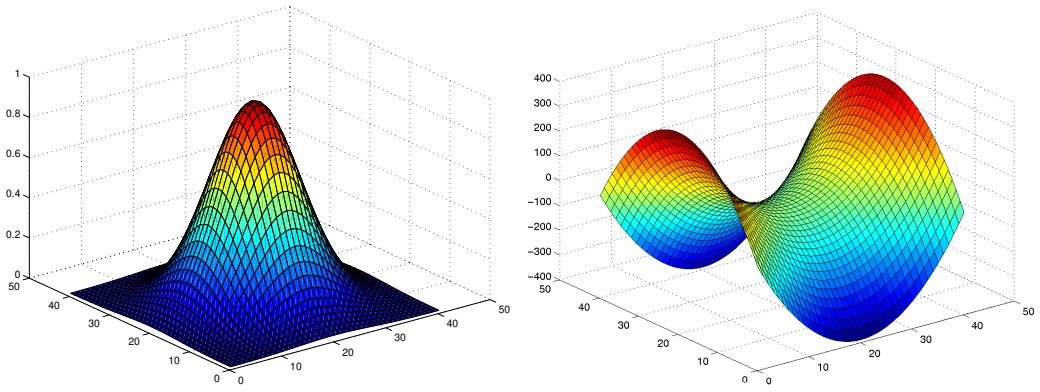


Figure 4-3: Local Maxima
Local maxima of a parameterized surface (shown at left) occur where the gradient is zero. Zero gradient, however, can also correspond to local minima, inflection points, or saddle points (shown at right).

A more qualitative definition states that a peak is any point whose magnitude is larger than those of all other points in a small neighborhood or *window* W:

$$h(u_0, v_0) \geq h(u, v) \quad \forall (u, v) \in W. \tag{4-8}$$

If determination of adjacency between cells is straightforward, as is the case in regularly-sampled planar parameter spaces, then each cell (u_0, v_0) can be tested by examining all neighbors in W. Any cell that passes the criterion in (4-8) is considered a local maximum. In practice, a square window around each cell is used whose size depends on the granularity of the discretization; the size is chosen so that W's angular coverage is approximately independent of cell size. A lowpass filter is convolved with the cells before peak detection to smooth the accumulation surface by attenuating high-frequency noise artifacts that give rise to false maxima. This is analogous to the smoothing of the gradient or Laplacian image before thresholding in edge detection algorithms [Can86].

Determination of cell adjacency within a single cube face is straightforward, but adjacency across cube faces requires complicated special cases at the boundaries. A simple scheme is used to approximate inter-face adjacency: after all data surfaces have been accumulated, each face is padded with a border whose width is equal to half of the window size. Cells from neighboring faces are used to fill this padded region, as shown in Figure 4-4, and all computations can then be performed on each face independently without the need for special adjacency cases.

The peak detection method outlined above generally produces hundreds of peaks, most of which are low-magnitude artifacts caused by outlier data. Thus the final step is to extract the few "significant" peaks, and discard the remainder. The magnitude p_m of a given peak is computed as the sum of all cell values in the window W around the peak:

$$p_m = \sum_{(u,v) \in W} h(u, v). \tag{4-9}$$

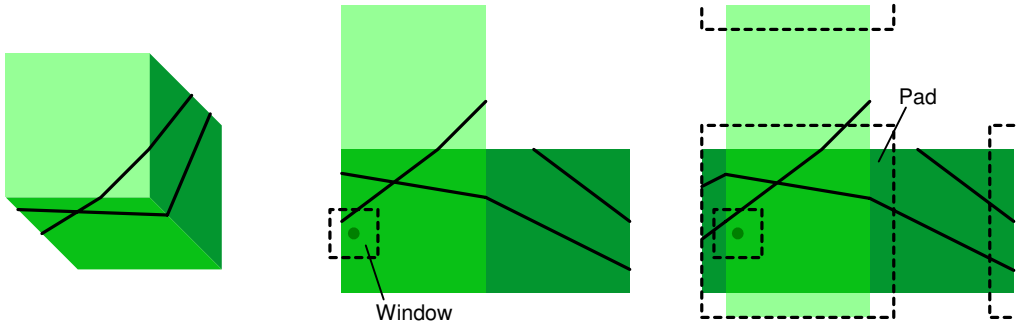


Figure 4-4: Spatial Adjacency in Cubic Discretization
To handle boundary conditions in which the search window extends beyond the edge of a given face, all three faces are "unfolded" and each is padded by its neighbor's values, which are wrapped according to antipodal symmetry.

Peaks generally have varying degrees of spread, caused by noisy data and discretization. The concentration p_c of a given peak can be approximated as the ratio of average accumulation values in W including and excluding the peak itself:

$$\begin{aligned} p_c &= 1 - \left(\frac{p_m - h(u_0, v_0)}{a - 1} \right) \Big/ \left(\frac{p_m}{a} \right) \\ &= \frac{ah(u_0, v_0) - p_m}{(a-1)p_m}, \end{aligned} \tag{4-10}$$

where a is the area of W. The overall measure of peak strength p_s is computed as the product of the magnitude and concentration,

$$p_s = p_m p_c = \frac{ah(u_0, v_0) - p_m}{a - 1}. \tag{4-11}$$

Sharp peaks with relatively low magnitude are thus given as much significance as smoother peaks with high magnitude.

Finally, the sample variance σ_p^2 is computed on the strength measure of all candidate peaks. Only those peaks are kept for which $p_s > \alpha\sigma_p$, where α is a threshold value. In practice a value of $\alpha = 4$ is used, corresponding to approximately 95% confidence in zero-mean normally distributed data. Performance of this peak detection method is evaluated in §7.1.

4.1.8 Peak Uncertainty

Aside from initializing more accurate techniques, Hough transform peaks can provide strong prior densities on random variables of interest. A peak on $\mathbb{S}^2$ can be expressed as a bipolar Bingham density (with $\kappa_1 \leq \kappa_2 \ll 0$) whose modal axis is aligned with the peak's direction. The parameter matrix $\boldsymbol{M}$ is approximated from a local neighborhood of accumulation values by using a weighted

scatter matrix, as in (3-30):

$$\boldsymbol{S} = \frac{\sum_{(u,v)\in W} h(u,v)\boldsymbol{r}(u,v)\boldsymbol{r}(u,v)^\top}{\sum_{(u,v)\in W} h(u,v)} \tag{4-12}$$

where $\boldsymbol{r}(u,v)$ is the unit vector from the optical center through the point. The matrix $\boldsymbol{M}$ is then computed by mapping $\boldsymbol{S}$ into Bingham parameters (§B.2).

4.2 Expectation Maximization

Real data often represents a mixture of contributions from several different generating processes. For example, samples of interest drawn from a single probability distribution can be contaminated by outlier samples, which by definition are drawn from fundamentally different distributions. Parameter estimation thus becomes complicated by the fact that data is unclassified; that is, not only are the generating distributions' parameters unknown, but so are the assignments of data points to these distributions.

There is an inherent coupling between classification and parameters estimation. A correct assignment of observations to distributions effectively partitions the data set so that estimation can be performed independently on each partition. Conversely, knowledge of the parameters allows each observation to be classified according to which known distribution is most likely to have produced it.

Algorithms such as k-means [MD89] attempt to assign a *hard* or explicit classification to each data point in an iterative procedure. Each observation $\boldsymbol{x}_i$ is explicitly assigned to the distribution j that produces the largest likelihood value $p(\boldsymbol{x}_i|\boldsymbol{\Theta})$ according to the current parameter estimates $\boldsymbol{\Theta}$. New parameters are then estimated for each partitioned subset, and the process repeats. Implementation of these algorithms is straightforward, but in practice their performance suffers because of data that straddles the boundary between distributions [Bis95]. The algorithms explicitly assign such data to one or the other of the distributions, when the data should actually be included in *both* distributions. Random outliers also tend to adversely affect hard classification algorithms.

The *expectation maximization* (EM) algorithm, first proposed by Demster et al [DLR77], can be viewed as an extension of the k-means algorithm that classifies data probabilistically, producing *soft* rather than hard assignments. It is a powerful tool for parameter and density estimation, and thus has recently become popular in vision research for solving a wide variety of different problems.

4.2.1 Mixture Models

§3.2.1 showed that the probability density of a random variable $\boldsymbol{x}$ given a data set $\mathcal{X}$ can be approximated by maximizing the likelihood function

$$L(\boldsymbol{\Theta}) = \sum_{i=1}^{k} \log p(\boldsymbol{x}_i|\boldsymbol{\Theta}) + \log p(\boldsymbol{\Theta}) \tag{4-13}$$

where the $\boldsymbol{x}_i$ represent individual measurements. Here the data is assumed to have been generated by a mixture of J distributions (where J is given), each of which has known exponential form parameterized by $\boldsymbol{\Theta_j}$. Thus, if it were known which process j gave rise to a given observation $\boldsymbol{x}_i$, then the conditional distribution of that observation $p(\boldsymbol{x}_i, j|\boldsymbol{\Theta})$ would be completely specified.

To obtain the unconditional density $p(\boldsymbol{x}_i|\boldsymbol{\Theta})$ required for maximization of (4-13), the joint density $p(\boldsymbol{x}_i, j|\boldsymbol{\Theta})$ is summed over j, written in Bayesian notation as a weighted sum of contributions from each distribution:

$$p(\boldsymbol{x}_i|\boldsymbol{\Theta}) = \sum_{j=1}^{J} p(\boldsymbol{x}_i|j, \boldsymbol{\Theta})p(j|\boldsymbol{\Theta}). \tag{4-14}$$

The weights $p(j|\boldsymbol{\Theta})$ are prior probabilities representing the percentage of measurements generated by process j.

4.2.2 Derivation

After incorporating the mixture model, the likelihood function to be maximized becomes

$$L(\boldsymbol{\Theta}) = \sum_{i=1}^{k} \log \left[\sum_{j=1}^{J} p(\boldsymbol{x}_i|j, \boldsymbol{\Theta})p(j|\boldsymbol{\Theta}) \right] + \log p(\boldsymbol{\Theta}). \tag{4-15}$$

Since the mixture densities $p(\boldsymbol{x}_i|j, \boldsymbol{\Theta})$ are exponential, it would be convenient if the logarithm were inside the inner sum. This section demonstrates how Jensen's inequality can be exploited to greatly simplify the expression in (4-15).

In an iterative procedure, maximization of $L(\boldsymbol{\Theta})$ is equivalent to maximization of $\Delta L(\cdot)$, the difference between the current likelihood and that computed from the previous step's parameter estimates $L(\boldsymbol{\Theta}) - L(\tilde{\boldsymbol{\Theta}})$. In other words, rather than finding the parameters $\boldsymbol{\Theta}$ that maximize the likelihood, one can find the parameters that *maximally increase* the likelihood at a given algorithm iteration. The likelihood increase can be written as

$$\Delta L(\cdot) = \sum_{i=1}^{k} \log \left[\frac{\sum_{j=1}^{J} p(\boldsymbol{x}_i|j, \boldsymbol{\Theta})p(j|\boldsymbol{\Theta})}{\sum_{m=1}^{J} p(\boldsymbol{x}_i|m, \tilde{\boldsymbol{\Theta}})p(m|\tilde{\boldsymbol{\Theta}})} \right] + \log \left[\frac{p(\boldsymbol{\Theta})}{p(\tilde{\boldsymbol{\Theta}})} \right]. \tag{4-16}$$

Multiplying the numerator and denominator of the first term by $p(\boldsymbol{x}_i|j, \tilde{\boldsymbol{\Theta}})p(j|\tilde{\boldsymbol{\Theta}})$ and regrouping gives

$$\begin{aligned} & \sum_{i=1}^{k} \log \sum_{j=1}^{J} \left(\frac{p(\boldsymbol{x}_i|j, \tilde{\boldsymbol{\Theta}})p(j|\tilde{\boldsymbol{\Theta}})}{\sum_{m=1}^{J} p(\boldsymbol{x}_i|m, \tilde{\boldsymbol{\Theta}})p(m|\tilde{\boldsymbol{\Theta}})} \right) \left(\frac{p(\boldsymbol{x}_i|j, \boldsymbol{\Theta})p(j|\boldsymbol{\Theta})}{p(\boldsymbol{x}_i|j, \tilde{\boldsymbol{\Theta}})p(j|\tilde{\boldsymbol{\Theta}})} \right) + \log \left[\frac{p(\boldsymbol{\Theta})}{p(\tilde{\boldsymbol{\Theta}})} \right] \\ = & \sum_{i=1}^{k} \log \sum_{j=1}^{J} \alpha_{ij} \left(\frac{p(\boldsymbol{x}_i|j, \boldsymbol{\Theta})p(j|\boldsymbol{\Theta})}{p(\boldsymbol{x}_i|j, \tilde{\boldsymbol{\Theta}})p(j|\tilde{\boldsymbol{\Theta}})} \right) + \log \left[\frac{p(\boldsymbol{\Theta})}{p(\tilde{\boldsymbol{\Theta}})} \right]. \end{aligned} \tag{4-17}$$

The likelihood increase can now be rearranged into a more convenient form. The coefficients α_{ij} are ratios of probabilities and thus real, nonnegative quantities; further, by definition their sum

over j is one. Jensen's inequality [Bis95] states that, for such a set of J coefficients, the following is true:

$$\log\left[\sum_{j=1}^{J}\alpha_{ij}f_j\right] \geq \sum_{j=1}^{J}\alpha_{ij}\log f_j. \tag{4-18}$$

In other words, the logarithm of the sum is lower bounded by the sum of the logarithms. Maximization of $\Delta L(\cdot)$ is thus equivalent to maximization of

$$\sum_{i=1}^{k}\sum_{j=1}^{J}\alpha_{ij}\log\left(\frac{p(\boldsymbol{x}_i|j,\boldsymbol{\Theta})p(j|\boldsymbol{\Theta})}{p(\boldsymbol{x}_i|j,\tilde{\boldsymbol{\Theta}})p(j|\tilde{\boldsymbol{\Theta}})}\right)+\log\left[\frac{p(\boldsymbol{\Theta})}{p(\tilde{\boldsymbol{\Theta}})}\right], \tag{4-19}$$

a task which is now greatly simplified by the exponential form of the mixture densities. A final simplification is performed by noting that terms involving the previous parameters $\tilde{\boldsymbol{\Theta}}$ are constant with respect to the maximization, so the final likelihood ratio reduces to

$$\sum_{i=1}^{k}\sum_{j=1}^{J}\alpha_{ij}\log[p(\boldsymbol{x}_i|j,\boldsymbol{\Theta})p(j|\boldsymbol{\Theta})]+\log p(\boldsymbol{\Theta}). \tag{4-20}$$

The algorithm proceeds by alternating between the expectation step (*E-step*) and the maximization step (*M-step*), which are described below.

4.2.3 Expectation Step

Before the likelihood in (4-20) can be maximized to solve for the unknown quantities, it is necessary to calculate the weights α_{ij}. The weights depend only on the mixture components $p(\boldsymbol{x}_i|j,\tilde{\boldsymbol{\Theta}})$, which are completely determined given a current parameter estimate, and the prior probabilities $p(j|\tilde{\boldsymbol{\Theta}})$, which are assumed available from the previous M-step.

The weights α_{ij} are actually posterior probabilities $p(j|\boldsymbol{x}_i,\tilde{\boldsymbol{\Theta}})$. Using Bayes' Rule,

$$p(j|\boldsymbol{x}_i,\tilde{\boldsymbol{\Theta}}) = \frac{p(\boldsymbol{x}_i|j,\tilde{\boldsymbol{\Theta}})p(j|\boldsymbol{\Theta})}{p(\boldsymbol{x}_i|\tilde{\boldsymbol{\Theta}})}, \tag{4-21}$$

and using the definition of mixture densities in (4-14) gives

$$p(j|\boldsymbol{x}_i,\tilde{\boldsymbol{\Theta}}) = \frac{p(\boldsymbol{x}_i|j,\tilde{\boldsymbol{\Theta}})p(j|\boldsymbol{\Theta})}{\sum_{m=1}^{J}p(\boldsymbol{x}_i|m,\tilde{\boldsymbol{\Theta}})p(m|\boldsymbol{\Theta})} = \alpha_{ij}. \tag{4-22}$$

4.2.4 Maximization Step

Once the posterior probabilities have been calculated, what remains is essentially a maximum likelihood problem in which each term is weighted by α_{ij}. The quantities to be estimated are the parameters $\boldsymbol{\Theta}$ and the prior probabilities $p(j|\boldsymbol{\Theta})$, which can be found by differentiating (4-20)

and setting to zero. Although solution of the resulting equations is problem-specific, the prior probabilities are always given by

$$p(j|\boldsymbol{\Theta}) = \frac{1}{k}\sum_{i=1}^{k}\alpha_{ij}. \tag{4-23}$$

Due to the linearity of differentiation, each parameter set $\boldsymbol{\Theta}_j$ can be estimated independently by taking a sum of derivatives over all i for a particular j.

4.2.5 Robust Clustering

As mentioned previously, data of interest is often corrupted by the influence of unmodeled outlier processes. In an EM framework, it is possible to remove much of this influence by incorporating additional mixture components representing the aggregate outlier processes. Similar techniques can be found in [Wel97, AT00b, CR00b].

Often, nothing is known about processes that generate outliers. The processes can thus be treated in a worst-case sense by modeling them as components with high uncertainty, such as uniform distributions over the appropriate parameter space. The components can be held fixed throughout the EM iterations, or can be allowed to vary along with the other components. The latter is preferred because it produces an approximate characterization of the outlier process when the algorithm converges.

4.2.6 Initialization

Two common challenges must often be faced when implementing any EM algorithm. First, the number of mixtures J must be known in order to formulate the mixture model; in practice it may be difficult to determine this number. Second, EM is guaranteed to converge only on local, not global, optima. Thus, reasonably accurate initial parameter estimates must be supplied if valid solutions are to be obtained.

In the absence of prior information, the probabilities $p(j|\boldsymbol{\Theta})$ are all initialized uniformly to $1/J$, and mixture components for outliers are initialized to random parameter values with high uncertainty. However, without initial values for the remaining mixture parameters, convergence to the correct solution is unlikely. It will be shown in later chapters that in many situations, the Hough transform can determine both the number of mixtures and approximate estimates of the parameters, making it an ideal companion for EM algorithms.

4.3 Markov Chain Monte Carlo

In the Bayesian inference formulation described in §3.2.2, it is necessary to evaluate integrals of the form in (3-19), or in general

$$I = \int f(\boldsymbol{\Theta})p(\boldsymbol{\Theta}|\mathcal{X})\mathrm{d}\boldsymbol{\Theta}. \tag{4-24}$$

When the underlying distributions are exponential (e.g. Bingham or Gaussian), analytic evaluation is straightforward; however, in the general case integrals of this form can be quite complicated and require numerical solution. If it were possible to draw samples $\mathbf{\Theta}_i$ from the distribution $p(\mathbf{\Theta}|\mathcal{X})$, then the integral could be approximated as

$$I \approx \frac{1}{N} \sum_{i=1}^{N} f(\mathbf{\Theta}_i), \tag{4-25}$$

or the average over the most likely values of the distribution.

Typically it is difficult, if not impossible, to directly obtain a set of samples having the required distribution, especially if the parameters reside in a high dimensional space. *Markov chain Monte Carlo* methods address this problem by stochastically traversing the parameter space to find regions of high likelihood.

4.3.1 Parameters as States

A *Markov chain* consists of a sequence of integer-valued random variables $\boldsymbol{y}^k$, whose values indicate the *state* at discrete time k. Given the current state, according to the *Markov property*, future states are independent of the past; that is,

$$p(\boldsymbol{y}^{k+1}|\boldsymbol{y}^k, \boldsymbol{y}^{k-1}, \ldots, \boldsymbol{y}^0) = p(\boldsymbol{y}^{k+1}|\boldsymbol{y}^k). \tag{4-26}$$

The next state thus depends only on the present state. The quantity $p(\boldsymbol{y}^{k+1} = j|\boldsymbol{y}^k = i)$ is known as a *transition probability* p_{ij} and describes the likelihood of transition from state i to state j. If $p(\boldsymbol{y}^k = i)$ is defined as the unconditional probability that the current state is i, then except in degenerate cases,

$$\lim_{k\to\infty} p(\boldsymbol{y}^k = i) = \pi_i. \tag{4-27}$$

In other words, the likelihood of being in state i after many transitions converges to a *steady-state* value π_i that depends only on the transition probabilities, regardless of the starting state. The set of π_i can thus be viewed as a limiting distribution of state probabilities.

If the parameter space of interest is viewed as a state space, and each parameter value $\mathbf{\Theta}_i$ as a particular state, then the steady-state probabilities of a Markov chain on this space define a distribution on the parameters. Thus, if the p_{ij} are chosen appropriately, the desired distribution $p(\mathbf{\Theta}|\mathcal{X})$ can be obtained simply by calculation of the π_i.

4.3.2 Metropolis Algorithm

Metropolis et al [MRR$^+$53] developed a procedure that produces steady-state probabilities having the required distribution. Starting from state i, which represents a particular parameter value $\mathbf{\Theta}_i$, a new candidate state is selected at random by some well-defined process. The conditional likelihood of the new parameters $\mathbf{\Theta}_j$ is computed and compared to that of the present parameters. If the likelihood increases, i.e. $p(\mathbf{\Theta}_j|\mathcal{X}) > p(\mathbf{\Theta}_i|\mathcal{X})$, then the new state is accepted. If the likelihood

decreases, then the new state is accepted with probability equal to the ratio of the likelihoods. Put more concisely, the *likelihood ratio* is computed as

$$\beta_{ij} = \frac{p(\boldsymbol{\Theta}_j|\mathcal{X})}{p(\boldsymbol{\Theta}_i|\mathcal{X})}. \tag{4-28}$$

If $\beta_{ij} > 1$, the new state is accepted; otherwise it is accepted with probability β_{ij} [MRR$^+$53]. If the candidate state is rejected, the current state is retained. The number of visits to each state is tabulated and divided by the number of steps, thus producing an approximation to the steady-state distribution.

Convergence of the Metropolis algorithm relies on two fundamental assumptions. First, the process must be in steady-state; that is, a large number of transitions must occur (to simulate $\lim_{k\to\infty}$) before tabulation of state visits begins. Second, the random perturbations used to generate new states must satisfy *detailed balance* to maintain equilibrium. Transitions must occur such that

$$\pi_j p_{ji} = \pi_i p_{ij} \quad \forall i, j, \tag{4-29}$$

meaning that the relative frequency of transitions from i to j is equal to the frequency of transitions from j to i.

4.3.3 Simulated Annealing

One difficulty with the Metropolis algorithm in practice is that it tends to become "stuck" in regions of locally high likelihood. The method of *simulated annealing*, introduced by Kirkpatrick et al [KGV83], alleviates this problem by incorporating a relaxation parameter T into the Metropolis acceptance criterion. Calculation of the likelihood ratio β_{ij} is unchanged, and the new state is accepted as before if $\beta_{ij} > 1$. However, if $\beta_{ij} < 1$, the new state is accepted with probability $\beta_{ij}^{1/T}$.

When $T = 1$, this criterion reduces to the ordinary Metropolis test. When $T > 1$, however, the probability of acceptance is higher, thus allowing a larger portion of the parameter space to be visited. Simulated annealing draws analogies from metallurgical annealing, in which a molten material is slowly cooled to form a perfect crystal in its lowest possible energy state. The parameter T can thus be viewed as a "temperature" that is lowered according to a predetermined schedule.

Typically, T is initialized to a large value, and states are generated using this value until equilibrium is reached. The temperature is then decreased, usually by a constant multiplicative factor slightly less than 1, and states are again generated until equilibrium is reached. The procedure repeats until $T = 1$, at which point the desired distribution is obtained.

4.4 Adjacency Computation

Some notion of feature correspondence is crucial to any 3-D vision task. In order to formulate correspondence, one must know at the very least which cameras view overlapping scene geometry and which cameras view completely disparate sections of space. Most often, nothing is known in advance about scene structure or camera motion; thus, given a set of images that is neither ordered nor registered, it is impossible to determine which images in the set depict common geometry.

This section describes a simple heuristic technique for approximate determination of camera sets viewing overlapping geometry when roughly-known pose is available.

4.4.1 Using Camera Pose

If precise camera pose is available, then the degree of overlap between the cameras' view volumes can be used to indicate the likelihood of viewing consistent geometry. In feature tracking methods, it is assumed that small temporal separation between images indicates small spatial separation between cameras. Although both structure and motion are unknown in such methods, the assumption of small inter-image camera motion provides a strong constraint on correspondence.

Additional information is required when camera baselines are wider, however. In the City Project [Tel98], images are acquired in arbitrary configurations and in arbitrary order, with pose approximately estimated by the instrumentation. Without scene structure estimates or other cues, the best indication of view coherence is camera adjacency as determined by this approximate pose.

4.4.2 Approximate Determination of Adjacency

There are several ways to determine approximate spatial adjacency given roughly-known pose. One is computation of a Delaunay triangulation of the cameras' positions [dBvKOS91]. Edges in the triangulation are formed so that the circumscribing circle of any particular triangle contains no other points, and these edges thus provide a notion of adjacency. One drawback of this method is that the set of k graph neighbors of a given point is not necessarily the set of k nearest neighbors; that is, edges do not necessarily indicate shortest scene-distance paths to other points. An alternate method of adjacency determination is thus to simply enumerate the set of k closest neighbor positions to each camera position.

Imposing an arbitrary cutoff of k neighbors may produce "false" adjacencies; that is, cameras may be reported as adjacent which are too far apart and therefore unlikely to view overlapping scene geometry. In practice, simple statistical thresholding can be used to alleviate this problem. The sample mean μ and sample variance σ^2 of all reported inter-camera distances are computed and every distance is compared to a threshold. If a given distance is greater than 3σ from μ, the adjacency between its respective cameras is discarded.

This method results in an *adjacency graph* whose nodes represent camera positions and whose edges connect spatially adjacent cameras (Figure 4-5). It is assumed that this graph is available for use in rotational camera registration and in pair-wise determination of camera translation directions. Determination of spatial adjacency is thus one justification for the assumption of approximate pose knowledge made in this thesis.

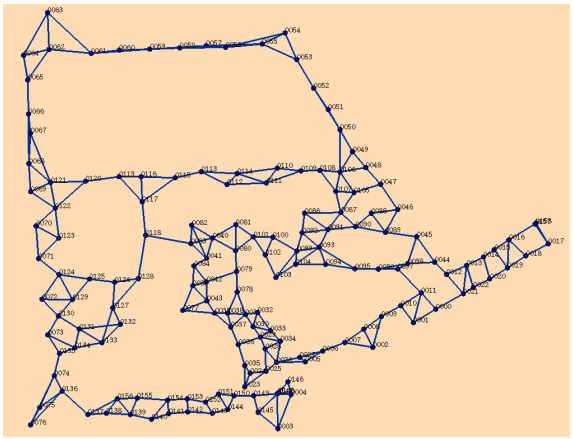


Figure 4-5: Camera Adjacency

Adjacency can be determined by a Delaunay triangulation or by simple computation of nearest neighbors. A sample adjacency graph is depicted; dots represent camera locations, and lines denote detected spatial adjacency.

CHAPTER 5

Orientation Recovery

CLASSICAL STRUCTURE FROM MOTION TECHNIQUES focus on explicit correspondence between point features, simultaneous recovery of both orientation and position, and estimation of 3-D scene geometry. Most formulations attempt to minimize error in the 2-D Euclidean space of the image, as the discrepancy between observed tokens and the projections of their reconstructed 3-D scene elements using estimated camera pose.

The pose recovery methods developed in this thesis differ from classical approaches in several ways. Here, orientation is estimated independently of position by using scene-relative 3-D line directions (*vanishing points*) as features which are invariant with respect to translation. The vanishing points are relatively small in number, and are estimated by fusing thousands of 2-D line features in a projective inference formulation. Correspondence between vanishing points in different cameras is probabilistic, and never determined explicitly. Finally, rotational error is minimized in 3-D rather than in the space of the image; that is, the angular discrepancy between vanishing points serves as an error measure in the more natural metric space of the original scene.

This chapter describes methods for the accurate recovery of rotational pose and its uncertainty among a large set of images; a high-level system diagram is depicted in Figure 5-1. §5.1 illustrates how projective geometry can be exploited to recover orientations independently of positions by detection and alignment of vanishing points, and §5.2 briefly outlines past work in this area. A novel technique for obtaining accurate estimates of multiple vanishing points in a single image is presented in §5.3. Once these features have been extracted from all images, they can be matched across images and used to recover relative rotations between the corresponding cameras. §5.4 reviews a classical, deterministic approach for optimal recovery of rotation in the two-camera case, then presents novel extensions that account for feature uncertainty and produce rotational uncertainty. The approach is extended still further in §5.5, which describes a novel algorithm that uses expectation maximization to simultaneously classify vanishing points, estimate global scene-relative line directions, and refine rotations for an arbitrary number of cameras.

The methods scale linearly with the number of cameras and the number of 2-D line features,

and incorporate all available data to automatically and robustly produce globally optimal orientations. Uncertainty in image line features and vanishing points, and in the orientations themselves, are carefully modeled and estimated using the projective inference techniques introduced in §3.5.3.

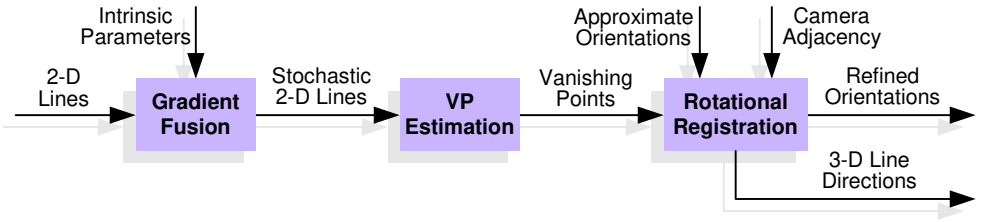


Figure 5-1: Rotational Registration
Line features are detected in all images and fused into vanishing points. Vanishing points in each image are then used to rotationally register the cameras.

5.1 Vanishing Points

Parallel 3-D lines viewed under perspective on an image plane converge to an apparent point of intersection known as the *vanishing point*. Vanishing points arise from particular projective relationships and have several different representations and interpretations. Barnard was among the first to apply them to vision, describing how they could be detected and used to extract information about 3-D geometry [Bar83]. Many researchers have since formalized and elaborated upon the original approach; dozens of papers have been written concerning the reliable detection and estimation of vanishing points for vision tasks from autonomous robot navigation [SABH93] to 3-D scene reconstruction [BB95].

The following sections illustrate the underlying geometry that gives rise to vanishing points and show that they are invariant with respect to viewpoint and translational pose error. This invariance suggests a method for correction of rotational pose error independently of error in position.

5.1.1 Geometry

Consider a 3-D line parallel to some unit direction $\boldsymbol{v}$, and its 2-D projection on the image surface (Figure 5-2). The two quantities are projectively equivalent; that is, any projective ray $\boldsymbol{p}$ that intersects the image line also intersects the scene line. The set of all such rays thus forms a plane $\mathcal{P}$ that includes the focal point, the 2-D line, and the original 3-D line. Let $\boldsymbol{l}$ represent the projective dual of the line, that is the direction on the sphere orthogonal to all rays $\boldsymbol{p}$ through the image line. Since by construction $\boldsymbol{l}$ is orthogonal to $\mathcal{P}$, it must also be orthogonal to the 3-D line; that is, $\boldsymbol{l} \cdot \boldsymbol{v} = 0$.

Similarly, *any* 3-D line parallel to $\boldsymbol{v}$ has a projective dual representation $\boldsymbol{l}_i$ for which $\boldsymbol{l}_i \cdot \boldsymbol{v} = 0$. The direction $\boldsymbol{v}$ is thus the normal to a plane containing all such dual rays $\boldsymbol{l}_i$ (Figure 5-2). Because of the projective equivalence between scene lines and image lines, observation of the 2-D lines alone suffices for this construction; thus, $\boldsymbol{v}$ can be recovered from a set of image lines if their

associated 3-D lines are known to be parallel to $\boldsymbol{v}$. For example, an error expression such as

$$\sum_{i=1}^{k} \boldsymbol{l}_i \cdot \boldsymbol{v} \tag{5-1}$$

can be minimized, or a set of noisy line measurements can be projectively fused to form a distribution on $\boldsymbol{v}$.

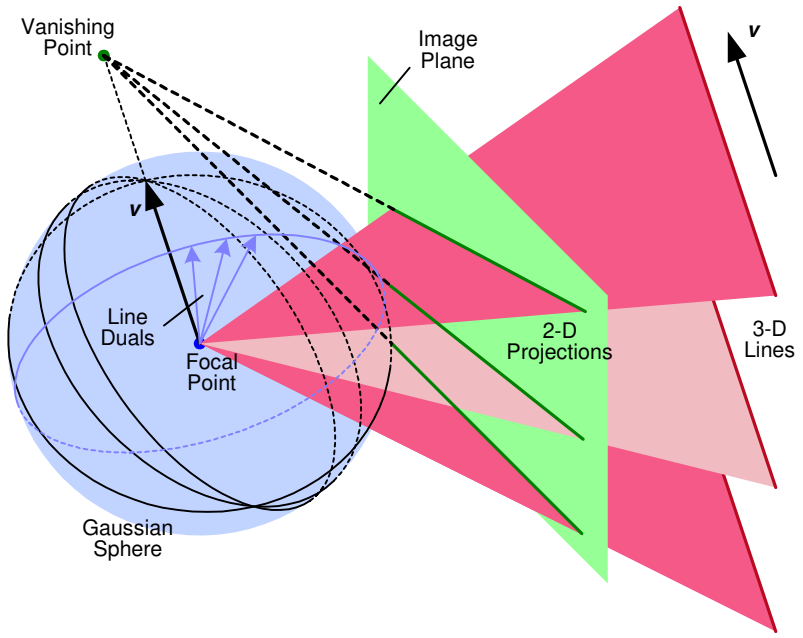


Figure 5-2: Geometry of Vanishing Points
Projections of parallel 3-D lines converge to an apparent vanishing point in the plane. Projectively, the vanishing point represents the intersection of a pencil formed by observations of the lines.

The term "vanishing point" technically refers to a 2-D quantity representing the apparent intersection in the image of 2-D lines that correspond to parallel 3-D lines. As discussed in §1.3.2, the intersection of two or more lines can be represented as a point on the sphere to which the lines' dual representations are orthogonal. This is precisely the relationship stated by $\boldsymbol{l}_i \cdot \boldsymbol{v} = 0$, so the projection of the recovered 3-D line direction $\boldsymbol{v}$ onto the original image is precisely the apparent intersection of all the 2-D lines. Since the intersection and the 3-D line direction represent the same entity, the term "vanishing point" (or VP) will be used in reference to both.

5.1.2 Translation Invariance

Vanishing points on the sphere are by definition scene-relative quantities; that is, they depend only on scene geometry (namely the directions of 3-D lines) and not on the pose of the camera. Cameras viewing lines in the same 3-D directions should thus produce identical vanishing points. Since these vanishing points depend on line *directions* and not *distances*, they are invariant with respect to camera position, except when occluded.

Recall from (1-2) that the transformation which takes points in camera coordinates to points in scene coordinates is specified by a rotation matrix $\boldsymbol{R}$ and a translation vector $\boldsymbol{t}$. Let $\boldsymbol{v}$ represent a vanishing point detected in camera coordinates and let $\boldsymbol{d}$ represent the same vanishing point expressed in scene coordinates. The two quantities are related by

$$\boldsymbol{d} = \boldsymbol{R}\boldsymbol{v}, \tag{5-2}$$

which does not depend on $\boldsymbol{t}$.

Vanishing points $\boldsymbol{v}^{\mathcal{A}}$ and $\boldsymbol{v}^{\mathcal{B}}$ detected with respect to two cameras $\mathcal{A}$ and $\mathcal{B}$ should coincide, if they correspond to the same scene geometry, with each other and with $\boldsymbol{d}$ in scene space. That is, it should be the case that

$$\boldsymbol{R}_{\mathcal{A}}\boldsymbol{v}^{\mathcal{A}} = \boldsymbol{R}_{\mathcal{B}}\boldsymbol{v}^{\mathcal{B}} \tag{5-3}$$

$$\boldsymbol{v}^{\mathcal{A}} = \boldsymbol{R}_{\mathcal{A}}^{\top}\boldsymbol{R}_{\mathcal{B}}\boldsymbol{v}^{\mathcal{B}}. \tag{5-4}$$

However, if the orientations of the cameras, represented by $\boldsymbol{R}_{\mathcal{A}}$ and $\boldsymbol{R}_{\mathcal{B}}$ (where $\boldsymbol{R}_{\mathcal{A}}^{\top}\boldsymbol{R}_{\mathcal{B}}$ represents the *relative* rotation of the cameras), are incorrectly specified, the relationship in (5-4) will not hold. In particular, there is some correction factor $\tilde{\boldsymbol{R}}$ which can be applied on the left of the relative rotation that forces the vanishing points to align. This correction factor is precisely the relative error in orientation between the two cameras, and its application is equivalent to holding camera $\mathcal{B}$ fixed and rotating camera $\mathcal{A}$ by the inverse of $\tilde{\boldsymbol{R}}$.

The above observations imply two important facts: vanishing points and camera orientations can be manipulated independently of camera positions; and orientational discrepancies between cameras can be corrected by rotating the cameras until their vanishing points align. This suggests a strategy for obtaining estimates of relative camera orientation: find the rotation $\boldsymbol{R}$ that best satisfies a set of constraints of the form

$$\boldsymbol{v}_j^{\mathcal{A}} - \boldsymbol{R}\boldsymbol{v}_j^{\mathcal{B}} = 0. \tag{5-5}$$

In the absence of information about the true 3-D directions of estimated vanishing points, only relative, not absolute, orientations can be found. This is because an arbitrary rotation applied to both sides of (5-3) leaves the relationship unchanged; the cameras are thus specified relative to an arbitrary rotational frame of reference.

It is also important to note that the rotation $\tilde{\boldsymbol{R}}$ is itself not specified uniquely in the above formulation. One degree of freedom remains; in particular, any further rotation of $\boldsymbol{v}^{\mathcal{B}}$ about $\boldsymbol{v}^{\mathcal{A}}$ still satisfies (5-5), so correspondence between at least two vanishing points per camera is required for a unique solution. As a result, one of the few domain-specific restrictions imposed in this work is that the viewed scene contain at least two sets of parallel lines, from which vanishing points can be inferred.

5.2 Related Vanishing Point Methods

There is a great deal of literature concerning the estimation of vanishing points and their application to rotational camera calibration. The following sections provide a brief review of existing techniques.

5.2.1 Rotational Pose from Vanishing Points

Decoupling orientation from position by using lines rather than points greatly simplifies the pose recovery formulation and has been utilized in other contexts. For example, Taylor uses explicit correspondence between image lines to determine 3-D line directions, which allows rotations between multiple images to be obtained separately from translations [TKA91, TK92]. Becker uses multiple lines in a single image to estimate vanishing points, and from these infers the camera's internal parameters and orientation [BB95]. Image-based angular discrepancy between detected line segments and the projections of the estimated 3-D lines serves as the error metric. Both methods require manual feature identification and correspondence, whose limitations were outlined in §2.2.5.

Vanishing points have also been used for autonomous navigation tasks whose goal is persistent knowledge of the attitude and position of a mobile robot. Approximate knowledge of 3-D line directions relative to the camera facilitates steering robots through corridors and other structures exhibiting parallelism. Most of these techniques assume three mutually orthogonal line directions [STI90], a constraint which is somewhat limiting but which considerably simplifies implementation and works well in practice. Since steering algorithms must operate in real time, and since pose is continuously updated and corrected through visual feedback, high pose accuracy at each step is unnecessary; thus uncertainty is not carefully analyzed. Moreover, since navigation tasks are often concerned only with the camera's current position and not its motion history, information is typically used only from a single image at a time rather than pooled into a single consistent solution set.

The idea of vanishing point correspondence across multiple images to recover rotational pose has also been examined. Leung describes a graph matching algorithm for obtaining correspondence between vanishing point features in multiple images [LM96]. The angles between vanishing points are noted as being invariant under rotation and are thus used to constrain candidate matches. Although the authors imply that the matching technique can be used for recovery of relative rotational pose, they do not discuss direct application to this problem. Moreover, results are only reported for pairs of images, and uncertainty is not addressed.

5.2.2 Vanishing Point Estimation

The vast majority of vanishing point estimation techniques rely exclusively on the Hough transform (for example, [LMLK94, Shu99]). However, as noted in §4.1.4, the accuracy of Hough transform techniques is inherently limited by discretization artifacts, and uncertainty in the estimates is difficult to characterize. More precise estimation in continuous spaces has been approached in several ways. Some techniques estimate vanishing points by intersecting all possible lines in the image plane and searching for high-incidence regions [MK95, LZ98], but suffer from instability and degeneracies when the image lines are nearly parallel, in which case intersections are near infinity. This problem can be solved by using projective rather than Euclidean line representations and computing intersections on the sphere; [MA84] present such a formulation, but their method still computes all possible intersections, and is thus quadratic in the number of line features.

In continuous space approaches, line classification is a daunting task. Clustering of proximal line intersections, for example by deterministic k-means algorithms, works well if the number of outliers is small, but produces somewhat poor estimates in the presence of significant noise. Collins

[Col93] proposes a simple and elegant solution that uses a Hough transform for reliable detection and clustering, followed by a more careful projective inference approach for accurate estimation of each vanishing point and characterization of error in the estimates. This hybrid clustering approach is deterministic, however, and uses a hard threshold to reject outliers, unnecessarily producing bias in the estimates. A similar hybrid approach, which uses a novel mixture model and probabilistic clustering rather than hard thresholds, is developed in the sections that follow.

5.3 Single-Camera Formulation

Image lines, represented by projective random variables $\boldsymbol{x}_i$, serve as the primary features for vanishing point recovery. However, the collection of lines in a given image is completely unclassified; that is, lines are not grouped into parallel sets, and random outliers are mixed with the true data. The problem of vanishing point estimation thus has three components. First, the number of groups J (that is, the number of prominent 3-D line directions) must be established. Next, lines $\boldsymbol{x}_i$ must be classified according to their corresponding 3-D direction or discarded as outliers. Finally, the vanishing point $\boldsymbol{v}_j$ for each group must be estimated.

These three problems are tightly coupled. A complete, deterministic classification of line features reduces the estimation problem to a set of J straightforward projective inference tasks, one for each line group; similarly, knowledge of the 3-D directions $\boldsymbol{v}_j$ facilitates accurate line classification. The EM algorithm described in §4.2 can effectively solve both problems by iteratively alternating between estimation of vanishing point directions and probabilistic classification. If properly initialized, the algorithm is guaranteed to converge on the optimal solution.

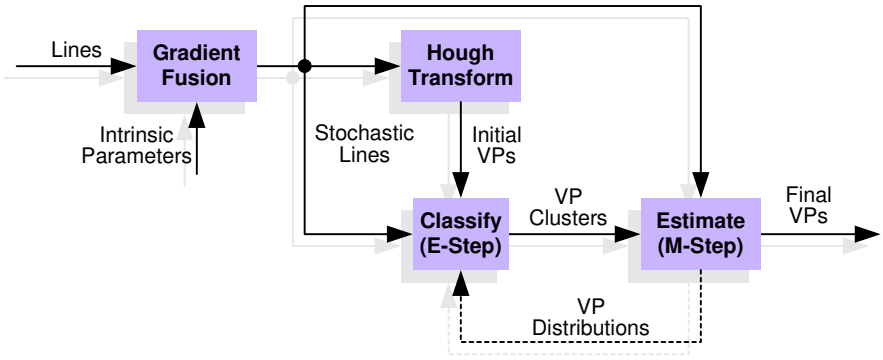


Figure 5-3: Vanishing Point Estimation
Stochastic line features from a single image are obtained from the fusion of gradient pixel distributions. These lines are used in a Hough transform, which finds prominent 3-D line directions to initialize an expectation maximization algorithm. The result is a set of accurately estimated vanishing points and a classification of lines.

This section presents an EM formulation for line classification and vanishing point estimation as inference problems on the sphere. For the moment, it is assumed that the algorithm is appropriately initialized; that is, the number of prominent 3-D directions J is known, and their approximate directions are available. §5.3.4 describes an efficient Hough transform technique for the recovery of these quantities. The overall system is summarized in Figure 5-3.

5.3.1 Mixture Model for Vanishing Points

Figure 5-2 shows that vanishing points are projective quantities constrained by the dual points of contributing projective lines. Thus each of the J observed vanishing points $\boldsymbol{v}_j$ is treated as a Bingham random variable with unknown parameter matrix $\boldsymbol{M}_j^{\boldsymbol{v}}$, formed by fusion of the appropriate uncertain line features. The entire data set $\mathcal{X}$ is a collection of unclassified samples from the set of random variables $\mathcal{V} = \{\boldsymbol{v}_0 \ldots, \boldsymbol{v}_J\}$, where $\boldsymbol{v}_0$ represents an unknown outlier distribution; thus, $\mathcal{X}$ is described by a mixture of $J+1$ Bingham densities, so that

$$p(\boldsymbol{x}_i|\mathcal{V}) = \sum_{j=0}^{J} p(\boldsymbol{x}_i|j,\mathcal{V})p(j|\mathcal{V}), \tag{5-6}$$

Each observation $\boldsymbol{x}_i$ represents an uncertain line feature with known Bingham distribution $\mathcal{B}_3(\boldsymbol{x}_i; \boldsymbol{M}_i)$. The parameter matrices $\boldsymbol{M}_i$ are obtained from constituent gradient pixels when the line feature is detected.

With the form of the underlying distributions specified, the expectation and maximization steps of the algorithm can proceed. These steps are described in the next two sections.

5.3.2 E-Step

Recall that in the E-step of the EM algorithm, a set of posterior probabilities α_{ij} is computed from (4-22) which effectively "weigh" each observation $\boldsymbol{x}_i$ when the parameters $\boldsymbol{M}_j^{\boldsymbol{v}}$ for distribution j are estimated in the subsequent M-step. Assuming the prior probabilities $p(j|\mathcal{V})$ and current parameter estimates $\boldsymbol{M}_j^{\boldsymbol{v}}$ are known (either from the previous step or from initialization), all that remains is to calculate the mixture component probabilities $p(\boldsymbol{x}_i|j,\mathcal{V})$.

Intuitively, $p(\boldsymbol{x}_i|j,\mathcal{V})$ represents the likelihood of the line $\boldsymbol{x}_i$ given that it belongs to vanishing point $\boldsymbol{v}_j$. If the line observation were deterministic, this likelihood would simply be given by $\mathcal{B}_3(\boldsymbol{x}_i; \boldsymbol{M}_j^{\boldsymbol{v}})$. However, $\boldsymbol{x}_i$ is a stochastic measurement which is itself represented by a probability distribution.

Bayesian arguments can therefore be used to determine the likelihood. Let $\boldsymbol{x}_i^0$ represent a particular measurement from the distribution of random variable $\boldsymbol{x}_i$; then

$$p(\boldsymbol{x}_i|j,\mathcal{V},\boldsymbol{x}_i^0) = \frac{1}{c(\boldsymbol{M}_j^{\boldsymbol{v}})}\exp((\boldsymbol{x}_i^0)^\top \boldsymbol{M}_j^{\boldsymbol{v}}(\boldsymbol{x}_i^0)). \tag{5-7}$$

Now, to eliminate the dependence on the particular value of the random variable, the joint likelihood is integrated over all possible measurement values:

$$\begin{aligned} p(\boldsymbol{x}_i|j,\mathcal{V}) &= \int p(\boldsymbol{x}_i|j,\mathcal{V},\boldsymbol{x}_i^0)p(\boldsymbol{x}_i^0)\mathrm{d}\boldsymbol{x}_i^0 \\ &= \int \frac{1}{c(\boldsymbol{M}_j^{\boldsymbol{v}})}\exp\left[(\boldsymbol{x}_i^0)^\top \boldsymbol{M}_j^{\boldsymbol{v}}(\boldsymbol{x}_i^0)\right]\frac{1}{c(\boldsymbol{M}_i)}\exp\left[(\boldsymbol{x}_i^0)^\top \boldsymbol{M}_i(\boldsymbol{x}_i^0)\right]\mathrm{d}\boldsymbol{x}_i^0 \\ &= \frac{1}{c(\boldsymbol{M}_j^{\boldsymbol{v}})c(\boldsymbol{M}_i)}\int \exp[(\boldsymbol{x}_i^0)^\top(\boldsymbol{M}_j^{\boldsymbol{v}}+\boldsymbol{M}_i)(\boldsymbol{x}_i^0)]\mathrm{d}\boldsymbol{x}_i^0 \\ &= \frac{c(\boldsymbol{M}_j^{\boldsymbol{v}}+\boldsymbol{M}_i)}{c(\boldsymbol{M}_j^{\boldsymbol{v}})c(\boldsymbol{M}_i)}. \end{aligned} \tag{5-8}$$

Thus, the likelihood can be calculated as a ratio of normalizing coefficients from three different Bingham densities.

5.3.3 M-Step

Once the weights are known, the Bingham parameter matrices $\boldsymbol{M}_j^{\boldsymbol{v}}$ of each vanishing point distribution can be estimated by maximizing the likelihood function in (4-20). The exponential form of the Bingham distribution makes calculation of the log likelihood straightforward. Every parameter matrix $\boldsymbol{M}_j^{\boldsymbol{v}}$ is computed independently as the fusion of k observations $\boldsymbol{x}_i$, each weighted by the α_{ij} from the E-step. Combining (4-20) and (3-34) yields

$$\boldsymbol{M}_j^{\boldsymbol{v}} = \sum_{i=1}^{k} \alpha_{ij} \boldsymbol{M}_i + \boldsymbol{M}_j^{\boldsymbol{0}} \tag{5-9}$$

where $\boldsymbol{M}_j^{\boldsymbol{0}}$ is the prior distribution on $\boldsymbol{v}_j$ (see §5.3.4).

There are several significant differences between this formulation and that presented by Collins [Col93]. First, all vanishing points are estimated simultaneously as a mixture of distributions with soft classification rather than one at a time by hard classification. The mixture model reduces bias and other artifacts caused by somewhat arbitrary thresholding. Second, vanishing points are estimated as equatorial probability *distributions* rather than as deterministic bipolar vectors that maximize a likelihood function. Finally, the data fusion technique uses the full description of uncertainty in each measurement, as opposed to heuristic scalar weighting by line length.

5.3.4 Initialization

Proper formulation and implementation of the EM algorithm decribed above relies on knowledge of the number of vanishing points J. In addition, as mentioned in §4.2.6, guaranteed convergence to the correct solution (i.e. avoidance of local optima) requires the availability of reasonably accurate parameter estimates. Both required quantities can be obtained using a Hough transform.

§4.1 describes the Hough transform as a valuable tool for approximate solution of overconstrained systems specified by (4-1) and (4-2), and for initialization of more accurate techniques. In vanishing point estimation, equations in the system take the form

$$\begin{array}{rcccl} f(\boldsymbol{x}_i, \boldsymbol{v}_j) & = & \boldsymbol{x}_i \cdot \boldsymbol{v}_j & = & 0 \\ g(\boldsymbol{v}_j) & = & \boldsymbol{v}_j \cdot \boldsymbol{v}_j - 1 & = & 0 \end{array}$$

where the $\boldsymbol{x}_i$ are the modal or polar directions of the data. The parameter constraints $g(\cdot)$ are inherently satisfied because $\boldsymbol{v}_j$ is a projective quantity defined on $\mathbb{S}^2$. Since vanishing points have arbitrary directions, the entire surface is discretized using the cubic parameterization of §4.1.5. Geometrically, each constraint $f(\cdot)$ represents a plane through the focal point with normal $\boldsymbol{x}_i$; intersection of this plane with three faces of the unit cube results in a set of at most three lines, which are easily discretized using standard line clipping and drawing algorithms [FvDFH90]. Uncertainty in the lines can be incorporated using the proximity function described in §4.1.6.

When all data has been accumulated, peaks in the accumulation space, which represent likely vanishing points directions, are detected using the methods of §4.1.7. The number of statistically

significant peaks is used as the number of mixture components J, and the peak directions (i.e. the vectors from the center of the cube through each peak) are used to initialize the EM algorithm.

Peak directions also serve as prior densities $p(\boldsymbol{v}_j)$, each of which is formulated as an equatorial Bingham density ($\kappa_1 \ll \kappa_2 \leq 0$) whose modal axis is aligned with the peak direction. The parameter matrix $\boldsymbol{M}_j^0$ for the prior density can be determined using the bipolar approximation of §4.1.8, then computing its dual distribution.

5.4 Two-Camera Registration

As mentioned in §5.1.2, the relative rotational offset between a given pair of cameras can be determined by aligning two or more distinct vanishing points viewed by both cameras. Thus, once vanishing points have been estimated by the above techniques, relative orientations can be found. This section first presents a classical, deterministic formulation for the two-camera case (§5.4.1), then introduces two novel extensions. §5.4.2 describes a model for uncertainty in the resulting rotations as the fusion of deterministic samples from a Bingham distribution on $\mathbb{S}^3$, and proceeds by considering how uncertainty in the vanishing points themselves affects the distribution of the resulting rotation.

The estimation methods assume that correspondence between vanishing points in different cameras is known, which is generally not the case. The remaining sections address the correspondence problem for the two-camera case and ambiguities that arise in practice.

5.4.1 Classical Pair Registration

Consider two cameras $\mathcal{A}$ and $\mathcal{B}$, each of which views a common set of J vanishing points. Let $\boldsymbol{v}_j^{\mathcal{A}}$ and $\boldsymbol{v}_j^{\mathcal{B}}$ denote the directions of a particular line direction $\boldsymbol{d}_j$ as seen by each camera, and further assume that camera $\mathcal{B}$ is free to rotate while camera $\mathcal{A}$ is held fixed. The problem is now to estimate a single rotation in the form of a quaternion $\boldsymbol{q}$ which, when applied to camera $\mathcal{B}$ and its vanishing points, produces the best alignment between the $\boldsymbol{v}_j^{\mathcal{A}}$ and the $\boldsymbol{v}_j^{\mathcal{B}}$.

This problem has been studied in various contexts. The most relevant discussion is in [Hor87, Hor91], who derives the optimal $\boldsymbol{q}$ as part of a more general 3-D to 3-D correspondence problem. The objective is to determine

$$\text{argmin}_{\boldsymbol{q}} \sum_{j=1}^{J} \|\boldsymbol{v}_j^{\mathcal{A}} - \boldsymbol{R}(\boldsymbol{q})\boldsymbol{v}_j^{\mathcal{B}}\|^2 \tag{5-10}$$

$$= \text{argmin}_{\boldsymbol{q}} \left[\boldsymbol{q}^\top \sum_{j=1}^{J} \boldsymbol{A}_j^\top \boldsymbol{A}_j \boldsymbol{q} \right] \tag{5-11}$$

$$= \text{argmin}_{\boldsymbol{q}} \boldsymbol{q}^\top \boldsymbol{A} \boldsymbol{q} \tag{5-12}$$

i.e. the $\boldsymbol{q}$ that minimizes a quadratic error function (a more detailed explanation can be found in §A.5). Each 4×4 matrix $\boldsymbol{A}_j$ is constructed as a linear function of its constituent vanishing points $\boldsymbol{v}_j^{\mathcal{A}}$ and $\boldsymbol{v}_j^{\mathcal{B}}$. The solution to (5-12) is the eigenvector corresponding to the minimum eigenvalue of the symmetric 4×4 matrix $\boldsymbol{A}$.

These methods produce optimal results using their respective error metrics but, aside from the scalar error residual, produce no notion of uncertainty in the result $\boldsymbol{q}$. There is also little treatment of uncertainty in the measurements themselves, with a few exceptions. Kanatani [Kan94] and Antone [AT00b] propose scalar weights that value correspondences by the certainty of their constituent direction pairs, but this weighting scheme allows only a limited description of randomness in the underlying observations. Chang [Cha89] and Prentice [Pre89] present errors-in-variables models for so-called *spherical regression*, but consider only limiting cases in which data points have very concentrated symmetric bipolar distributions.

The next section addresses these issues of uncertainty in more detail, proposing methods for obtaining descriptive error measures on the estimated rotations by incorporation of uncertainty in the underlying data.

5.4.2 Stochastic Pair Registration

Recall from §3.4.1 that rotational uncertainty can be described as a Bingham distribution on $\mathbb{S}^3$ characterized by a 4×4 matrix of parameters $\boldsymbol{M}_{\boldsymbol{q}}$. The matrix $\boldsymbol{A}$ obtained in (5-12), when properly normalized, is analogous to a sample second moment matrix: it is symmetric and positive semidefinite, and its eigvenvalues sum to unity. $\boldsymbol{A}$ is composed of a sum of J matrices $\boldsymbol{A}_j^\top \boldsymbol{A}_j$ that also possess these properties. Each of these constituent "samples" $\boldsymbol{q}_j$ is formed from an individual vanishing point correspondence, contributing at most rank 2 to the sum. Thus, a parameter matrix $\boldsymbol{M}_{\boldsymbol{q}}$ for the distribution on the resulting quaternion $\boldsymbol{q}$ can be obtained directly from $\boldsymbol{A}$ using the results from §B.2.

This method can be used only for data sets that give each measurement equal weight. Extension to scalar-weighted data is rather straightforward, involving a weighted sum of constituent sample matrices normalized by the total weight. In the general case, however, where vanishing points are described by Bingham-distributed uncertainty, the Bingham distribution on $\mathbb{S}^3$ induced by each correspondence must be computed.

Every matrix $\boldsymbol{A}_j$ is a function of the vanishing point directions in its constituent correspondence. Thus, the parameters of the Bingham distribution associated with $\boldsymbol{A}_j$ can also be expressed as a function of these directions. Given particular sample values of vanishing point distributions $\boldsymbol{v}_j^{\mathcal{A}}$ and $\boldsymbol{v}_j^{\mathcal{B}}$, define $\boldsymbol{M}(\boldsymbol{v}_j^{\mathcal{A}}, \boldsymbol{v}_j^{\mathcal{B}})$ as the parameter matrix of the associated distribution. The contribution of correspondence j can then be obtained by Bayesian integration over all possible sample values of the two constituent vanishing points:

$$\begin{aligned} p(\boldsymbol{q}_j) &= \int_{\mathbb{S}^2}\int_{\mathbb{S}^2} p(\boldsymbol{q}_j|\boldsymbol{v}_j^{\mathcal{A}}, \boldsymbol{v}_j^{\mathcal{B}}) p(\boldsymbol{v}_j^{\mathcal{A}}) p(\boldsymbol{v}_j^{\mathcal{B}}) \mathrm{d}\boldsymbol{v}_j^{\mathcal{A}} \mathrm{d}\boldsymbol{v}_j^{\mathcal{B}} \\ &= \int_{\mathbb{S}^2}\int_{\mathbb{S}^2} \mathcal{B}_4(\boldsymbol{q}_j; \boldsymbol{M}(\boldsymbol{v}_j^{\mathcal{A}}, \boldsymbol{v}_j^{\mathcal{B}})) \mathcal{B}_3(\boldsymbol{v}_j^{\mathcal{A}}; \boldsymbol{M}^{\mathcal{A}}) \mathcal{B}_3(\boldsymbol{v}_j^{\mathcal{B}}; \boldsymbol{M}^{\mathcal{B}}) \mathrm{d}\boldsymbol{v}_j^{\mathcal{A}} \mathrm{d}\boldsymbol{v}_j^{\mathcal{B}}. \end{aligned} \tag{5-13}$$

Approximation of this quantity by a Bingham distribution is described in §A.6.

Once distribution parameters $\boldsymbol{M}_j$ have been determined for each correspondence $\boldsymbol{q}_j$, the final distribution is given simply by the parameters

$$\boldsymbol{M}_{\boldsymbol{q}} = \sum_{j=1}^{J} \boldsymbol{M}_j. \tag{5-14}$$

5.4.3 Two-Camera Vanishing Point Correspondence

The registration methods above assume that one-to-one correspondence has been established between vanishing points detected in a given pair of images. Determination of correspondence, as mentioned in §1.4.3, is generally a difficult task without additional information; however, if the two relevant cameras view a significant portion of common scene geometry, then the assumption of approximately known initial pose is typically enough to establish consistent correspondence. This section presents a few heuristic methods to determine local (i.e. two-camera) correspondence that initializes a global technique described in §5.5.3.

If two cameras view overlapping scene geometry, then the sets of vanishing points detected in each camera are likely to contain common members. In this case it is assumed that cameras $\mathcal{A}$ and $\mathcal{B}$ have in common a set of vanishing points related by a single rotation $\boldsymbol{q}$ which preserves the relative (intra-camera) angles between them.

Since a minimum of two correspondences is needed to find a unique rotation relating the two cameras, relative angles between pairs of vanishing points in each camera can be used as a matching criterion. For example, if the angle between $\boldsymbol{v}_1^{\mathcal{A}}$ and $\boldsymbol{v}_2^{\mathcal{A}}$ differs significantly from that between $\boldsymbol{v}_1^{\mathcal{B}}$ and $\boldsymbol{v}_2^{\mathcal{B}}$, then these two pairs (which constitute a *pair couplet*) cannot possibly match. Thus, only those pair couplets are considered whose relative angles are within a small threshold of each other. Angular thresholds are related to the Bingham parameters of the respective vanishing point distributions; highly concentrated distributions thus have tighter thresholds than do distributions with more spread. It should also be noted that, since vanishing points are axial, there are two possible angles to consider that sum to π; the minimum of the two is always chosen for angle comparison (Figure 5-4a).

A set of scores $\mathcal{S} = \{s_1, \ldots, s_k\}$ is computed, one for each pair couplet meeting the relative angle criterion above. Several additional criteria are also evaluated for each score. First, the pair from camera $\mathcal{B}$ is rotated to the pair from camera $\mathcal{A}$ by $\boldsymbol{q}$ using the deterministic pair registration technique from §5.4.1; the direction of a given vanishing point is taken as the major axis of its associated Bingham distribution. The angle of rotation θ_i required to align the two pairs is noted, and the remaining vanishing points from camera $\mathcal{B}$ are then rotated by $\boldsymbol{q}$ and compared with each vanishing point from camera $\mathcal{A}$. The total number N_i of vanishing points that align to counterparts in camera $\mathcal{A}$ within a threshold angle, including the original pair, is also noted.

Each score is then computed as $s_i = N_i/\theta_i$. This score emphasizes correspondence sets containing many matches, while preserving the assumption that the relative rotations are already known to reasonable accuracy. The correspondence set with the highest score is chosen as the "correct" set, for later use in global rotational alignment (§5.5.3).

Let J represent the number of vanishing points viewed by each camera. Then enumeration of all possible vanishing point pairs per camera is $\mathcal{O}(J^2)$, and enumeration of all possible pair couplets is $\mathcal{O}(J^4)$. Computation of correspondence sets for each couplet is $\mathcal{O}(J^2)$, raising the overall work required to $\mathcal{O}(J^6)$. The order of this rather brute-force technique is high, but the value of J is small—typically less than 6.

5.4.4 Correspondence Ambiguity

Rotation as presented in §5.4 requires correspondence between signed directions, but vanishing points are axial (i.e. undirected) quantities. In truth, for each pair couplet meeting the relative

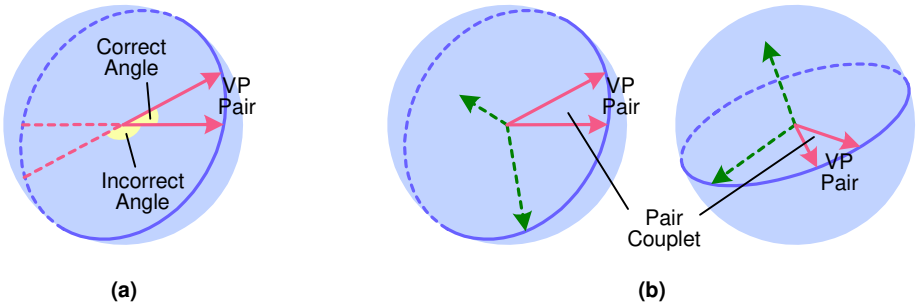


Figure 5-4: Pair Couplets
(a) There are two possible choices when comparing relative angles in axial quantities. By convention, the smaller of the two is chosen. (b) A matching pair couplet is depicted. Relative angles between the pairs are identical despite the fact that the cameras are not rotationally aligned.

angle criterion, *two* different rotations (and associated scores) must be computed, one for each combination of sign that maintains relative angle consistency. This is illustrated in Figure 5-5. There are thus twice as many scores to be evaluated, but evaluation otherwise proceeds as described above.

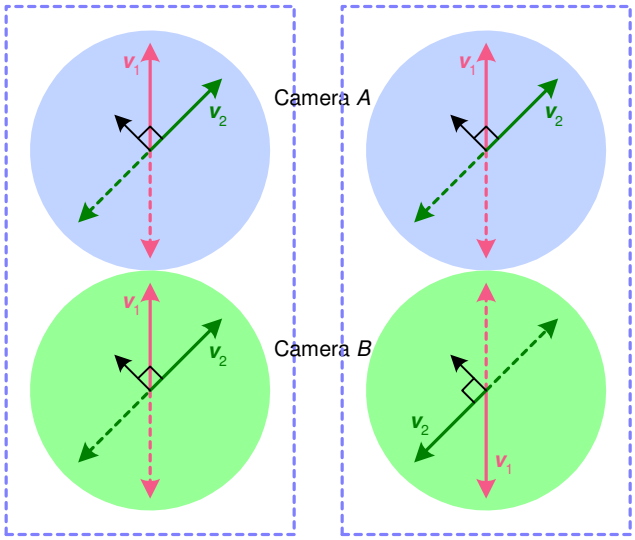


Figure 5-5: Two Solutions to Optimal Rotation
There are two possible rotations, differing by $180°$, that align axial features.

Other ambiguities can also arise that are not so easily resolved, especially in urban scenes consisting of mutually orthogonal lines (Figure 5-6). Since relative angles between multiple pairs of vanishing points can be identical within a single image, there may exist several plausible match configurations. The matching algorithm must thus rely on prior knowledge, such as the fact that since approximate pose is known, the rotational discrepancy between any two cameras should be relatively small; solutions implying large rotation are unlikely (hence the score criteria in §5.4.3).

Another assumption is that nearly all urban scenes contain vertical lines, so the vanishing point directions closest to "up" in each image can be assumed to match, thus constraining correspondence sets somewhat and reducing computation to $\mathcal{O}(J^4)$.

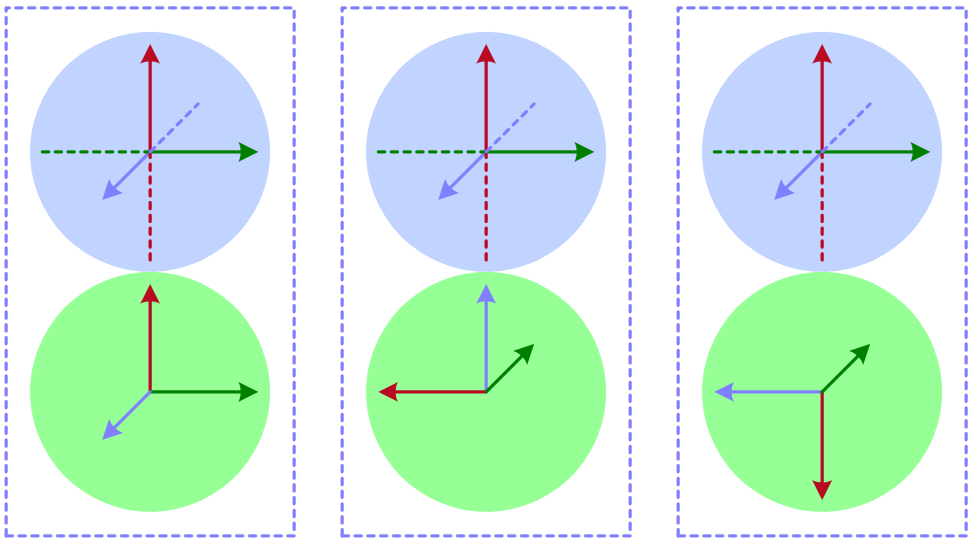


Figure 5-6: Vanishing Point Correspondence Ambiguity
An example of ambiguities that can arise in symmetric configurations. Without additional information, there is no way to differentiate between match candidates.

If there is significant error in initial rotational pose and if correspondence ambiguities exist, the matching algorithm can fail, finding a plausible but incorrect match assignment. In City Project data, initial camera pose is sufficiently accurate to avoid this problem.

5.5 Multi-Camera Registration

The above treatment of rotational registration is deficient in two main respects. First, it detemines explicit or "hard" correspondence among vanishing points rather than stochastic correspondence; and second, it considers only two cameras at a time. This section presents a multi-camera extension for rotational registration which addresses the above concerns and produces a globally-optimal set of camera orientations along with their associated uncertainty.

It may seem at first that the two-camera method can be directly extended to handle multiple cameras in a sequential fashion. This type of sequential approach is often used when data consists of image streams, such as video [FZ98]; images are locally registered a pair or triple at a time as the stream progresses. Such purely local methods invariably propagate and accumulate error, however, since pose estimates from early images feed subsequent images. To minimize this effect and distribute error equally among all cameras, pose must be recovered globally by simultaneously considering all available data.

As is typical in 3-D vision problems, pose recovery consists of the two coupled sub-problems of correspondence and registration. That is, given a grouping of vanishing points into sets, where

each set represents observations of a true scene-relative line direction, estimation of relative rotations becomes simpler; and, conversely, given a set of accurate camera orientations, determination of correspondence is simplified. These facts, at a high level, suggest an iterative bundle-adjustment scheme that alternately estimates orientations given correspondence, then establishes correspondence given orientations.

The basic idea is shown in Figure 5-7. Rotations and correspondence are initially produced by exhaustive search (§5.5.3), and global (scene-relative) line directions are estimated based on vanishing point clusters. Each camera is then rotated until its vanishing points optimally align with these global directions, and the process repeats. There are two levels of feedback in the process: one at the high level of rotational bundle adjustment, and the other in the estimation of global line directions, which alternates between determination of probabilistic correspondence and estimation of directional distributions.

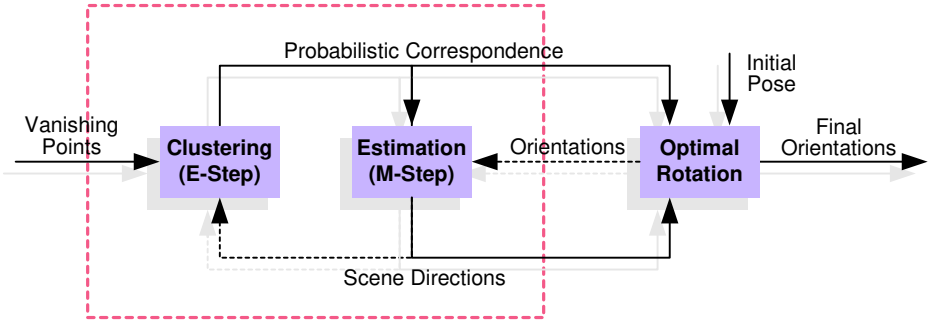


Figure 5-7: Global Orientation Recovery
Vanishing points are aligned with one another to determine camera orientations. A two-level feedback hierarchy is used; at the top level, rotations and scene-relative line directions are alternately estimated. The dotted box encloses the bottom level of feedback, in which vanishing points are classified and line directions estimated.

5.5.1 EM for Multi-Camera Registration

This alternation between classification and estimation suggests application of an EM algorithm, which would also circumvent the need for explicit correspondence and provide an adequate probabilistic estimation framework. At its core, the problem is to determine the probability distributions of a set of rotations in the form of quaternions, $\mathcal{Q} = \{\boldsymbol{q}_1, \ldots, \boldsymbol{q}_M\}$, based solely on a data set $\mathcal{V} = \{\mathcal{V}^1, \ldots, \mathcal{V}^M\}$, where $\mathcal{V}^i$ is the set of vanishing points $\boldsymbol{v}_j^i$ detected in image i. Probabilistically, this can be written as

$$\text{argmax}_{\mathcal{Q}}[p(\mathcal{Q}|\mathcal{V})]. \tag{5-15}$$

However, the rotations depend on scene-relative line directions $\mathcal{D}$, as well as correspondence $\mathcal{C}$ between these directions and the vanishing points in each individual image. The likelihood to be

maximized can thus be rewritten as

$$\begin{aligned} p(\mathcal{Q}|\mathcal{V}) &= \int_{\mathcal{D}} p(\mathcal{Q},\mathcal{D}|\mathcal{V})\mathrm{d}\mathcal{D} \\ &= \int_{\mathcal{D}} p(\mathcal{Q}|\mathcal{D},\mathcal{V})p(\mathcal{D}|\mathcal{V})\mathrm{d}\mathcal{D} \\ &= \int_{\mathcal{D}} \sum_{\mathcal{C}} p(\mathcal{Q},\mathcal{C}|\mathcal{D},\mathcal{V})p(\mathcal{D}|\mathcal{V})\mathrm{d}\mathcal{D} \\ &= \int_{\mathcal{D}} \sum_{\mathcal{C}} p(\mathcal{Q}|\mathcal{C},\mathcal{D},\mathcal{V})p(\mathcal{C}|\mathcal{D},\mathcal{V})p(\mathcal{D}|\mathcal{V})\mathrm{d}\mathcal{D}. \end{aligned} \tag{5-16}$$

Note that a sum is taken over $\mathcal{C}$ rather than an integral, because the set of correspondence configurations is discrete. The quantity $p(\mathcal{D}|\mathcal{V})$ represents the prior distribution on global line directions given only the vanishing point data; this quantity is taken to be uniform, since in the absence of rotational pose, nothing is known about this distribution. The quantity $p(\mathcal{C}|\mathcal{D},\mathcal{V})$ is the prior distribution on correspondence given only global line directions and vanishing points (not rotations). This distribution can be approximated from the pair-wise correspondences established in §5.4.3; further details are given in §5.5.3.

If $\mathcal{D}$ and $\mathcal{C}$ are treated jointly, then maximization of (5-16) is similar to maximization of a likelihood formed by a mixture of distributions, where the mixture components are global line directions and correspondence. The high-level EM algorithm alternates between two steps: first, compute the likelihoods $p(\mathcal{C},\mathcal{D}|\mathcal{Q},\mathcal{V})$; next, maximize the expression

$$\int_{\mathcal{D}} \sum_{\mathcal{C}} p(\mathcal{C},\mathcal{D}|\mathcal{Q},\mathcal{V}) \log p(\mathcal{Q}|\mathcal{C},\mathcal{D},\mathcal{V})\mathrm{d}\mathcal{D} \tag{5-17}$$

The likelihoods computed in the first step thus serve as weights on the conditional log-likelihood to be maximized. Conditioned on line directions and correspondence, the quaternions are independent of one another because vanishing points in each camera can be rotated in isolation to optimally align with the global line directions. Thus,

$$\begin{aligned} \log p(\mathcal{Q}|\mathcal{C},\mathcal{D},\mathcal{V}) &= \log \prod_{i=1}^{M} p(\boldsymbol{q}_i|\mathcal{C},\mathcal{D},\mathcal{V}) \\ &= \sum_{i=1}^{M} \log p(\boldsymbol{q}_i|\mathcal{C},\mathcal{D},\mathcal{V}) \end{aligned} \tag{5-18}$$

and each quaternion can be estimated independently. Maximization proceeds as described in §5.4.2, with the Bingham distribution of orientation $\boldsymbol{q}_i$ specified by the parameter matrix $\boldsymbol{M}_i^{\boldsymbol{q}}$, which represents the weighted sum of correspondence matrices of the form in (5-14).

5.5.2 EM for Multi-Camera Correspondence

The above formulation solves the M-step of the bundle adjustment, but the E-step still remains—the likelihoods $p(\mathcal{C},\mathcal{D}|\mathcal{Q},\mathcal{V})$ must be computed. Intuitively, these likelihoods represent distributions on correspondence $\mathcal{C}$ and scene-relative line directions $\mathcal{D}$ given the current set of orientation

estimates $\mathcal{Q}$. $\mathcal{C}$ and $\mathcal{D}$ are coupled, however; knowledge of the line directions influences the groupings, and vice versa.

Let $\tilde{\boldsymbol{v}}_j^i$ represent vanishing point j in image i after rotation by $\boldsymbol{q}_i$; the set of all such directions serves as the pool of data to be grouped. Further, let $\boldsymbol{d}_k$ represent a particular scene-relative 3-D line direction. The problem then becomes to simultaneously estimate the $\boldsymbol{d}_k$ and classify the $\tilde{\boldsymbol{v}}_j^i$.

This formulation is identical to the vanishing point estimation problem posed in §5.3. The collective data set $\tilde{\mathcal{V}}$ is drawn from a weighted mixture of Bingham distributions of $\boldsymbol{d}_k$; the only difference is that these distributions are now bipolar rather than equatorial, but this fact does not affect the algorithm. Application of the EM algorithm results in a set of parameters describing the line direction distributions, as well as a probabilistic assignment of individual vanishing points to each global line direction. After convergence, these results are fed back into the M-step of §5.5.1, thus completing the higher level E-step.

5.5.3 Construction and Initialization

Expectation maximization algorithms work only when properly initialized. The number of mixtures (in this case J, the number of 3-D line directions) must be known, and the algorithm must begin with a reasonable set of initial values (rotations and correspondence). If possible, prior distributions should also be supplied. This section outlines an algorithm that provides adequate initialization for the EM techniques described above.

Camera adjacency is first determined using the method of §4.4, which results in a graph whose nodes represent cameras and whose edges indicate proximal camera pairs. Next, the set of all adjacent camera pairs is extracted from the graph. The two-camera correspondence technique of §5.4.3 is then applied to each pair, and unique vanishing point matches are extracted.

A list of global line directions is constructed, each containing a set of references to its constituent vanishing points. The algorithm proceeds as follows:

```
Clear list of global line directions
For each camera pair in adjacency graph
    Apply two-camera VP correspondence
        For each VP pair matched
            If neither VP exists in any global line direction then
                Create new global line direction and add to list
                Link both constituent VPs to this new direction
            Else if one VP exists then
                Find its global line direction
                Link other VP to this direction
            Else if both VPs exist then
                If associated with different global line directions then
                    Merge two global line directions into one
```

This algorithm produces a list of vanishing point sets, each of which represents observations of a single scene-relative 3-D line direction. These sets are the components of the mixture model in §5.5.2, and correspondence weights can be initialized to binary values according to the grouping

produced above. Any camera not having at least two vanishing point entries in the list of line directions is tagged as unalignable, since at least two correspondences are needed for unique rotational registration. In practice, only about 5% of the cameras in a given configuration have insufficient vanishing points for alignment.

5.5.4 Merging and Separating Clusters

As the EM algorithm proceeds, separate vanishing point clusters that truly represent the same 3-D direction, or single clusters that represent multiple directions, may arise. The latter misclassifications can result from distinct 3-D lines having nearly identical directions that are fused due to noisy observations; the former usually results from the graph traversal described in the previous section.

After each rotation step of the EM algorithm, all pairs of cluster distributions are compared, and if two sufficiently overlap (e.g. with 95% probability) they are merged into one, decrementing J by one. Similarly, clusters containing vanishing points significantly different from their respective modal directions are split, incrementing J by one.

CHAPTER 6

Position Recovery

ESTIMATION OF STRUCTURE AND MOTION from image information comprises several tightly-coupled problems. Correspondence, camera pose, and scene structure all have nonlinear dependencies and are severely underconstrained without imposition of further *a priori* assumptions. The availability of accurate camera orientation from methods described in the preceding chapter greatly simplifies the problem; however, there still exists a strong coupling between correspondence and translational pose, which cannot be effectively separated. This chapter describes methods which complete the solution to the pose recovery problem by utilizing point features to estimate camera positions, but which do not require explicit correspondence between these features.

§6.1 describes the simplified epipolar geometry resulting from known rotational pose and discusses geometric constraints that may be used to reject physically unlikely point matches. Given a set of explicit matches within this framework, estimation of the direction of translation between a given pair of cameras reduces to a projective inference problem quite similar to that of single vanishing point estimation. §6.2 formulates the inference problem assuming known correspondence, then introduces the notion of probabilistic correspondence which circumvents the problem of explicit feature classification.

A high-level diagram of the algorithm is shown in Figure 6-1 below [AT00a]. First, translation directions are estimated between all relevant camera pairs in the data set. A Hough transform efficiently finds the most likely motion direction in a given pair by looking for consistency among all possible feature matches. This approximate direction serves to initialize an expectation maximization method whose E-step, which requires sampling from an extremely high-dimensional distribution, relies on a Markov Chain Monte-Carlo algorithm. This so-called *MCEM* algorithm, presented in §6.3, determines the best motion direction by averaging over all possible correspondence sets.

Once all relevant pair-wise motion directions have been computed, they are assembled into a global optimization that estimates the camera positions most consistent with these directions (§6.4). A final step, described in §6.5, performs rigid 3-D to 3-D registration on the resulting set of

cameras to find the best metric scale, position, and orientation given the approximate initial pose estimates.

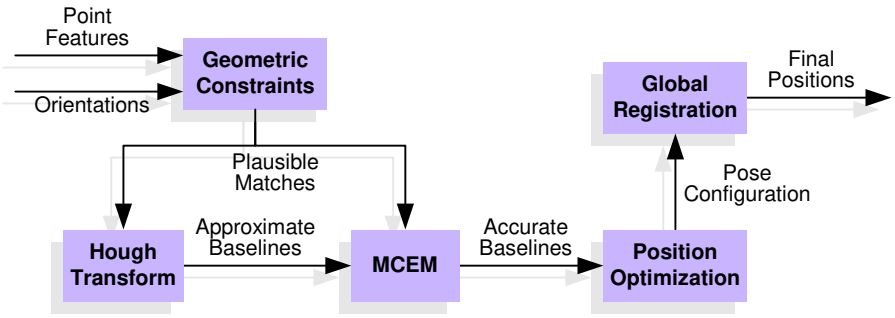


Figure 6-1: Translational Registration
Direction of motion between all adjacent camera pairs is determined without explicit feature correspondence. First, geometric constraints are imposed to reduce the number of possible point matches. Next, a Hough transform determines approximate baseline directions, which are then refined by a probabilistic technique. The pair-wise directions are assembled into a global optimization to find the most consistent camera positions. Finally, the resulting cameras are registered with the initial pose to recover metric scale, orientation, and position.

6.1 Two-Camera Translation Geometry

Given two rotationally registered cameras $\mathcal{A}$ and $\mathcal{B}$, and two sets of respective features $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ and $\mathcal{Y} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_M\}$, the goal is to determine the direction of motion $\boldsymbol{b}$ from $\mathcal{A}$ to $\mathcal{B}$ most consistent with the available data. This section describes geometric relationships that can be exploited to solve this problem.

The problem has been previously studied in various contexts. For short camera baselines, correspondence can be established by temporal feature tracking [SKS95]. Other short-baseline methods such as RANSAC [FZ98] attempt to solve both matching and motion problems simultaneously by choosing solutions most consistent with randomly sampled subsets of the data. Neither method employs a solid treatment of uncertainty, however, and both require explicit correspondence. Probabilistic formulations [Wel97, RCD99, DSTT00] are more appealing because they do not involve explicit correspondence.

6.1.1 Epipolar Geometry with Known Orientation

As introduced in §1.4.1, an epipolar plane $\mathcal{P}$ contains two camera centers and a 3-D point seen by both cameras. Projections of the 3-D point onto each of the images, $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$ respectively, must therefore also lie in $\mathcal{P}$ (see Figure 6-2).

For rotationally registered cameras, the following relation holds:

$$(\boldsymbol{x} \times \boldsymbol{y}) \cdot \boldsymbol{b} = 0. \tag{6-1}$$

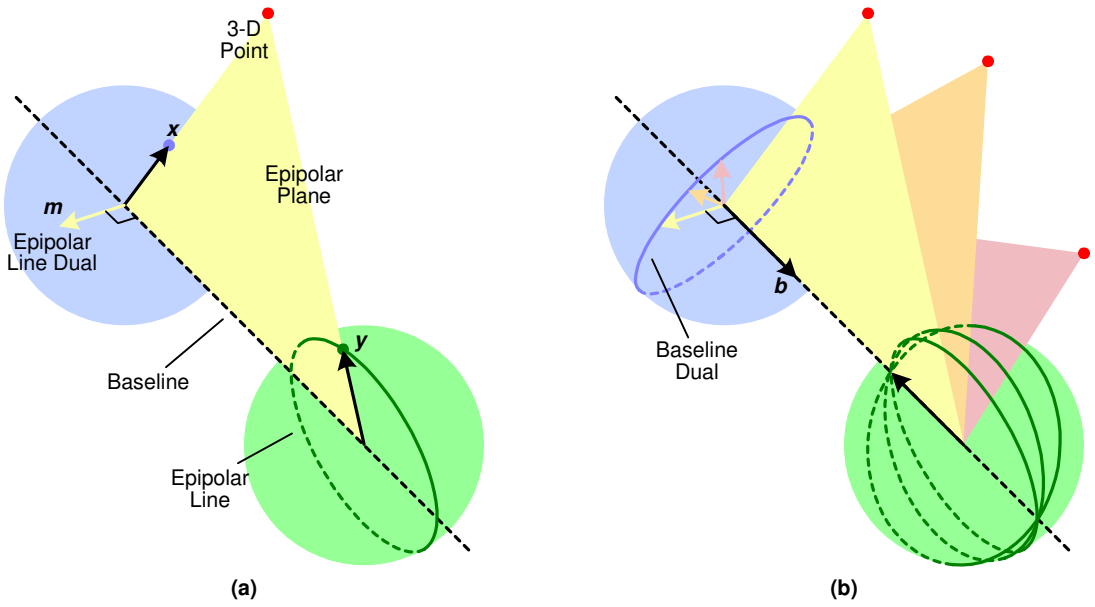


Figure 6-2: Pair Translation Geometry
The epipolar geometry for two rotationally aligned cameras is similar to the geometry of vanishing points. (a) A single 3-D point lies in an epipolar plane containing the baseline and any projective observations of the point. The epipolar line is analogous to an image line feature. (b) The epipolar planes induced by a set of 3-D points forms a pencil coincident with the baseline. The normals of these planes thus lie on a great circle orthogonal to the baseline direction.

Intuitively, the cross product of $\boldsymbol{x}$ with $\boldsymbol{y}$ is orthogonal to $\mathcal{P}$, and thus necessarily orthogonal to the baseline $\boldsymbol{b}$ as well, since $\mathcal{P}$ contains $\boldsymbol{b}$. Here, observations consist only of the 2-D feature projections, and the baseline is unknown; however, (6-1) provides a constraint on $\boldsymbol{b}$ up to unknown scale. This suggests that $\boldsymbol{b}$ can be inferred solely from two or more corresponding pairs of features.

Define $\boldsymbol{m}_{ij} \equiv \boldsymbol{x}_i \times \boldsymbol{y}_j$. For the correct pairs of i and j—that is, for those (i, j) couplets in which feature $\boldsymbol{x}_i$ truly matches feature $\boldsymbol{y}_j$—the constraint in (6-1) becomes

$$\boldsymbol{m}_{ij} \cdot \boldsymbol{b} = 0. \tag{6-2}$$

If the $\boldsymbol{m}_{ij}$ are viewed as projective epipolar lines, then the baseline $\boldsymbol{b}$ can be viewed as a projective *focus of expansion*, and its antipode the *focus of contraction*, the apparent intersections of all epipolar lines. This construction is analogous to that of vanishing points; estimation of the baseline, therefore, is identical to the estimation of vanishing points in the previous chapter, assuming correspondence is known.

6.1.2 Geometric Constraints on Correspondence

Both correspondence and the baseline are initially unknown, so the above construction seems hopelessly underconstrained. There are NM possible individual feature matches, and more importantly a combinatorial number of possible correspondence *sets* that can be chosen, making the search space enormous (see §6.3.2).

However, additional information can be used to drastically lower the dimension of the search space [QM88] both by reducing the number of features N and M in each image, and by eliminating many of the candidate correspondences. The constraints presented here rely on two assumptions: first, that each point feature represents the intersection of two or more 2-D line features; and second, that the baseline is known to lie within some restricted region. Bounds on this region can be obtained using the roughly-known initial pose.

Knowledge of 3-D line directions and classification of 2-D line features obtained from rotational pose recovery both provide strong cues for feature culling and point correspondence rejection. Presumably, objects consisting of parallel lines possess sufficient structure for determination of translational offsets; thus, image features not associated with any parallel line sets can be safely discarded. In particular, lines having high "outlier" probability (according to the mixture model in (5-6)), along with any point features inferred by these lines, are deemed invalid. Points inferred from lines shorter than a given threshold in Euclidean image space can also be discarded, as such lines are unreliable.

A set of all possible candidate matches is constructed from the surviving sets of point features. Each match is examined and kept or discarded according to the following criteria:

- ***Directions of constituent lines.*** If the 3-D point inferred by a given match truly corresponds to the intersection of two or more 3-D lines, then the 3-D directions of image lines forming a given image point $\boldsymbol{x}_i$ should be identical to those forming the point $\boldsymbol{y}_j$ (Figure 6-3). Matches $\boldsymbol{m}_{ij}$ for which this condition does not hold are discarded.

- ***Baseline uncertainty bound.*** A given angular bound on the translation direction induces a conservative equatorial band within which all correct epipolar plane normals must lie (Figure 6-4); any $\boldsymbol{m}_{ij}$ outside this band is discarded, since it implies "sideways" motion. Furthermore, any match for which $\boldsymbol{y}_j$ is closer than $\boldsymbol{x}_i$ to $\boldsymbol{b}$ is also discarded, as such a match implies "backward" motion.

- ***Depth of 3-D point.*** If the angle between $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$ exceeds a threshold, the 3-D point inferred by the match (via triangulation of the two feature rays) is too close to the camera or suggests an abnormally wide baseline; such matches are therefore rejected.

6.2 Inference of Translation Direction

This section describes methods for inferring the translation direction between a pair of cameras, first assuming explicit correspondence is known, then relaxing this assumption. As noted above, a given correspondence between features $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$ constrains the inter-camera baseline $\boldsymbol{b}$ according to (6-1), and a set of such correspondences can be used to estimate $\boldsymbol{b}$. One method is by minimization of an objective function such as

$$E = \sum_{(i,j)\in\mathcal{F}} \boldsymbol{m}_{ij} \cdot \boldsymbol{b}; \tag{6-3}$$

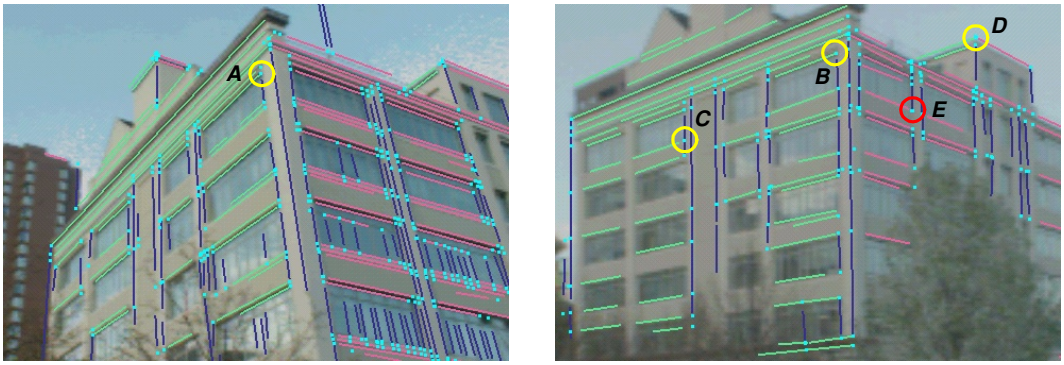


Figure 6-3: Line Constraints
Two images viewing the same building are shown, and possible matches for a particular point feature A in the first image are considered. Point B is the true match, but C and D are also plausible because they are formed by the intersection of lines whose directions match those of the lines forming A. The directions of the lines forming E do not match those forming A, so E is rejected. Note that D is formed by the intersection of three rather than two distinct line directions.

here, $\mathcal{F}$ is the set of F pairings (i, j) that represent the true matches. The optimal least-squares baseline direction can be found by constructing an $F \times 3$ matrix $\boldsymbol{A}$ whose rows contain the feature cross products $\boldsymbol{m}_{ij}$, then choosing the eigenvector associated with the smallest eigenvalue of $\boldsymbol{A}^\top \boldsymbol{A}$.

Projective fusion techniques as described in §3.5.3 can be used to estimate the probability density of $\boldsymbol{b}$. Recall from §1.3.1 that each point feature in the image represents the intersection of two image lines, each of which is an uncertain equatorially-distributed Bingham variable with known parameters. Bingham uncertainty in the intersection can be determined by using the cross product operator (3-36) introduced in §3.5.4, so that the parameters of each image point's distribution are known. Each correspondence between random variables $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$ induces an epipolar line $\boldsymbol{m}_{ij}$ via the cross product operator, so (3-35) can be used to find the Bingham uncertainty of $\boldsymbol{m}_{ij}$.

The problem that now remains is to determine the distribution of $\boldsymbol{b}$ given a set of observations $\boldsymbol{m}_{ij}$ (for $(i, j) \in \mathcal{F}$) with known uncertainty. The relationship in (6-3) implies that $\boldsymbol{b}$ represents an equatorial distribution, which can be estimated using techniques identical to those of vanishing point estimation (§5.3).

6.2.1 Motion Direction from Known Correspondence

If true correspondences between the feature sets $\mathcal{X}$ and $\mathcal{Y}$ are known, the baseline variable can be inferred according to the fusion equation

$$\boldsymbol{M}_{\boldsymbol{b}} = \sum_{(i,j)\in\mathcal{F}} \boldsymbol{M}_{ij} + \boldsymbol{M}_0 \tag{6-4}$$

where $\boldsymbol{M}_{ij}$ represents the uncertainty of the epipolar line $\boldsymbol{m}_{ij}$, $\boldsymbol{M}_0$ is the prior distribution on $\boldsymbol{b}$, and the sum is taken only over indices associated with the true matches. Equivalently, inference can be performed by associating a binary-valued variable b_{ij} with *every possible* correspondence;

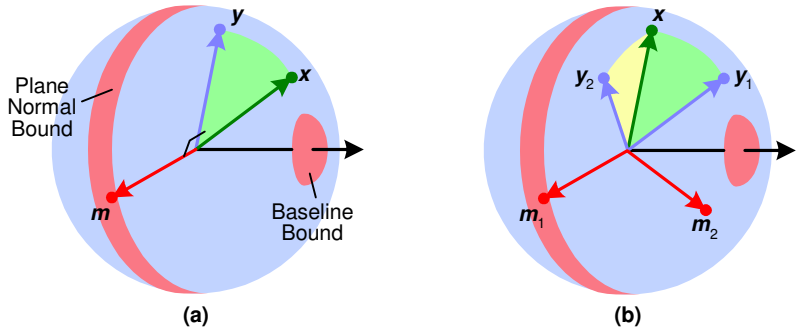


Figure 6-4: Direction Constraints
(a) Uncertainty in the baseline direction induces an equatorial band of uncertainty for epipolar lines. The match between features $\boldsymbol{x}$ and $\boldsymbol{y}$ is plausible because it implies motion in the correct direction. (b) The match between $\boldsymbol{x}$ and $\boldsymbol{y}_1$ is rejected because it implies backward motion; the match with $\boldsymbol{y}_2$ is also rejected because its epipolar line does not lie in the uncertainty band.

the variable is defined by

$$b_{ij} = \begin{cases} 1, & \text{if } \boldsymbol{x}_i \text{ matches } \boldsymbol{y}_j \\ 0, & \text{otherwise.} \end{cases} \tag{6-5}$$

The Bingham parameters of $\boldsymbol{b}$ can then be determined by

$$\boldsymbol{M}_{\boldsymbol{b}} = \sum_{i=1}^{M} \sum_{j=1}^{N} b_{ij} \boldsymbol{M}_{ij} + \boldsymbol{M}_0, \tag{6-6}$$

where the new sum is evaluated over every possible (i, j) pairing.

6.2.2 Motion Direction from Probabilistic Correspondence

Because motion directions and point features are uncertain quantities, and because of ambiguities in epipolar geometry that may arise from particular viewpoints, *hard* or *explicit* correspondence cannot always be determined (Figure 6-5). Thus, in the more general case, continuously-valued variables $w_{ij} \in [0, 1]$, rather than binary-valued variables $b_{ij} \in \{0, 1\}$, can be applied to the observations $\boldsymbol{m}_{ij}$, effectively representing the probability that feature $\boldsymbol{x}_i$ matches feature $\boldsymbol{y}_j$.

Inference of $\boldsymbol{b}$ in this weighted formulation becomes

$$\boldsymbol{M}_{\boldsymbol{b}} = \sum_{i=1}^{M} \sum_{j=1}^{N} w_{ij} \boldsymbol{M}_{ij} + \boldsymbol{M}_0, \tag{6-7}$$

with more emphasis given to matches with higher likelihood. Note that the binary variables b_{ij} represent the deterministic limit of the w_{ij} in this probabilitic formulation.

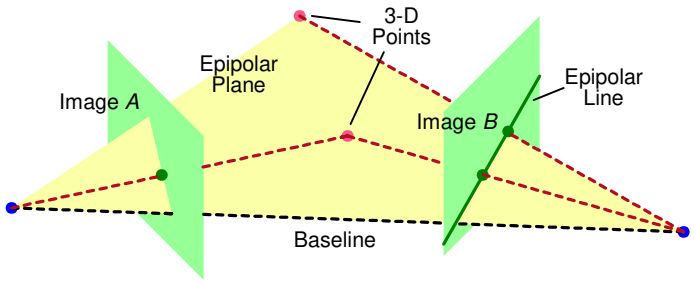


Figure 6-5: Correspondence Ambiguity
It is not always geometrically possible to determine one-to-one correspondence. For instance, if a given point feature in image A lies on the same epipolar plane as two distinct 3-D points, then its match in image B is inherently ambiguous.

6.2.3 Weight Variables

In reality, each feature observed in one image has at most one true match in the other image. A true match exists only if the feature observation corresponds to a 3-D point, and if its counterpart in the other image is visible; otherwise, the feature has *no* matches—either it is itself spurious, or its match is occluded or otherwise missed by detection.

In the case of binary variables, the above condition can be enforced by requiring that at most one b_{ij} for every i and at most one b_{ij} for every j is equal to one, and that the rest are equal to zero. More formally,

$$\begin{aligned} \sum_{j=1}^{N} b_{ij} &\leq 1 \qquad \forall i \\ \sum_{i=1}^{M} b_{ij} &\leq 1 \qquad \forall j. \end{aligned} \tag{6-8}$$

Inequality constraints are mathematically inconvenient, however; thus, the "null" features $\boldsymbol{x}_0$ and $\boldsymbol{y}_0$ are appended to $\mathcal{X}$ and $\mathcal{Y}$, respectively, and the inequality constraints of (6-8) become equality constraints via the introduction of binary-valued *slack variables* b_{i0} and b_{0j} [CR00b], which take value one if $\boldsymbol{x}_i$ (or $\boldsymbol{y}_j$, respectively) matches no other feature, and zero otherwise. Thus,

$$\begin{aligned} \sum_{j=0}^{N} b_{ij} &= 1 \quad \forall i \in \{1, \ldots, M\} \\ \sum_{i=0}^{M} b_{ij} &= 1 \quad \forall j \in \{1, \ldots, N\}. \end{aligned} \tag{6-9}$$

To ensure valid weights w_{ij} in the probabilistic case, an analogous condition must be satisfied:

$$\begin{aligned} \sum_{j=0}^{N} w_{ij} &= 1 \quad \forall i \in \{1, \ldots, M\} \\ \sum_{i=0}^{M} w_{ij} &= 1 \quad \forall j \in \{1, \ldots, N\}. \end{aligned} \tag{6-10}$$

This condition enforces a symmetric (two-way) distribution over all correspondences: each feature in the first image can match a set of possible features in the second image, with the weights normalized so that they sum to one, and vice versa.

The set of weights can also be represented by an $(M+1) \times (N+1)$ matrix $\boldsymbol{W}$ (or $\boldsymbol{B}$, in the binary case), whose rows represent the features $\mathcal{X}$, whose columns represent the features $\mathcal{Y}$, and whose individual entries are the weights themselves (Figure 6-6). The condition in (6-10) is then equivalent to the requirement that the weight matrix be *doubly stochastic*, i.e. that both its rows and its columns sum to one.

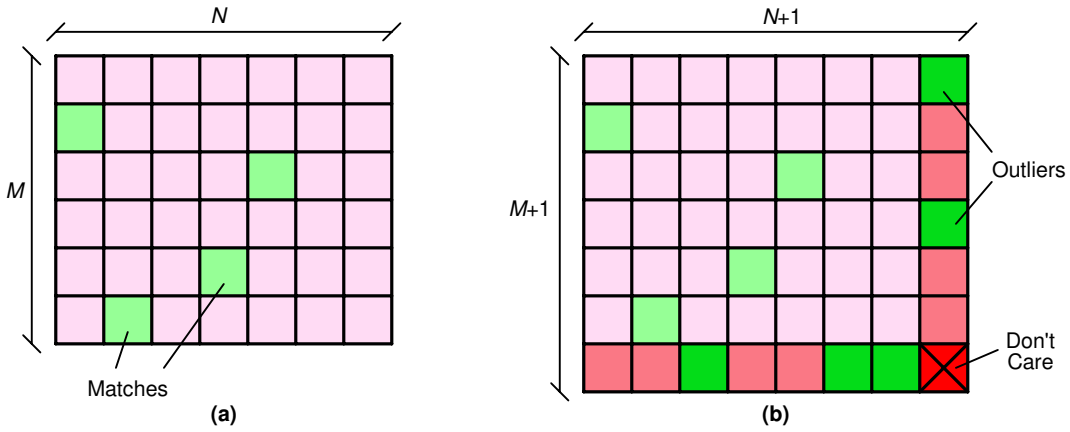


Figure 6-6: Augmented Match Matrix
The match matrix encodes correspondences between features in two different images. (a) An example of a binary match matrix. Rows represent features in the first camera, and columns represent features in the second. There can be at most one entry per row and per column. (b) The matrix is augmented with an extra row and column to account for outliers and missing features. In the augmented matrix, there must be exactly one entry in each row and in each column.

6.2.4 Obtaining a Prior Distribution

Because motion direction and correspondence are tightly coupled, it is difficult to determine these quantities without prior information. However, as this section will demonstrate, utilization of initial pose estimates and the geometric constraints introduced in §6.1.2 allows for reasonably accurate estimates of $\boldsymbol{b}$ to be obtained *without* knowledge of correspondence.

The constraint equations in (6-1) are identical in form to those that govern the estimation of a vanishing point from observations on its equatorial band. Here, arguments analogous to those

in §5.3 are presented for estimation of $\boldsymbol{b}$ from unclassified correspondences. Let $\mathcal{M}$ represent the set of *all* plausible correspondences (epipolar lines) between $\mathcal{X}$ and $\mathcal{Y}$, and let the special subset $\mathcal{M}' \in \mathcal{M}$ contain only the F true matches. If all lines in $\mathcal{M}$ are drawn on $\mathbb{S}^2$, those in $\mathcal{M}'$ (in the absence of noise) will intersect perfectly at the motion direction $\boldsymbol{b}$, and the remainder (which represent false matches) will intersect at random points on the sphere.

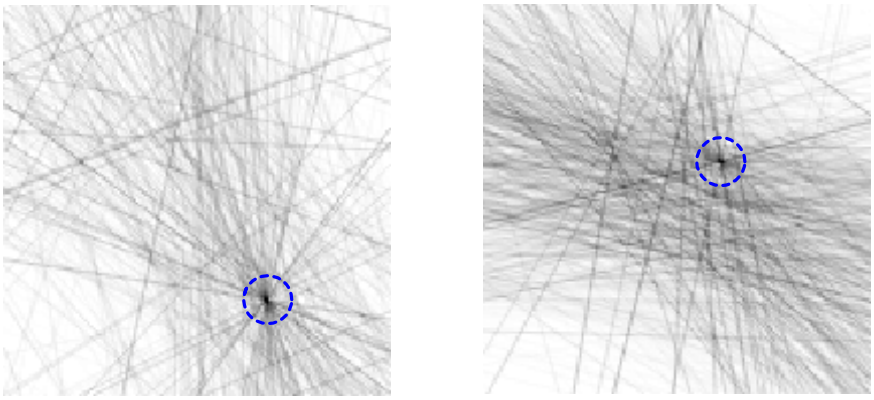


Figure 6-7: Hough Transform for Baseline Estimation
Two examples of Hough transforms for baseline estimation. Epipolar lines for all plausible matches are accumulated; the transform peak represents the baseline direction.

In essence, the point of maximum incidence on $\mathbb{S}^2$ is the most likely direction of motion. This point can be found by discretizing $\mathbb{S}^2$ and accumulating all candidate epipolar lines $\mathcal{M}$ in a Hough transform, as described in §4.1 and §5.3.4. Since the motion direction is approximately known from initial pose estimates, the transform need only be formulated over a small portion of the sphere's surface around this initial direction. Examples are shown in Figure 6-7.

Using the methods of §4.1.7, the motion direction $\boldsymbol{b}_0$ can be determined as the peak with highest magnitude. False correspondences greatly outnumber true correspondences, however, because there are MN possible matches and only F (at most $\min(M, N)$) true matches. The desired peak may therefore be obscured by spurious peaks arising from certain geometric anomalies. For example, a point feature in one image lying very close to the initial direction of motion can match many features in the other image, thus producing a perfectly sharp false peak if all matches are equally weighted (Figure 6-8).

To solve this problem, a mutually consistent set of weights w_{ij} must be assigned to the epipolar lines in $\mathcal{M}$ such that features having many possible matches are de-emphasized. In order to ensure that the normalization condition in (6-10) is satisfied, an iterative normalization procedure proposed by Sinkhorn [Sin64, Sin67] is utilized to transform an initial (invalid) match matrix into a valid doubly stochastic matrix.

First, the matrix $\boldsymbol{W}$ is set to zero; entries for matches satisfying the geometric constraints of §6.1.2, as well as all entries in row $M + 1$ and column $N + 1$, are then assigned an initial value of one. Sinkhorn's algorithm alternatively normalizes the rows and columns until convergence as

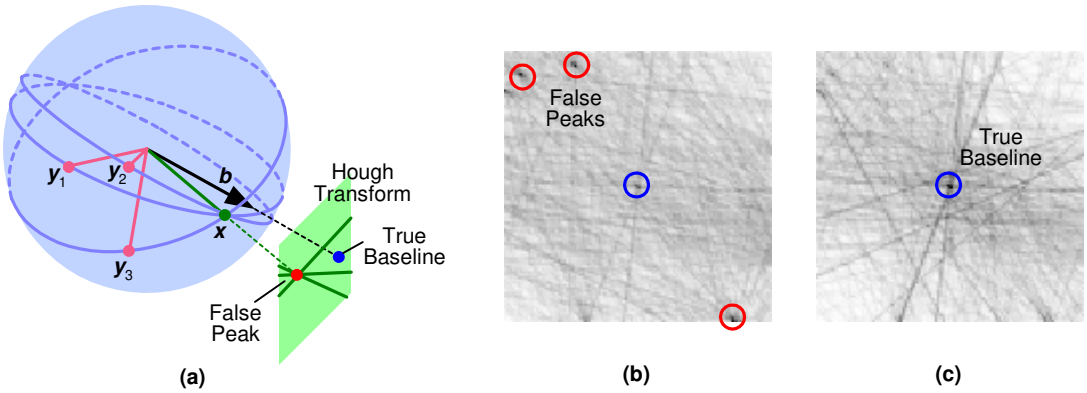


Figure 6-8: False Hough Transform Peaks
(a) False peaks in the Hough transform can be caused by features too close to the direction of motion, which have many matches and thus produce high-incidence regions. (b) An example in which false peaks are evident. (c) The same example after normalization.

follows:

$$w'_{ij} = w_{ij} \Big/ \sum_{j=0}^{N} w_{ij} \quad \forall i \in \{1, \ldots, M\}$$
$$w''_{ij} = w'_{ij} \Big/ \sum_{i=1}^{M} w'_{ij} \quad \forall j \in \{1, \ldots, N\}.$$

Each entry in the matrix is normalized by the sum of entries in its row; each entry in the resulting matrix is then normalized by the sum of entries in its column, and so on. The algorithm produces a provably *unique* factorization $\boldsymbol{W}' = \boldsymbol{D}_1 \boldsymbol{W} \boldsymbol{D}_2$ [Sin64], such that $\boldsymbol{W}'$ is doubly stochastic. The new matrix does not represent the "correct" distribution, because it is somewhat arbitrarily initialized, but it provides a practical approximation for the purposes of the Hough transform technique described above.

Recall from §4.1.5 that, for a planar accumulation space, each linear constraint of the form in (6-1) contributes a single straight line to the transform. Thus, once a set of weights has been obtained by the above method, the epipolar lines are accumulated, and when drawn, their accumulation values are weighted by the appropriate value w_{ij}. This normalization to a valid probability distribution over correspondences dramatically improves the coherence of the true motion direction (Figure 6-8c).

Although accuracy is inherently limited by the discrete nature of the Hough transform, the resulting motion direction estimate $\boldsymbol{b}_0$ can be used to initialize more accurate techniques. Further, it can be used as a strong prior distribution (with parameters $\boldsymbol{M}_0$ in the notation of §6.2) in subsequent inference tasks. The matrix $\boldsymbol{M}_0$ is obtained by using the bipolar approximation in §4.1.8, then computing the dual distribution.

6.3 Monte Carlo Expectation Maximization

In general, true feature correspondence is completely unknown; feature point measurements and uncertainty serve as the only available information for inference of motion. This section outlines a method for determining accurate motion estimates from this information alone, *without* requiring explicit correspondence, by employing an expectation algorithm in which the posterior distribution is discretely sampled.

Using maximum likelihood notation,

$$\boldsymbol{b}^* = \text{argmax}_{\boldsymbol{b}}[p(\boldsymbol{b}|\mathcal{M})] \tag{6-11}$$

The conditional probability above can be expanded using Bayesian inference techniques discussed in §3.2.2:

$$\begin{aligned} p(\boldsymbol{b}|\mathcal{M}) &= \sum_{\boldsymbol{B}} p(\boldsymbol{b}, \boldsymbol{B}|\mathcal{M}) \\ &= \sum_{\boldsymbol{B}} p(\boldsymbol{b}|\boldsymbol{B}, \mathcal{M}) p(\boldsymbol{B}|\mathcal{M}) \end{aligned} \tag{6-12}$$

where $\boldsymbol{B}$ is a valid binary-valued correspondence matrix, and $p(\boldsymbol{B}|\mathcal{M})$ is the prior distribution on the correspondence set. This prior distribution is assumed to be uniform, but can equally well incorporate the geometric match constraints of §6.1.2. Note also that the likelihood is expressed as a summation rather than an integration, because the collection of all possible correspondence sets is discrete.

6.3.1 Structure from Motion without Correspondence

The expression in (6-11) suggests that the optimal estimate of the motion direction $\boldsymbol{b}$ can be found *without using explicit correspondence*, by maximizing $p(\boldsymbol{b}|\mathcal{M})$ alone [DSTT00]. Correspondence sets can be treated as nuisance parameters in a Bayesian formulation, as illustrated by (6-12), in which the likelihood is evaluated over all possible matrices $\boldsymbol{B}$.

The expectation maximization algorithm lends itself well to this type of optimization problem. The expression to be maximized in (6-12) is analogous to the mixture model of (4-14); $\boldsymbol{b}$ is formed by the aggregate contributions of a combinatorial number of mixture components, each of which represents a valid correspondence set $\boldsymbol{B}$. The algorithm alternates between the M-step, in which a log likelihood function is maximized given a posterior likelihood, and the E-step, in which the likelihood function is evaluated given the current parameter estimate $\boldsymbol{b}$. Unlike in [DSTT00], convergence to the optimal solution is guaranteed because of the initial estimate provided by the Hough transform approach above.

The log likelihood to be maximized is

$$L = \sum_{\boldsymbol{B}} p(\boldsymbol{B}|\boldsymbol{b}, \mathcal{M}) \log p(\boldsymbol{b}|\boldsymbol{B}, \mathcal{M}). \tag{6-13}$$

Substitution of (6-6) into (6-13) gives

$$L \propto \sum_{\boldsymbol{B}} p(\boldsymbol{B}|\boldsymbol{b}, \mathcal{M}) \sum_{i=1}^{M} \sum_{j=1}^{N} b_{ij} \boldsymbol{b}^\top \boldsymbol{M}_{ij} \boldsymbol{b} + \boldsymbol{b}^\top \boldsymbol{M}_0 \boldsymbol{b} \tag{6-14}$$

Now, define w_{ij} as the marginal posterior probability of match b_{ij}, regardless of the other matches; that is,

$$w_{ij} \equiv p(b_{ij} = 1|\boldsymbol{b}, \mathcal{M}) = \sum_{\boldsymbol{B}} \delta(i,j) p(\boldsymbol{B}|\boldsymbol{b}, \mathcal{M}); \tag{6-15}$$

then (6-14) becomes

$$\sum_{i=1}^{M} \sum_{j=1}^{N} w_{ij} \boldsymbol{b}^\top \boldsymbol{M}_{ij} \boldsymbol{b} + \boldsymbol{b}^\top \boldsymbol{M}_0 \boldsymbol{b}. \tag{6-16}$$

Maximization of (6-16), given the set of weights w_{ij}, can be easily performed using the technique described in §6.2.2; however, determination of the w_{ij} is not so straightforward. Individual matches are not mutually independent, because knowledge about one match provides knowledge about others. For example, given that $b_{ij} = 1$, it must be true that

$$b_{ik} = 0 \qquad \forall k \neq j, \tag{6-17}$$

which follows from the implicit constraints in (6-9). The condition for independence in (3-8) is thus violated, because

$$p(b_{ik} = 1|b_{ij} = 1, \boldsymbol{b}, \mathcal{M}) = 0 \neq p(b_{ik} = 1|\boldsymbol{b}, \mathcal{M}), \tag{6-18}$$

and the joint likelihood $p(\boldsymbol{B}|\boldsymbol{b}, \mathcal{M})$ cannot be factored. Precise evaluation of (6-16) apparently requires evaluation of (6-14), a difficult task due to the combinatorial number of terms. However, the following sections will show that the w_{ij} can be evaluated efficiently by Monte Carlo sampling.

6.3.2 Counting Correspondence Sets

Although not crucial to solving the optimization problem, it is interesting to illustrate the enormity of the sample space by counting the precise number of possible correspondence sets. Assume for the moment that the number of valid matches F is known. A particular correspondence set is represented by an $M \times N$ binary matrix $\boldsymbol{B}$ containing at most one non-zero entry per row and per column. A non-zero entry in position (i, j) represents a particular correspondence between feature $\boldsymbol{x}_i$ and feature $\boldsymbol{y}_j$.

Initially, $\boldsymbol{B}$ contains all zeros. F correspondences are then placed in the matrix one at a time, always preserving the condition in (6-9). The first correspondence to be placed has MN possible positions (i.e. anywhere in the matrix). The second, however, has only $(M-1)(N-1)$ possibilities, because it cannot be placed in the same row or column as the initial correspondence; the third has $(M-2)(N-2)$ possibilities, and so on. Thus, for a given number of correspondences F, the number of possible match sets S_F is given by

$$\begin{aligned} S_F F! &= (M)(N)(M-1)(N-1)\cdots(M-F+1)(N-F+1) \\ &= (M)(M-1)\cdots(M-F+1)(N)(N-1)\cdots(N-F+1) \\ &= \binom{M}{F}\binom{N}{F}(F!)^2 \end{aligned} \tag{6-19}$$

Here S_F has been multiplied by $F!$ to account for permutations of the chosen matches; the order in which matches are placed is unimportant.

The number of correspondences F can itself vary. The total number of possible match sets S is thus given by a sum over all values of F,

$$\begin{aligned} S &= \sum_{F=0}^{F'} S_F \\ &= \sum_{F=0}^{F'} \binom{M}{F} \binom{N}{F} F!, \end{aligned} \tag{6-20}$$

where $F' = \min(M, N)$. For an image pair observing only 20 features (such that $M = N = 20$), S is on the order of 10^{21}.

6.3.3 Previous Expectation Maximization Methods

Various authors have used the EM algorithm to solve similarly formulated structure from motion problems. For example, [Wel97] uses EM for object recognition tasks in which features in a new image are to be correlated with features in a template image via a one-way matching function. Wells makes the simplifying assumption that all correspondences are independent, and implements an iterative algorithm that alternates between finding the best motion given a set of probabilistic correspondence weights, and finding the best weights given the current motion estimate.

So-called *iterated closest point* (*ICP*) algorithms very similar to EM have also been proposed [BM92]. Rather than using continuous probabilities, however, these algorithms use binary weights b_{ij}, making an explicit choice of the "closest" matches given the current motion estimate. ICP algorithms suffer from boundary problems not unlike the k-means algorithm (see §4.2) because they do not properly treat uncertainty, nor do they handle the implicit match dependencies of §6.2.3.

Rangarajan and Chui use EM for rigid and non-rigid point matching across two cameras [CR00a], but the formulation lacks the rigor of that presented in §6.3.1. Individual likelihoods analogous to the marginal probabilities w_{ij} are computed as though the individual correspondences are mutually independent, and Sinkhorn's algorithm is then used to normalize these weights to satisfy (6-10). The weights so obtained do not represent the true distribution, and are sensitive to initialization.

The EM formulation proposed by Dellaert is the most theoretically sound, and in some ways the most general [DSTT00]; it applies to multi-camera structure from motion and can be used even when nothing is initially known about camera motion or scene geometry. The techniques are restricted to the case where all features are visible in all images and the number of visible 3-D points is known; in other words, outlier features and occlusion are not handled. Nevertheless, the formulation is sound, and forms the basis for the algorithm presented in the next section.

6.3.4 Sampling the Posterior Distribution

As described in §4.3, Markov chain Monte Carlo algorithms are useful for evaluating sums of the form in (6-14). In this context, the state at time k corresponds to a valid binary match matrix $\boldsymbol{B}^k$.

The annealing algorithm can be summarized as follows:

Start with initial temperature $T = T_0$
Loop until $T \leq 1$ (E-step):
  Set $k = 0$
  Start with valid state $\boldsymbol{B}^0$
  Compute initial parameter matrix $\boldsymbol{M}^0$
  Compute initial likelihood coefficient $c(\boldsymbol{M}^0)$
  Set $\boldsymbol{A} = \boldsymbol{0}$
  Loop until k sufficiently high (steady state):
    Randomly perturb state to $\tilde{\boldsymbol{B}}^k$
    Evaluate the likelihood ratio β
    If $\beta \geq 1$ then keep new state
    Else keep new state with probability $\beta^{1/T}$
    If new state kept then
      Set $\boldsymbol{B}^{k+1} = \tilde{\boldsymbol{B}}^k$
      Compute $\boldsymbol{M}^{k+1}$ and $c(\boldsymbol{M}^{k+1})$
    Else set $\boldsymbol{B}^{k+1} = \boldsymbol{B}^k$
    Set $\boldsymbol{A} = \boldsymbol{A} + \boldsymbol{B}^{k+1}$
    Set $k = k + 1$
  Set $\boldsymbol{W} = \boldsymbol{A}/k$
  Solve for new $\boldsymbol{b}$ given $\boldsymbol{W}$ (M-step)
  Set $T = \alpha T$ (for $0 < \alpha < 1$)
  Set $n = n + 1$

In a particular E-step loop, $\boldsymbol{A}$ is an $(M+1) \times (N+1)$ accumulation matrix that counts the number of visits to each state. $\boldsymbol{W}$ is a valid matrix of marginal probabilities (weights) w_{ij} obtained by averaging all state visits, analogous to (4-25). The initial temperature T_0 is set to a relatively low value; high initial temperatures serve to explore larger regions of the parameter space, which is unnecessary because the Hough transform provides a reasonably accurate initial estimate $\boldsymbol{b}_0$. The value of T_0 is chosen according to uncertainty bounds on the initial estimate $\boldsymbol{b}_0$, and is typically between 1.5 and 2.0 in practice. The likelihood function used to compute the ratio β is

$$\begin{aligned} p(\boldsymbol{B}^k|\boldsymbol{b},\mathcal{M}) &= c(\boldsymbol{M}^k)\exp\left[\boldsymbol{b}^\top \boldsymbol{M}^k \boldsymbol{b}\right] \\ &= c(\boldsymbol{M}^k)\exp\left[\boldsymbol{b}^\top \sum_{i=1}^{M}\sum_{j=1}^{N} b_{ij}\boldsymbol{M}_{ij}\boldsymbol{b}\right] \end{aligned} \tag{6-21}$$

where $\boldsymbol{b}$ is taken as the modal direction of the current baseline distribution estimate. Efficient calculation of β is described in §6.3.6.

The MCMC algorithm requires a valid starting state, and random state perturbations that satisfy detailed balance (meaning effectively that every valid state is reachable from every other valid state). Thus perturbations must be defined which can visit the entire state space. These perturbations are described in the following sections.

6.3.5 Match Perturbations

For the case where $\boldsymbol{B}^k$ is a square permutation matrix (i.e. all features are visible in all images), Dellaert proposes simple swap perturbations, so that $\boldsymbol{B}^{k+1}$ is identical to $\boldsymbol{B}^k$ except for a single row (or, equivalently, column) swap. It can be proven that all states are reachable using these perturbations. When the number of visible 3-D features is unknown, however, and when outliers and occlusion are present, detailed balance is no longer satisfied by simple match swapping, because such swapping preserves the number of valid matches; therefore, states with greater or fewer matches than the current state are never reached.

This thesis generalizes Dellaert's technique, in the two-camera case, to handle an unknown number of visible 3-D features, and also to handle outliers and occlusion. The state matrix $\boldsymbol{B}$ and the probability matrix $\boldsymbol{W}$ are each augmented with an extra row and column (§6.2.3) to account for features that have no matches. Novel perturbations in addition to row and columns swaps are also introduced which allow all states to be visited.

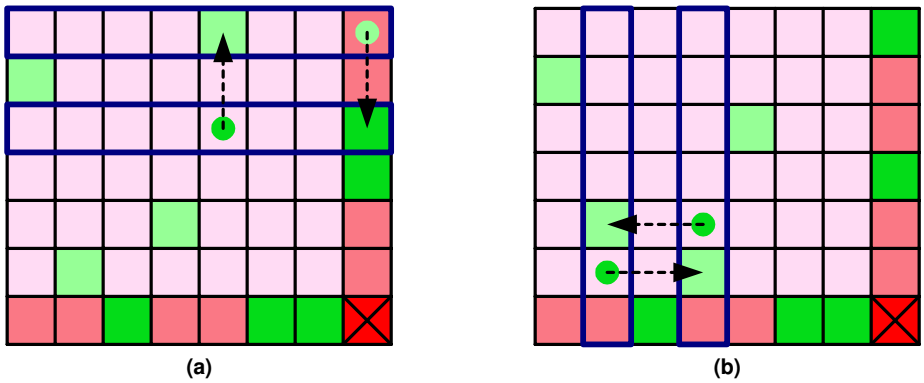


Figure 6-9: Row and Column Swaps
(a) Two rows of the match matrix, including outliers, are interchanged. (b) Two columns are interchanged.

In particular, to allow the number of valid matches to change, two complementary operations are proposed. The *split* perturbation converts a valid match into two outlier features, and the *merge* perturbation joins two outlier features into one valid match. Figure 6-10 depicts these operations in terms of the correspondence matrix $\boldsymbol{B}$.

6.3.6 Efficient Sampling

The sampling algorithm outlined above seems at first to be computationally expensive, especially for the large state matrices typical of real images containing many features. However, three optimizations can be applied to significantly improve the algorithm's performance. The majority of any given state matrix is zero; in fact, out of MN possible entries, a maximum of $N + M - 1$ are non-zero (this corresponds to the case where all features are outliers). Thus the first optimization is to use sparse matrix representations for $\boldsymbol{B}$ and for state perturbations. Because of the geometric match constraints from §6.1.2, many configurations $\boldsymbol{B}$ are invalid. Thus, the second optimization is to consider only those state perturbations involving valid matches.

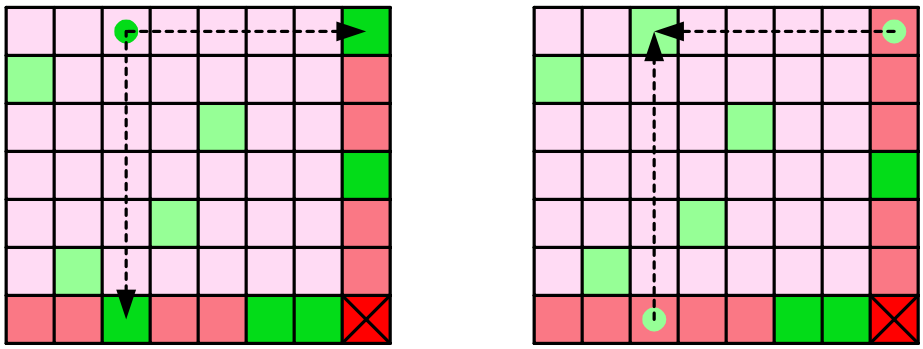


Figure 6-10: Split and Merge Perturbations
(a) A valid correspondence is split into two outliers, thus reducing the number of valid matches by one. (b) Any two outliers can be merged into a valid correspondence to increase the number of valid matches by one.

The final optimization involves computation of the likelihood ratios β. Each perturbation represents only an incremental change in the state that involves at most four entries in $\boldsymbol{B}$. The exponential form of the likelihood function in (6-21) facilitates computation of ratios:

$$\begin{aligned}\beta &= \frac{p(\tilde{\boldsymbol{B}}^k|\boldsymbol{b},\mathcal{M})}{p(\boldsymbol{B}^k|\boldsymbol{b},\mathcal{M})} \\ &= \frac{c(\tilde{\boldsymbol{M}}^k)\exp\left[\boldsymbol{b}^\top\tilde{\boldsymbol{M}}^k\boldsymbol{b}\right]}{c(\boldsymbol{M}^k)\exp\left[\boldsymbol{b}^\top\boldsymbol{M}^k\boldsymbol{b}\right]} \\ &= \frac{c(\tilde{\boldsymbol{M}}^k)}{c(\boldsymbol{M}^k)}\exp\left[\boldsymbol{b}^\top(\tilde{\boldsymbol{M}}^k-\boldsymbol{M}^k)\boldsymbol{b}\right]. \end{aligned} \tag{6-22}$$

In the case of swapping two rows, say row m which contains a one in column n and row p which contains a one in column q, most terms in the sum of (6-21) remain unchanged; only the entries b_{mn}, b_{pq}, b_{mq}, and b_{pn} differ. The new parameter matrix is given by

$$\tilde{\boldsymbol{M}}^k = \boldsymbol{M}^k + \boldsymbol{M}_{mq} + \boldsymbol{M}_{pn} - \boldsymbol{M}_{mn} - \boldsymbol{M}_{pq}, \tag{6-23}$$

which involves only four new terms that can be computed from the current parameter matrix.

Split and merge perturbations have equally simple incremental computations, since they also involve only a few entries of $\boldsymbol{B}^k$. If a valid correspondence b_{mn} is split, then the new parameter matrix becomes

$$\tilde{\boldsymbol{M}}^k = \boldsymbol{M}^k - \boldsymbol{M}_{mn}; \tag{6-24}$$

if two outliers are merged, the new parameter matrix is

$$\tilde{\boldsymbol{M}}^k = \boldsymbol{M}^k + \boldsymbol{M}_{mn}. \tag{6-25}$$

Incremental computation of the difference $(\tilde{\boldsymbol{M}}^k - \boldsymbol{M}^k)$ in (6-22) is thus quite straightforward.

6.3.7 Comparison to Consensus Sampling

Some algorithms, such as RANSAC [HZ00] and LMS [CC91], attempt to find the most consistent motion by randomly choosing minimal subsets of F points per image and examining the consistency of the remaining features. Assume here that $M = N$; then by the arguments of §6.3.2, there are $\mathcal{O}(N^{2F})$ such subsets, of which at most $\mathcal{O}(N^F)$ are completely self-consistent (i.e. represent true feature matches). The probability of choosing a "correct" subset is then $\mathcal{O}(N^{-F})$, decreasing significantly as the number of features increases and requiring an expected $\mathcal{O}(N^F)$ samples for correct convergence (in practice, a much smaller number of samples is used). A fundamental matrix is computed for each putative subset that summarizes the implied epipolar geometry; these algorithms must then evaluate the consistency of this geometry by comparing each feature in one image to each feature in the other, requiring $\mathcal{O}(N^2)$ computation. Typically, sets of $F = 6$ points are used; for $N = 20$, then, the number of operations required to guarantee convergence is on the order of 10^{11}.

Such sampling methods do not traverse the correspondence space in a principled way; instead, they choose features completely at random. In addition, they consider correspondences independently when testing consistency rather than as sets, and thus cannot account for implicit constraints, match ambiguities, or feature noise.

By contrast, the techniques presented in this chapter require $\mathcal{O}(N^2)$ computation. The space of features and correspondences is significantly decreased by strong geometric constraints, and sampling can be reduced to linear order through the use of sparse matrix routines, as mentioned in §6.3.6. The correspondence space is also sampled in a structured manner according to geometric constraints and aggregate match likelihoods, so that regions of high likelihood are found quickly and efficiently.

6.4 Multi-Camera Method

Translations recovered between camera pairs are merely directions, and thus can only be determined up to unknown scale. This section illustrates how these directions can be assembled into a set of constraints on the camera positions and used to recover a globally self-consistent pose configuration.

6.4.1 Constraint Equations

Let $\boldsymbol{p}_i$ represent the unknown position of a given camera, which also has a set of neighbors determined by the adjacency methods of §4.4 at unknown positions $\boldsymbol{p}_j$. The baseline *direction* $\boldsymbol{b}_{ij}$ to each neighbor is known, determined by the pair-wise alignment described in §6.3; however, the *distance* α_{ij} to each neighbor along this direction is unknown. This suggests a set of linear vector constraint equations of the form

$$\boldsymbol{p}_j = \boldsymbol{p}_i + \alpha_{ij}\boldsymbol{b}_{ij}. \tag{6-26}$$

The above equation states simply that the position of camera j can be obtained by starting at camera i and traveling a distance α_{ij} along the inter-camera direction $\boldsymbol{b}_{ij}$. By convention, $\alpha_{ij} > 0$, so $\alpha_{ij} = \alpha_{ji}$ and $\boldsymbol{b}_{ij} = -\boldsymbol{b}_{ji}$.

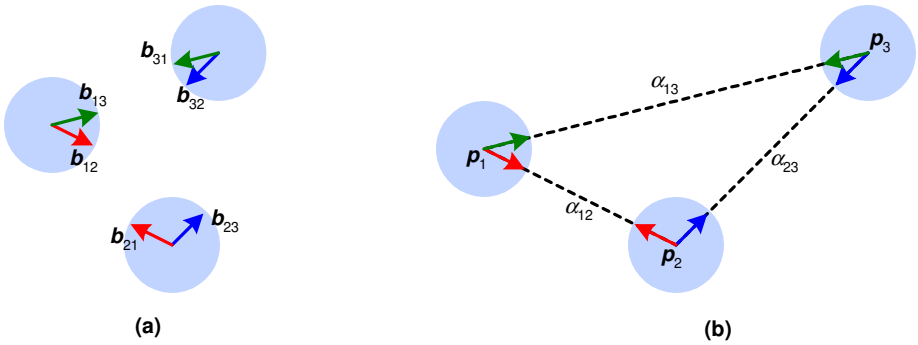


Figure 6-11: Assembling Translation Directions
(a) After motion directions are estimated between all relevant camera pairs, camera positions are still unknown. (b) A pose configuration consistent with all motion directions can be determined.

Let N be the number of unknown camera positions to be determined, each of which has three degrees of freedom, and let M be the number of equations of the form (6-26), each of which provides three constraints, but also introduces one unknown α_{ij}. There are then a total of $3N + M$ unknown quantities and $3M$ constraints on these quantities, and it would seem from these numbers alone that a unique solution requires $\frac{3}{2}N$ motion directions. The baseline geometry and underlying topological structure, however, can make determination of a unique solution more complicated.

Any globally-consistent camera configuration retains its consistency under Euclidean transformations. Rotation modifies the constraint equations (6-26), since they depend on scene-relative directions; however, scale and translation of the entire configuration do not affect the constraints. To illustrate this, let $\tilde{\boldsymbol{p}}_i = s\boldsymbol{R}\boldsymbol{p}_i + \boldsymbol{t}$ for any camera position $\boldsymbol{p}_i$, where s is an isotropic scale factor, $\boldsymbol{R}$ is a rotation matrix, and $\boldsymbol{t}$ is a translation vector. Then the constraint equation becomes

$$\begin{aligned} \tilde{\boldsymbol{p}}_j &= \tilde{\boldsymbol{p}}_i + \alpha_{ij}\boldsymbol{b}_{ij} \\ s\boldsymbol{R}\boldsymbol{p}_j + \boldsymbol{t} &= s\boldsymbol{R}\boldsymbol{p}_i + \boldsymbol{t} + \alpha_{ij}\boldsymbol{b}_{ij} \\ \boldsymbol{p}_j &= \boldsymbol{p}_i + \frac{\alpha_{ij}}{s}(\boldsymbol{R}^\top \boldsymbol{b}_{ij}) \\ \boldsymbol{p}_j &= \boldsymbol{p}_i + \tilde{\alpha}_{ij}\tilde{\boldsymbol{b}}_{ij} \end{aligned}$$

equivalent in form to (6-26). If $\boldsymbol{R} = \boldsymbol{I}$, meaning that no rotation occurs, then $\boldsymbol{b}_{ij} = \tilde{\boldsymbol{b}}_{ij}$ and the only change is that the unknown distance values that would have been obtained by solution of (6-26) are scaled down by s. Thus, an arbitrary global origin and scale can be applied to any solution configuration regardless of the number of known directions, inherently underconstraining the system of equations (Figure 6-12b).

The topology of the camera adjacency graph bears directly on degeneracy of the constraints in (6-26), as shown in Figure 6-12a. A *sufficient* condition for unique solution of these constraints (up to unknown translation and isotropic scale) is that the graph is fully triangulated [dBvKOS91]; triangles uniquely propagate global scale to all nodes in the graph. This condition is overly strict, however; equivalent rigidity can often be attained without full triangulation. Necessary conditions on the topology are highly dependent on the relative positions of the cameras, and beyond the scope of this work; see for example [TD99].

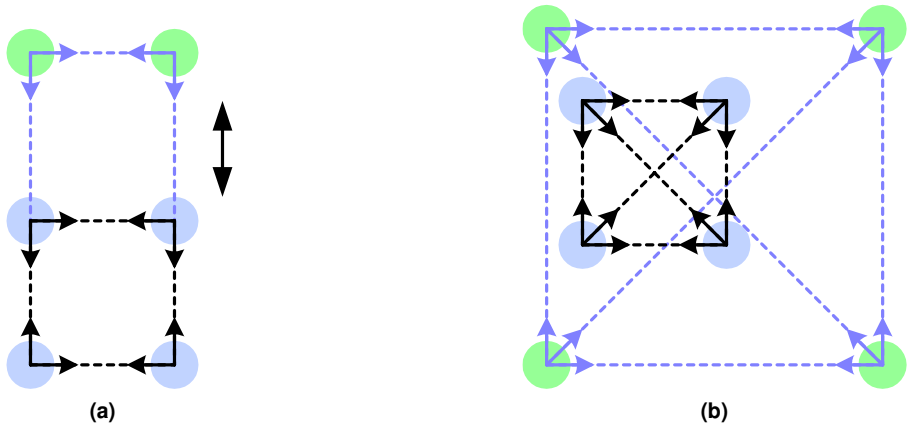


Figure 6-12: Topological Degeneracies
(a) Many camera topologies are degenerate; in this example, the configuration can be stretched. (b) Even when the adjacency graph is fully triangulated, there are inherent ambiguities of global scale and translation.

Degenerate configurations arise frequently in practice and preclude determination of unique pose sets. One solution that addresses both degeneracy and unknown Euclidean transformations is the addition of further constraints that utilize the initial, approximately known camera positions. Global scale can be imposed on the configuration by the constraint

$$\sum_{i,j} \alpha_{ij} = \sum_{i,j} \|\boldsymbol{p}_i^0 - \boldsymbol{p}_j^0\| \tag{6-27}$$

where $\boldsymbol{p}_i^0$ is the initial position of camera i, summing over all relevant camera pairs. This constraint states that the sum of all unknown inter-camera distances should equal the sum of all initial inter-camera distances, thus roughly preserving the global scale of the original configuration. Using the sum of distances is preferable to using a single distance because it allows unbiased treatment of the entire configuration and distributes error. Stronger conditions on the global scale could be enforced by constraining each α_{ij} individually, but this is unnecessary in practice.

Translational and topological ambiguities can be resolved using a set of weak constraints on the camera positions

$$\epsilon \boldsymbol{p}_i = \epsilon \boldsymbol{p}_i^0 \quad \forall i \tag{6-28}$$

for some small $\epsilon > 0$ (typically $\epsilon = 10^{-4}$), which have negligible effect on non-degenerate system modes but which constrain degenerate modes according to the initial configuration. Effectively, the constraints in (6-28) provide support in the nullspace of the equation system. There are now $3N + 3M + 1$ constraints described by (6-26), (6-27), and (6-28). These constraints are all linear in the $3N + M$ unknowns $\boldsymbol{p}_i$ and α_{ij}, and can thus be solved using ordinary linear least-squares techniques [Str88].

6.4.2 Constraints with Uncertainty

The previous formulation is completely deterministic and does not take into account the uncertainty in the pair-wise baselines estimated in §6.2. Qualitatively speaking, directions with high certainty

should be emphasized more in the global formulation than those with low certainty.

Although the true baseline directions are strongly correlated (due, for example, to closed loops in the camera topology), these directions are estimated in isolation, and uncertainty in the estimates is thus assumed to be uncorrelated. In the most general case, a 3×3 covariance matrix $\boldsymbol{C}_{ij}$ related to the Bingham uncertainty in baseline $\boldsymbol{b}_{ij}$ can be applied to each of the M constraints in (6-26) in order to provide adequate relative emphasis. The new constraints remain linear and take the form

$$\boldsymbol{C}_{ij}^{-\frac{1}{2}}(\boldsymbol{p}_j - \boldsymbol{p}_i - \alpha_{ij}\boldsymbol{b}_{ij}) = 0, \tag{6-29}$$

where the matrix $\boldsymbol{C}_{ij}$ can be obtained by computing the sample covariance from the Bingham parameters of $\boldsymbol{b}_{ij}$ (§B.2).

6.4.3 Uncertainty in Final Positions

Solution of the above system of equations results in a set of mutually consistent camera positions. The equation system can also produce Euclidean covariance matrices on these positions that describe the relative stability or certainty of the estimates, which is one of the goals of this work.

Let $\boldsymbol{A}\boldsymbol{u} = \boldsymbol{v}$ represent the set of linear equations formulated in (6-28), (6-27), and (6-29), where $\boldsymbol{u}$ is the $(3N + M) \times 1$ vector of unknowns, $\boldsymbol{A}$ is the $(3N + 3M + 1) \times (3N + M)$ matrix of coefficients on these unknowns, and $\boldsymbol{v}$ is the $(3N + 3M + 1) \times 1$ vector of constants. It can be shown (e.g. [PTVF92]) that the $(3N + M) \times (3N + M)$ covariance matrix $\boldsymbol{\Lambda}$ for the optimal least-squares parameter estimate $\tilde{\boldsymbol{u}}$ of this system is given by

$$\boldsymbol{\Lambda} = (\boldsymbol{A}^\top \boldsymbol{A})^{-1}. \tag{6-30}$$

This matrix describes the joint, correlated uncertainty among all camera positions $\boldsymbol{p}_i$ and all inter-camera distances α_{ij}. However, the uncertainty model for camera pose proposed in §3.4 calls for independent 3×3 covariance matrices on each of the positions. These can be obtained by finding the marginal density function (§3.1.2) of each position from the joint density.

In this formulation, the space of parameters has become Euclidean rather than projective, so for small perturbations a Gaussian model suffices to describe uncertainty. The matrix $\boldsymbol{\Lambda}$ can be treated as the $(3N + M) \times (3N + M)$ covariance of a Gaussian distribution, and the 3×3 submatrix $\boldsymbol{\Lambda}_i$ corresponding to the coefficients of camera position $\boldsymbol{p}_i$ represents the marginal density of this position (i.e. the integral of the joint density over all other parameters). This method for obtaining uncertainties of position estimates is attractive because of its simple incorporation of all constraints and relative weights.

6.5 Metric Registration

Pose estimates recovered using the methods of §6.4 are globally consistent relative to each other. However, they reside in a locally defined and somewhat arbitrary coordinate system that does not necessarily correspond to the metric space of the scene. A rigid transformation consisting of translation, rotation, and scale can express camera pose with respect to any coordinate system of interest while preserving the local relationships among cameras.

In the absence of ground-truth 3-D measurements, the approximate initial pose serves as a reference frame to which the camera configuration is registered. In the City Project, pose estimates produced by the acquisition platform are expressed in units of meters in Earth-centered, Earth-fixed (ECEF) coordinates, which has its origin at the center of the Earth, its $\hat{\boldsymbol{z}}$ axis through the North Pole, and its $\hat{\boldsymbol{x}}$ axis through the Prime Meridian (zero longitude) [Her95]. If these estimates are unbiased, then the Euclidean transformation that best fits recovered camera positions to initial camera positions produces the best expression of pose in the desired coordinate frame.

6.5.1 Absolute Orientation

This section outlines the approach to absolute orientation, or 3-D to 3-D registration, as proposed by Horn [Hor87, Hor91]. The goal is to find the translation, rotation, and scale that best align the N recovered camera positions, or source points $\boldsymbol{x}_i$, with the N initial positions, or target points $\boldsymbol{y}_i$. A sequence of transformations on the source points that results in optimal alignment to the target points is presented rather than explicit expressions for the aggregate transformation.

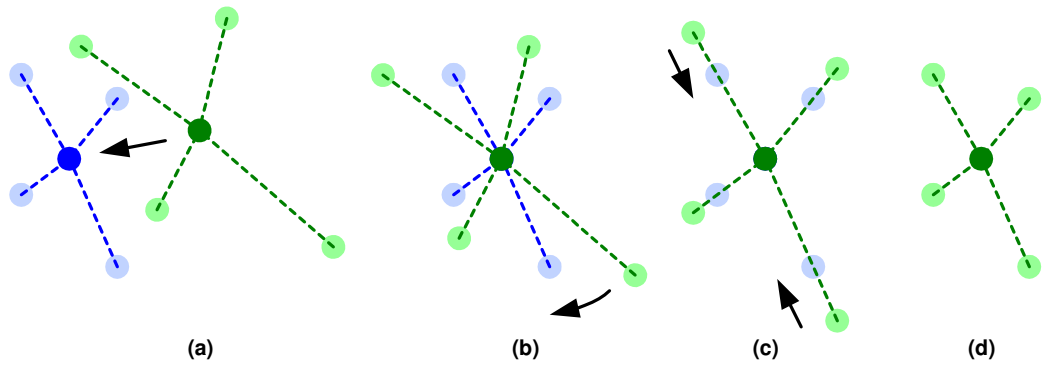


Figure 6-13: Metric Registration Process
A two-dimensional depiction of metric registration. (a) The original configuration is shifted so that the two centroids coincide. (b) Rays from the centroid to each camera rotationally aligned. (c) The optimal scale is computed and applied. (d) The final configuration.

First, each point set is translated so that its centroid is coincident with the origin. A new set of points is thus defined so that

$$\tilde{\boldsymbol{x}}_i = \boldsymbol{x}_i - \boldsymbol{x}_0, \qquad \tilde{\boldsymbol{y}}_i = \boldsymbol{y}_i - \boldsymbol{y}_0 \tag{6-31}$$

where

$$\boldsymbol{x}_0 = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i, \qquad \boldsymbol{y}_0 = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{y}_i. \tag{6-32}$$

This allows rotation and scale to be applied relative to the same origin, namely the centroid $\boldsymbol{x}_0$ and $\boldsymbol{y}_0$ of the two 3-D point sets.

The source points are then rotated by a matrix $\boldsymbol{R}$ to optimally align the rays through the points $\tilde{\boldsymbol{x}}_i$ and $\tilde{\boldsymbol{y}}_i$ originating at $\boldsymbol{x}_0$ and $\boldsymbol{y}_0$, respectively, as shown in Figure 6-13. The rotation $\boldsymbol{R}$ is estimated using the deterministic two-camera rotation method described in §5.4.1. Next, the optimal

scale factor s is computed as

$$s = \sqrt{\frac{\sum_{i=1}^{N} \tilde{\boldsymbol{y}}_i \cdot \tilde{\boldsymbol{y}}_i}{\sum_{i=1}^{N} \tilde{\boldsymbol{x}}_i \cdot \tilde{\boldsymbol{x}}_i}}. \tag{6-33}$$

Finally, the points are shifted from the origin back to the target points' centroid $\boldsymbol{y}_0$. The overall transformation acting on the source points is thus given by

$$\begin{aligned} \boldsymbol{g}(\boldsymbol{x}_i) &= s\boldsymbol{R}(\boldsymbol{x}_i - \boldsymbol{x}_0) + \boldsymbol{y}_0 \\ &= s\boldsymbol{R}\boldsymbol{x}_i + \boldsymbol{t} \end{aligned} \tag{6-34}$$

where $\boldsymbol{t} = \boldsymbol{y}_0 - s\boldsymbol{R}\boldsymbol{x}_0$. This is consistent with the derivation in [Hor87].

Probabilistic transformations are not necessary here because the target positions are not ground-truth quantities. However, the previously estimated pose uncertainty must undergo a similar set of transformations, described in the next section.

6.5.2 Transforming Uncertainty

Modification of the camera pose necessitates appropriate modification of its uncertainty. Let $\boldsymbol{x}$ be the position of a given camera before metric registration, with uncertainty described by a Gaussian random variable with mean at $\boldsymbol{x}$ and with covariance matrix $\boldsymbol{\Lambda}_{\boldsymbol{x}}$, and let $\boldsymbol{y}$ be the camera's position after registration. From (6-34),

$$\boldsymbol{y} = s\boldsymbol{R}\boldsymbol{x} + \boldsymbol{t}. \tag{6-35}$$

The new covariance is then given by

$$\begin{aligned} \boldsymbol{\Lambda}_{\boldsymbol{y}} &= \langle \boldsymbol{y}\boldsymbol{y}^\top \rangle - \langle \boldsymbol{y} \rangle \langle \boldsymbol{y}^\top \rangle \\ &= \langle (s\boldsymbol{R}\boldsymbol{x} + \boldsymbol{t})(s\boldsymbol{x}^\top \boldsymbol{R}^\top + \boldsymbol{t}^\top) \rangle - (s\boldsymbol{R}\langle \boldsymbol{x} \rangle + \boldsymbol{t})(s\langle \boldsymbol{x}^\top \rangle \boldsymbol{R}^\top + \boldsymbol{t}^\top) \\ &= s^2 \boldsymbol{R} \langle \boldsymbol{x}\boldsymbol{x}^\top \rangle \boldsymbol{R}^\top - s^2 \boldsymbol{R} \langle \boldsymbol{x} \rangle \langle \boldsymbol{x}^\top \rangle \boldsymbol{R}^\top \\ &= s^2 \boldsymbol{R} \boldsymbol{\Lambda}_{\boldsymbol{x}} \boldsymbol{R}^\top \end{aligned} \tag{6-36}$$

and is therefore independent of the translation $\boldsymbol{t}$.

Camera orientation is not affected by pure translation or scale; thus, rotational pose uncertainty is altered only by the rotation $\boldsymbol{R}$. A given camera's orientation is represented by a unit quaternion $\boldsymbol{q}$, which is a Bingham random variable $\mathcal{B}_4(\boldsymbol{q}; \boldsymbol{\kappa}, \boldsymbol{U})$. Intuitively, the concentration parameters $\boldsymbol{\kappa}$ should remain unchanged by the rotation; however, the orthogonal columns of $\boldsymbol{U}$, each of which is a distinct quaternion, should be transformed by $\boldsymbol{R}$. As outlined in §A.3, a quaternion acts on another quaternion as a matrix multiplication. The new orientation quaternion $\tilde{\boldsymbol{q}}$ is then given by $\tilde{\boldsymbol{q}} = \boldsymbol{Q}\boldsymbol{q}$, where $\boldsymbol{Q}$ is a 4×4 matrix representing $\boldsymbol{R}$. The same matrix can be used to transform the columns of $\boldsymbol{U}$, resulting in a new random variable distributed as $\mathcal{B}_4(\tilde{\boldsymbol{q}}; \boldsymbol{\kappa}, \boldsymbol{Q}\boldsymbol{U})$.

6.6 Summary of Position Recovery

This chapter presented a sequence of steps for the recovery of metrically aligned camera positions given known orientations and scene-relative 3-D line directions. First, camera adjacencies are determined from the approximately known initial pose. For each adjacent camera pair, a direction of motion is determined. To determine this motion, geometric constraints are imposed which drastically reduce the number of putative point feature matches, after which a Hough transform determines the most likely motion direction by considering all matches simultaneously. A MCEM algorithm then alternately refines the motion direction and computes probabilistic correspondence by sampling over all correspondence sets.

All two-camera motion directions are assembled into a set of linear constraints on camera positions, producing a globally-consistent pose configuration. Finally, this configuration is rigidly transformed for metric alignment with the original camera pose using a classical 3-D to 3-D alignment technique. The end result is a set of consistent camera positions and orientations, as well as estimates of their uncertainty.

CHAPTER 7

Experiments

AN IMPORTANT OBJECTIVE of this work, aside from development of sound theory and methods, is to demonstrate the experimental validity of the theoretical results. Performance of the techniques presented in this thesis must be assessed in various ways to establish their utility and verify their efficacy. Most often, real-world data is "uncalibrated"; that is, no ground-truth information about the data is known. It is therefore difficult to quantitatively assess the performance of any algorithm operating on such data, because truly correct results are unavailable for comparison.

There are three alternatives for addressing this problem. The first is to somehow generate correct results for a real data set using prior information; for example, to assess line classification techniques, real image lines can be manually partitioned into parallel sets based on a user's prior knowledge about the scene, and the results can be compared to those generated by automatic techniques. This method of assessment can be quite effective, but only if such prior knowledge is obtainable, and if the manual tasks are not overly daunting.

The second assessment method is to generate synthetic data which closely resembles real data, but whose parameters can be adjusted to study the effects of various input corruption on the final results. Synthetic data is generated by creating an ideal model (e.g. geometry and pose) whose characteristics are completely specified, then simulating observations of this model (e.g. feature projections) corrupted by random noise with controllable parameters (e.g. projection error and outliers). If these parameters are chosen so that the simulated data resembles real data, then comparison of the known model with that produced by an algorithm of interest is a valid measure of the algorithm's performance.

The third and final method of assessment examines the self-consistency of system outputs [LLF98]. Consistency measures are application specific; in this context, examples include 3-D ray discrepancies for corresponding point features, deviations of final camera baselines from their initial estimates, and bounds on estimated pose uncertainty.

This chapter presents an array of experiments designed to highlight the strengths and weaknesses of the automatic registration methods developed in this thesis. §7.1 quantitatively illustrates

performance degradation of various system stages with varying degrees of random input perturbations. §7.2 presents end-to-end results on several real-world data sets using qualitative performance metrics.

7.1 Simulation

Automatic registration comprises many smaller stages and components. While there is merit to evaluating each component individually, interaction between components is not captured by such evaluation; the whole is generally greater than the sum of its parts. It is therefore important, from a systems standpoint, to also examine how these components interact and to make end-to-end observations.

With this in mind, several experiments were designed and run on simulated data to characterize the performance of individual and combined system stages. This section briefly describes each experiment and presents the results; further comments are given in the next chapter.

7.1.1 Projective Data Fusion Performance

The projective inference and fusion techniques presented in Chapter 3 were qualitatively and quantitatively assessed. Two data sets were generated, each designed to simulate a different type of spherical distribution. The first set imposed a coplanar constraint to form equatorial distributions, while the second imposed a collinearity constraint to form bipolar distributions. 50 different runs of 200 points were generated for each of the distribution types.

Points in a given run were generated as follows. First, a projective point $\boldsymbol{y}$ to be used as a reference (for example, the modal direction of a bipolar distribution) was randomly chosen. Next, a data point $\tilde{\boldsymbol{x}}_i$ was randomly chosen to precisely satisfy the appropriate projective constraint. A bimodal Bingham distribution with controllable concentration parameters and major axis aligned with $\tilde{\boldsymbol{x}}_i$ was then generated, and a single random sample was drawn from this distribution and used as the noisy observation $\boldsymbol{x}_i$. The parameter matrix $\boldsymbol{M}_i$ of the Bingham distribution was rotated to align with $\boldsymbol{x}_i$ and used as the point's measurement uncertainty.

The overall distribution of the points in a given run was then computed by projective fusion. The major axis of this distribution was compared to the reference point $\boldsymbol{y}$ described above; results are shown in Figure 7-1. Tests were also run on varying numbers of points generated by uniform and bipolar distributions (Figure 7-2).

The data fusion experiments above confirm the behavior predicted by laws of large numbers (§3.1.4). First, there is a direct dependence of estimation error on error in the underlying data; and second, there is an inverse dependence of covariance on the number of data points used. Quantities are thus estimated with higher certainty as the sample size increases.

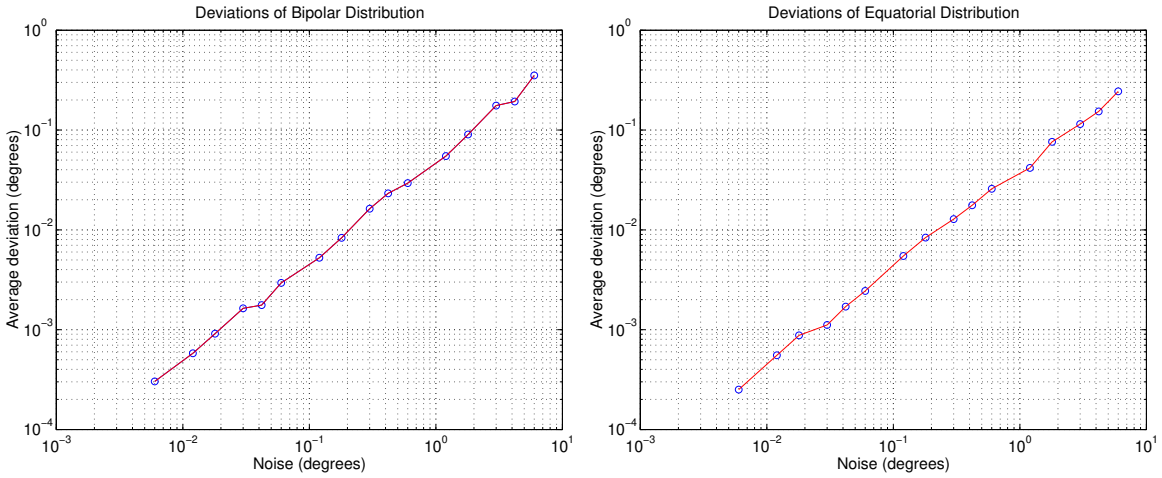


Figure 7-1: Deviation of Estimated Axes

Noisy samples were generated according to a Bingham distribution satisfying projective constraints, and fused to form a summarizing distribution. The major axes of this distribution were compared with the axes of the original distribution and the average angular error reported. Estimates of bipolar and equatorial distributions are shown at left and at right, respectively.

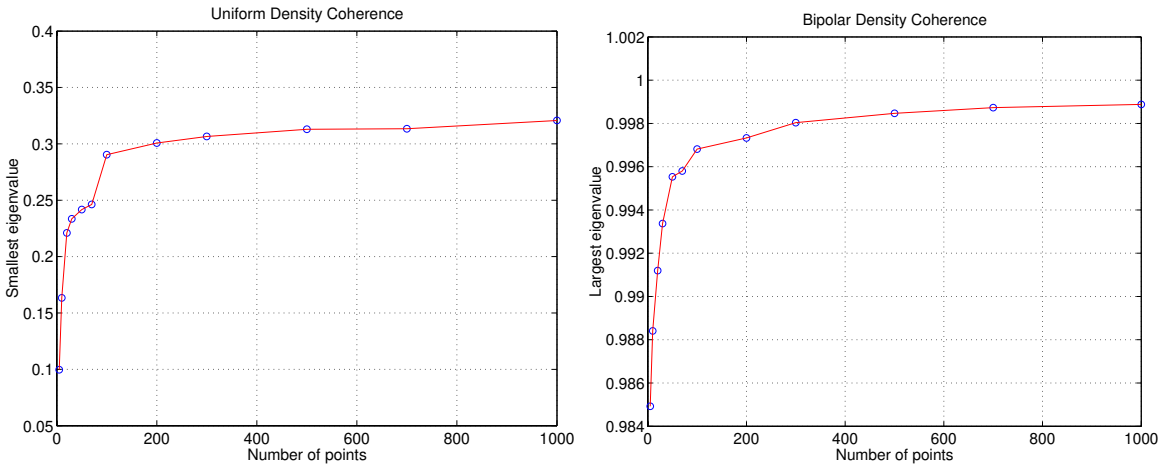


Figure 7-2: Distribution Coherence for Data Fusion

Several data sets of varying sample sizes were generated by Bingham densities with known parameters. For each set, a new Bingham density was formed by projective fusion of the samples, and the eigenvalues of its resulting covariance matrix was examined. The eigenvalues of a uniform distribution, shown at left, should all be equal to $\frac{1}{3}$; the minimum eigenvalue quickly approaches this value as the number of points increases. The maximum eigenvalue of a perfect bipolar distribution, shown at right, should be equal to 1; this value is asymptotically approached as the sample size increases.

7.1.2 Single-Camera Vanishing Point Estimation

Several experiments were designed to evaluate the vanishing point estimation techniques of §5.3. Parallel sets of ideal 3-D lines were generated and projected onto the unit sphere; the same process as in §7.1.1 was then applied to these ideal projections to produce randomly perturbed data with controllable noise levels. A controllable percentage of random outlier lines was also produced, generated by a uniform distribution on the sphere, and the number of distinct 3-D directions (denoted by J in the notation of Chapter 5) was varied.

The Hough transform method for EM initialization was performed on 50 data sets, each containing a mixture of 500 points and outliers. The percentage of true peaks detected, as well as the angular deviation of the peaks from the true 3-D line directions, were both examined as a function of measurement noise, outlier percentage, and J (Figure 7-3). Cells were sized so that the maximum angular coverage was $1°$, and a window size of 5 cells (roughly $5°$) was used for peak detection. The proximity function for each point was taken to be a Gaussian with standard deviation σ of roughly $2°$, and the peak acceptance coefficient α was equal to 2.

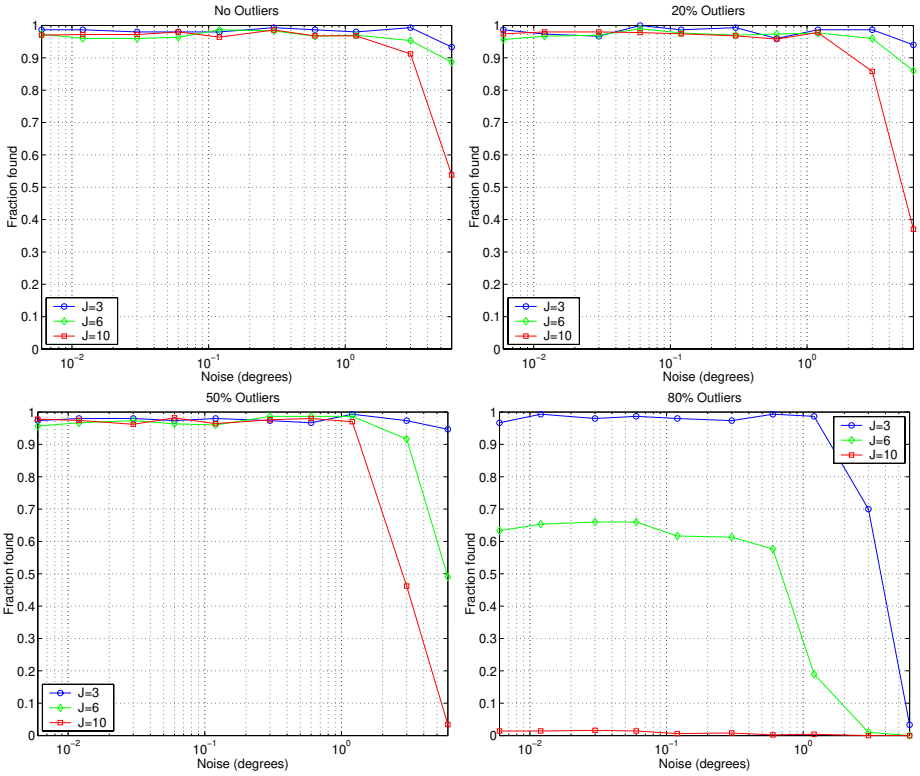


Figure 7-3: Percentage of Vanishing Points Detected
Each plot shows the average percentage of true vanishing points detected as a function of point projection error, number of true line directions, and number of outlier features (expressed as a percentage of the total number).

Successfully detected vanishing points were consistently within about $1°$ of the true directions. A small number of false peaks were identified (about 2%), but only when feature noise exceeded several degrees. Performance of the expectation maximization algorithm (as initialized by the

Hough transform) was also assessed, varying the same parameters as above. Angular devations of the estimates from the true directions are shown in Figure 7-4.

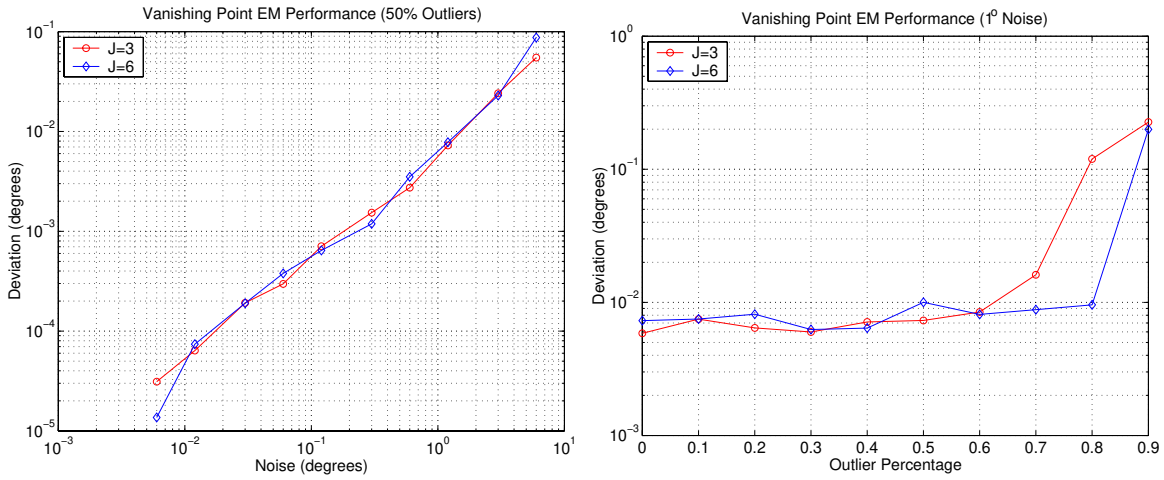


Figure 7-4: EM Vanishing Point Error
Average error in vanishing points estimated by the EM algorithm are plotted as a function of line feature noise with 50% outliers (left) and outlier percentage with 1° feature noise (right).

Vanishing point estimation has proven to be quite robust. The Hough transform and peak detection methods provide sound initial estimates, with performance degradation occurring only at very high levels of feature noise; for moderate feature noise, all correct vanishing points are detected even with an outlier-to-data ratio of 4:1. Mixture model estimates obtained from the EM algorithm are fairly consistent with the data fusion experiments in §7.1.1. Initialization and prior distributions provided by the Hough transform make the EM algorithm robust against outliers. Performance degrades as the number of contributing vanishing points increases, because features tend to crowd the closed projective space and vanishing point clusters "interfere" with one another. However, since there are typically only two to six prominent line directions in typical real-world data (e.g. urban scenes), interference effects are negligible in practice.

7.1.3 Two-Camera Rotational Pose

To assess the two-camera rotation method, a set of 4 randomly generated 3-D line directions was viewed by two cameras and perturbed by controllable noise. Outlier directions were also added to each camera, and the stochastic two-camera registration method described in §5.4 was applied to 50 such data sets. Angular error was obtained using the method described in §A.4.

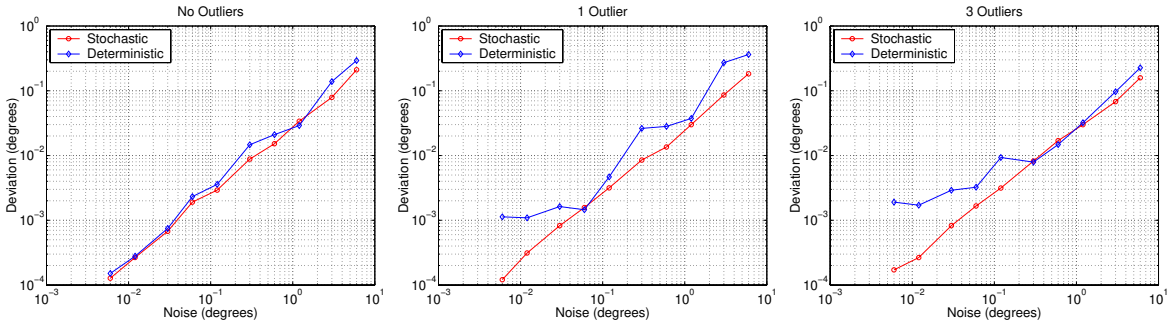


Figure 7-5: Comparison of Two-Camera Rotation Methods
The stochastic two-camera rotational registration technique is compared with the classical deterministic technique with 4 vanishing points. The plots show relative pose error as a function of vanishing point noise with 0, 1, and 2 outlier directions introduced. Behavior of the stochastic method exhibits more stability and consistency.

Incorporation of stochastic correspondence and vanishing point uncertainty improves rotational registration. Side-by-side comparison of deterministic and stochastic registration methods in Figure 7-5 shows the novel method to be more stable and consistent than the deterministic rotation technique of §5.4.1, and also to produce more accurate estimates.

7.1.4 Multi-Camera Orientation

End-to-end rotational pose recovery was examined by generating parallel 3-D lines and outliers as above, and projecting this geometry onto randomly situated cameras with controllable pose perturbations. As long as correspondence was unambiguous (§5.4.4), correct orientations were correctly recovered for arbitrary initial rotational error, even up to $180°$. Figure 7-6 shows error in orientation estimates.

As expected, accuracy of multi-camera rotational alignment increases (though slightly) with the number of vanishing points, since each vanishing point match represents a single sample of a distribution and estimates generally improve with more observations. It is unclear why estimation error does not decrease more quickly as the number of cameras increases; one would expect that more observations of a single entity (namely a given 3-D line direction) should increase the certainty with which that entity is estimated, and consequently the accuracy of rotational estimates.

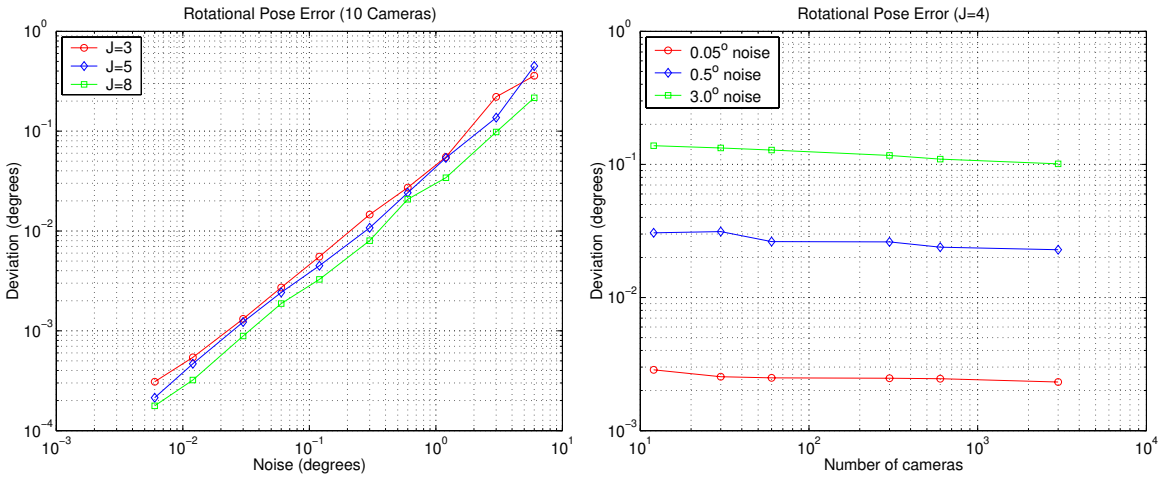


Figure 7-6: Multi-Camera Orientation Performance

Performance of multi-camera rotational registration was examined. At left, the average angular pose error is reported as a function of vanishing point noise for 10 cameras viewing varying numbers of 3-D line directions. At right, rotational error is plotted as a function of the number of cameras in the configuration with varying degrees of noise in 4 vanishing points.

7.1.5 Two-Camera Translational Pose

For translational pose recovery, 3-D points were randomly generated and projected onto cameras at different spatial positions. Controllable projection noise (in the form of bipolar Bingham distributions) and outlier observations were generated by a process similar to that for line features (§7.1.2).

The two-camera method was evaluated in several ways. Performance of the Hough transform and MCEM algorithm described in §6.2 and §6.3 was assessed by comparing the estimated motion direction with the true camera baseline using different levels of feature noise and outlier percentages. The true motion direction was perturbed by a random amount and used to initialize this technique; the uncertainty bound discussed in §6.1.2 was chosen as 3σ, where σ^2 is the variance of the perturbation.

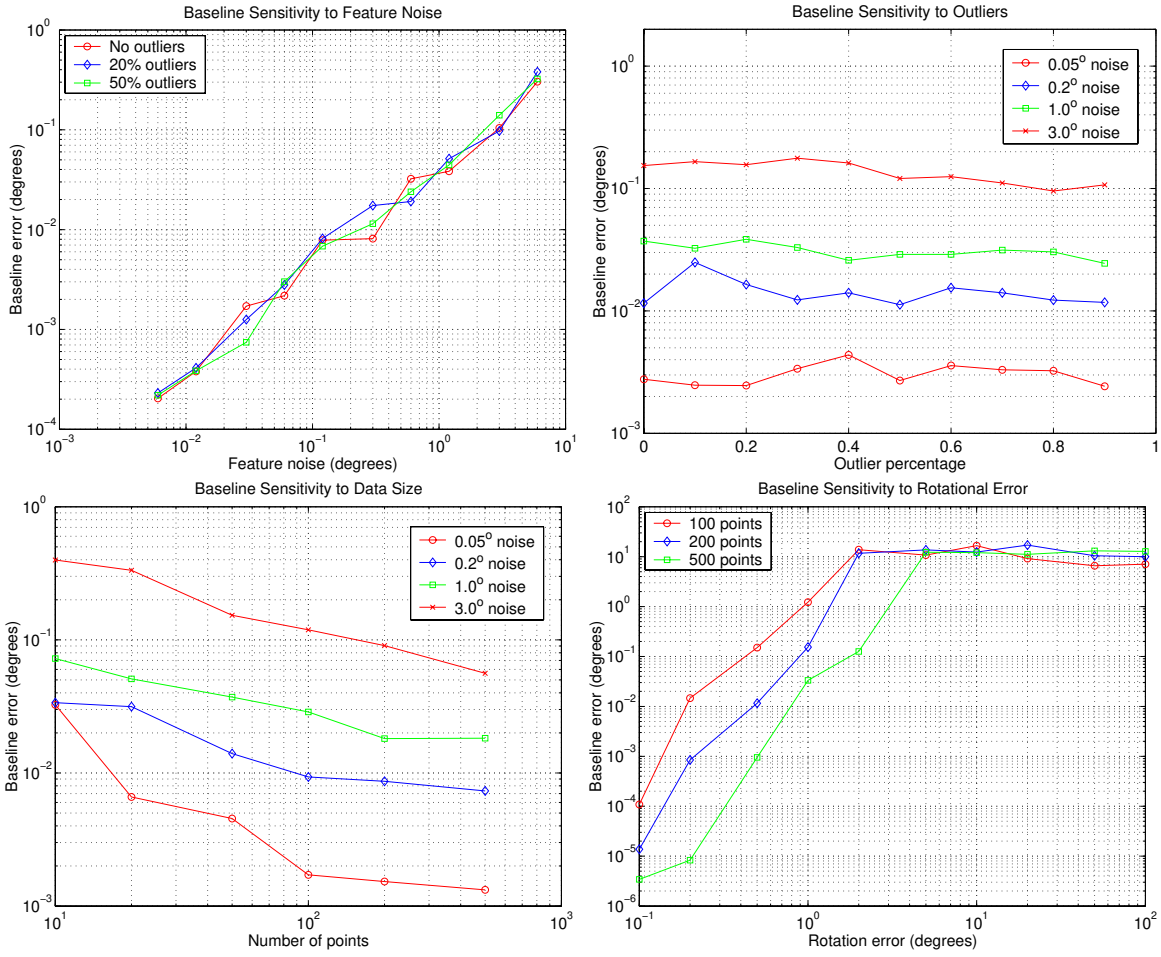


Figure 7-7: Baseline Estimate Accuracy
Sensitivity of baseline estimates was examined as a function of projective point feature noise, outlier percentage, number of points observed, and rotational pose error.

Like vanishing points, baseline estimates are robust against even extremely high outlier percentages due to the Hough transform initialization. The MCEM technique behaves as expected,

with a direct dependence on feature noise and increasing accuracy as the number of true correspondences increases. The method is sensitive to rotational error, however, because the entire epipolar formulation fundamentally depends on accurate camera orientations. Baseline estimation error quickly increases with rotational misregistration; the plateau in Figure 7-7 occurs because of the explicit bound imposed on the baseline direction (§6.1.2).

The correspondence probability matrix from the MCEM algorithm of §6.3.4 is shown at different stages of state evolution in Figure 7-8. Although match matrices produced by MCEM do not perfectly capture true feature correspondence when significant noise is present, this correspondence is never explicitly needed; in this context, the true performance measure is baseline accuracy, which directly affects the accuracy of global registration (Figure 7-10).

Direction estimates obtained by the MCEM algorithm are compared to those obtained by a simple deterministic ICP method in Figure 7-9. The ICP algorithm is identical to the MCEM algorithm, except that instead of estimating probabilistic match weights at each E-step, ICP determines the set of "best" explicit matches given the current baseline direction. The comparison in Figure 7-9 suggests that probabilistic correspondence provides more stablity and accuracy than does explicit correspondence.

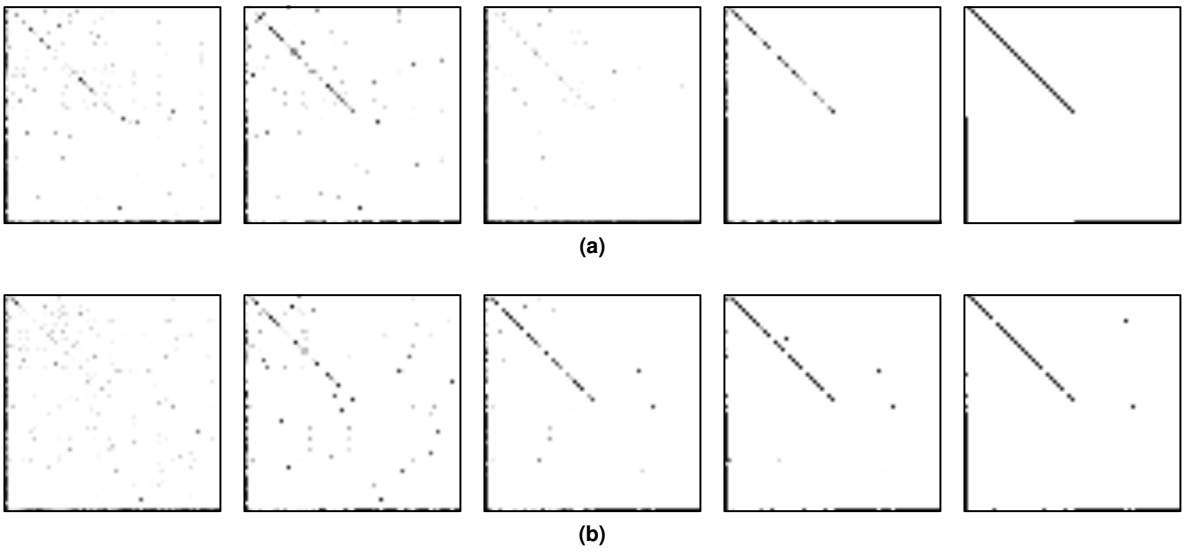


Figure 7-8: MCEM State Matrix Evolution

Evolution of the state matrix encoding correspondence is shown at various stages of the MCEM algorithm. (a) Successive iterations for point feature noise of $0.05°$, in which correspondence is perfectly recovered. (b) Iterations for point feature noise of $0.5°$; a few features are misclassified.

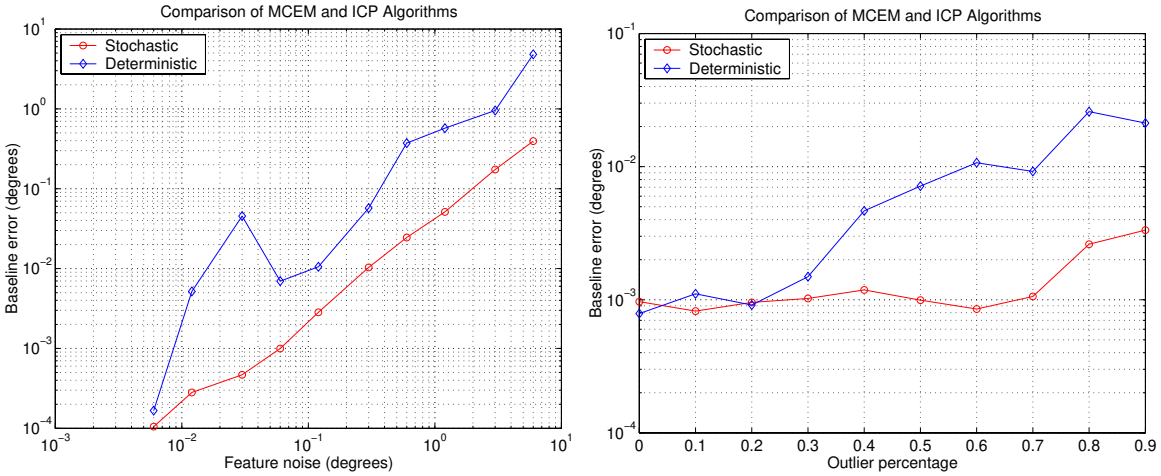


Figure 7-9: MCEM Baseline Comparison

Performance of the MCEM algorithm is compared with a simple ICP algorithm for the same data sets. Angular baseline error is reported as a function of feature noise and outlier percentage for both methods.

7.1.6 Global Registration

The final system stage is global registration, which involves determination of a consistent pose configuration given a set of inter-camera baseline directions. A set of camera positions was randomly generated, and all pair-wise motion directions assembled. Each direction was then perturbed by a Bingham noise process with controllable parameters; the perturbed set of directions, as well as adjacency information, then drove the global registration algorithm. Results (in terms of deviation from the initial "true" pose) are shown in Figure 7-10

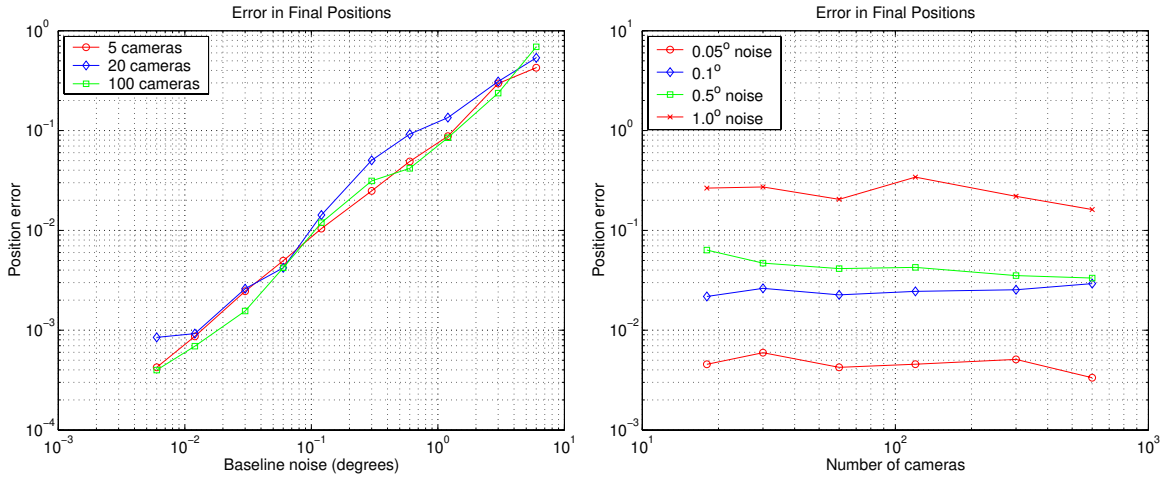


Figure 7-10: Global Registration Results
Accuracy of global position recovery is plotted as a function of baseline error and number of cameras in the configuration. Performance stays roughly constant with the size of the configuration.

Surprisingly, there is no evidence that the accuracy of final position estimates depends on the number of cameras in the configuration. Although this seems at first to contradict intuition and the laws of large numbers, it is a reasonable result in light of the fact that each camera has a roughly constant number of neighbors (§4.4). Global pose relies solely on baselines, and regardless of the total number of cameras, the number of baseline estimates *per camera* remains constant. Because the algorithm treats all cameras equally and considers all information simultaneously, error is distributed equally throughout the configuration; thus, for a given number of neighbors, the error should remain bounded (roughly constant) even as the camera topology grows.

7.2 Real Data

The pose recovery system was applied to several data sets from the City Project. Although it is difficult to quantify accuracy when no metric measurements are available, several qualitative descriptions of error and consistency were devised and accompany each data set:

- ***Data size***. Tables list the number of hemispherical nodes ("Nodes") and rectangular images ("Images") in the set, as well as the average and total number of line features ("Lines"), point features ("Points"), and vanishing points ("VPs") detected. The number of adjacent camera pairs ("Pairs") and the average baseline distances are also reported.

- ***Computation time***. The algorithms were implemented and run on a 250 MHz SGI Octane with 1.5 Gbytes of RAM. Average and total running times for each stage of the pose recovery system are reported (not counting file I/O).

- ***Pose offsets***. Accuracy of the initial pose was assessed by examining the degree to which the pose recovery algorithms rotated and translated each camera. Average and maximum deviations from the initial orientations and positions are reported. These quantities also provide a notion of the degree of robustness exhibited by the automated techniques.

- ***Error bounds***. The system produces estimates of uncertainty in every geometric entity. Probability density parameters of vanishing points, orientations, and positions are transformed into approximate 95% confidence bounds ("VP Bound", "Rot Bound", and "Trans Bound", respectively), and the average and maximum sizes of these bounds are reported.

- ***Feature consistency***. The probabilistic match matrices estimated during baseline computations were converted to binary match matrices by extracting the maximum entry of each row and column. Each entry exceeding a threshold (corresponding roughly to 80% probability) was taken as an unambiguous match, and its constituent point features were examined using two error measures. The first is the average 3-D discrepancy between extruded rays, denoted as "3-D Ray Error"; the second is average 2-D discrepancy (in image pixels) between an epipolar line and its corresponding point, denoted as "2-D Epi Error".

7.2.1 TechSquare Data Set (81 Nodes)

Camera pose registered using manual correspondence was available for the TechSquare data set [CMT98], which consists of 81 nodes spanning an area of roughly 285 by 375 meters. This data thus lends itself to qualitative comparison of the automatic techniques developed in this thesis with bundle adjustment from hand correspondence. Of the 81 nodes in the set, 75 (or roughly 92%) were "successfully" registered; 6 of the nodes were discarded due to insufficient vanishing point information.

Consistency of epipolar geometry indicates that initial rotational errors of over 17° and position errors of over 6 meters were automatically corrected by the system. The system recovered global pose consistent on average to 0.072° of orientation, 5.6 cm of position, and 1.22 pixels. The maximum error reported was 0.098° of orientation, 11.0 cm of position, and 5.71 pixels.

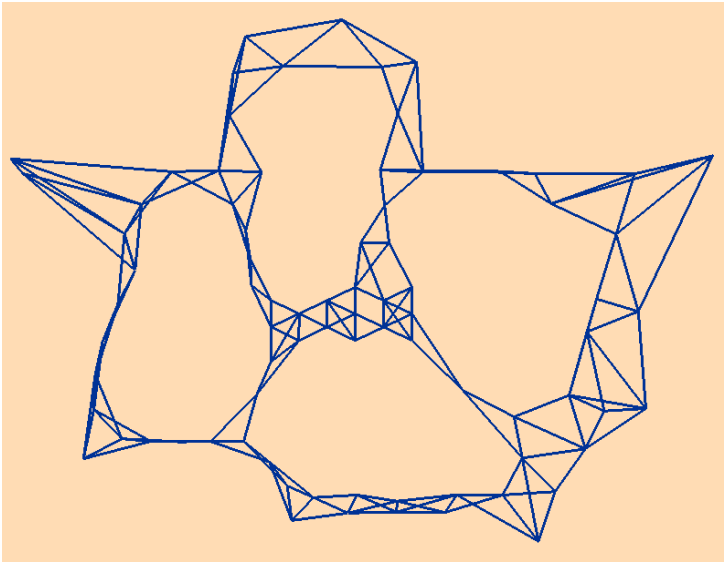


Figure 7-11: TechSquare Node Configuration
A top-down map view of the node configuration and adjacencies for the TechSquare data set. The average baseline was 30.88 meters.

	Per Image	Per Node	Total
Nodes	—	—	81
Pairs	—	—	189
Images	—	48	3899
Lines	218	10,516	851,819
Points	227	10,958	887,598
VPs	0.69	3.6	9

Table 7-1: TechSquare Data Size

	Per Node	Total
VP Hough	0.19 s	0 m 15 s
VP EM	6.68 s	7 m 54 s
Rotation EM	—	0 m 46 s
Total	6.87 s	8 m 55 s

	Per Pair	Total
Baseline Hough	8.1 s	25 m 31 s
Baseline MCEM	45.3 s	2 h 23 m
Global Opt	—	0 m 53 s
Total	53.4 s	2 h 49 m

Table 7-2: TechSquare Computation Times

	Average	Maximum
VP Angle Error	0.056°	0.09°
VP Bound	0.18°	0.80°
Rot Offset	1.53°	17.18°
Rot Bound	0.072°	0.098°
Trans Offset	0.70 m	6.70 m
Trans Bound	5.6 cm	11.0 cm

	Average	Maximum	Std. Dev.
3-D Ray Error	9.6 cm	12.4 cm	3.3 cm
2-D Epi Error	1.22 pixel	5.71 pixel	2.33 pixel

Table 7-3: TechSquare Error Assessment

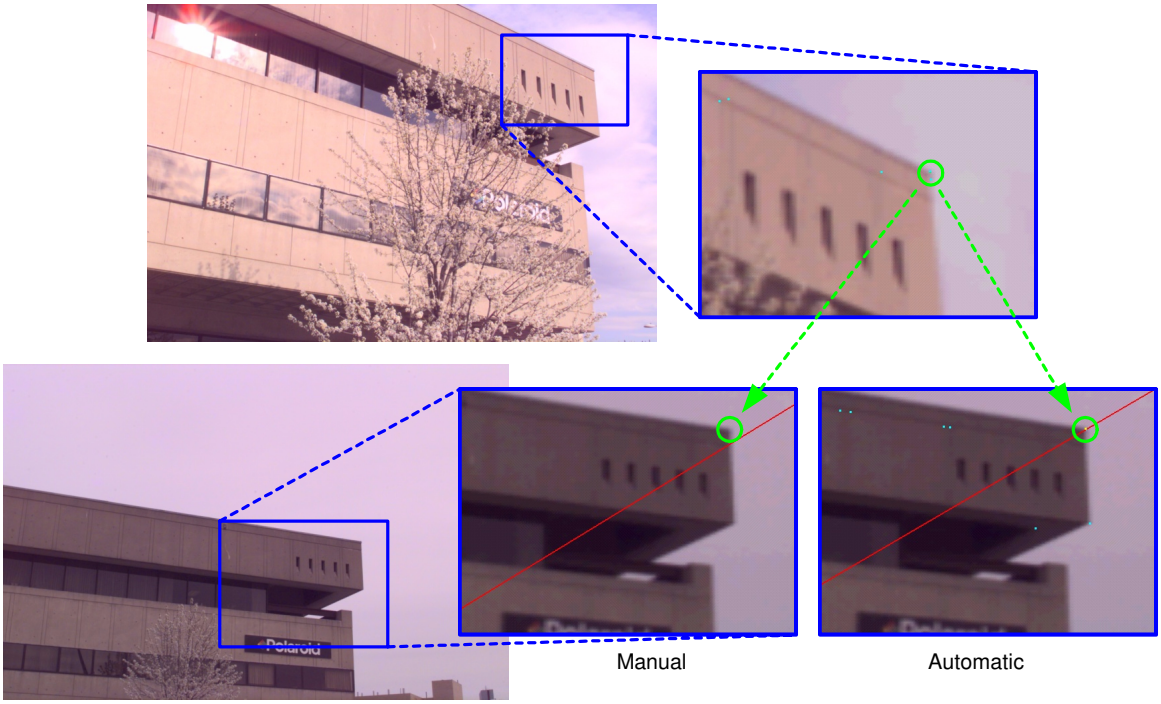


Figure 7-12: TechSquare Epipolar Geometry Comparison I
A particular feature in one image and its corresponding epipolar line in another image, as computed using cameras generated by manual correspondence vs. automatic refinement methods. Note the error resulting from manual correspondence.

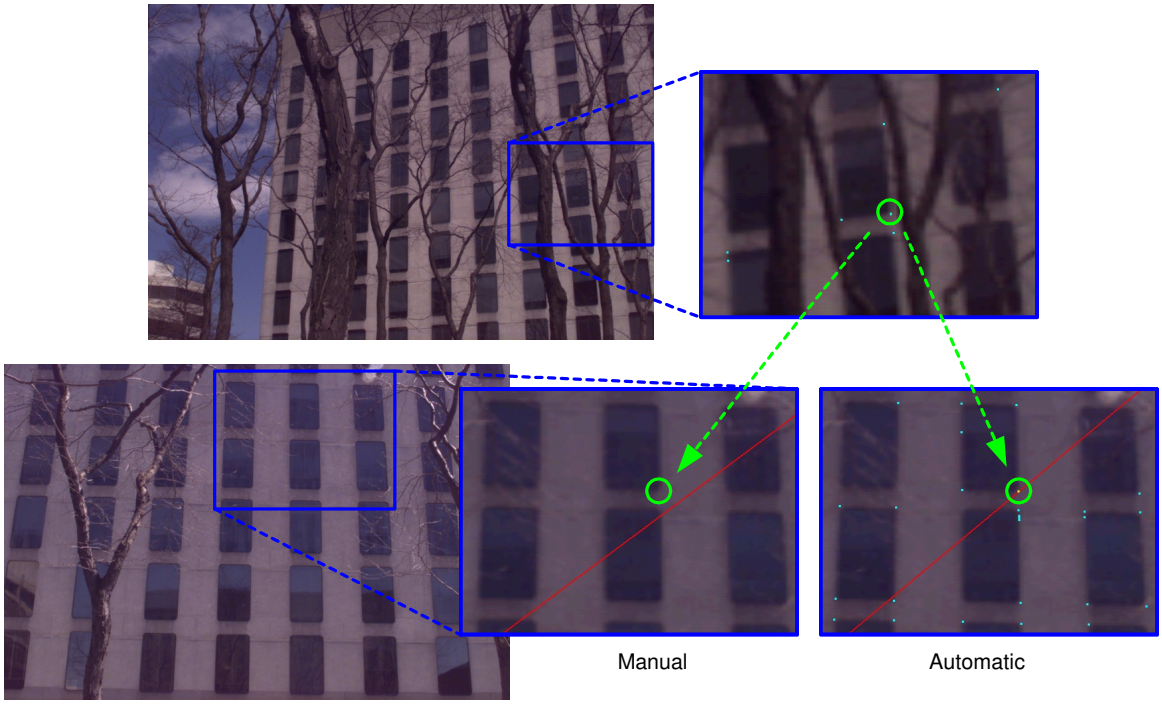


Figure 7-13: TechSquare Epipolar Geometry Comparison II
A feature whose match may be difficult for a human operator to identify. Epipolar lines are drawn as computed using cameras generated by manual correspondence vs. automatic refinement. Note the error in epipolar geometry resulting from the manual technique.

7.2.2 GreenBuilding Data Set (30 nodes)

A relatively small number of nodes from a set with noisy initial pose was chosen to demonstrate the robustness of the automatic techniques with respect to initial pose error. The 30 nodes spanned an area of roughly 80 by 115 meters, and all were successfully registered.

With rotational error of 6.83° and position error of 5.97 meters even over short baselines, the system was able to recover pose consistent on average to 0.067° of orientation, 4.5 cm of position, and 2.21 pixels, and within 0.12° of orientation, 8.1 cm of position, and 4.17 pixels.

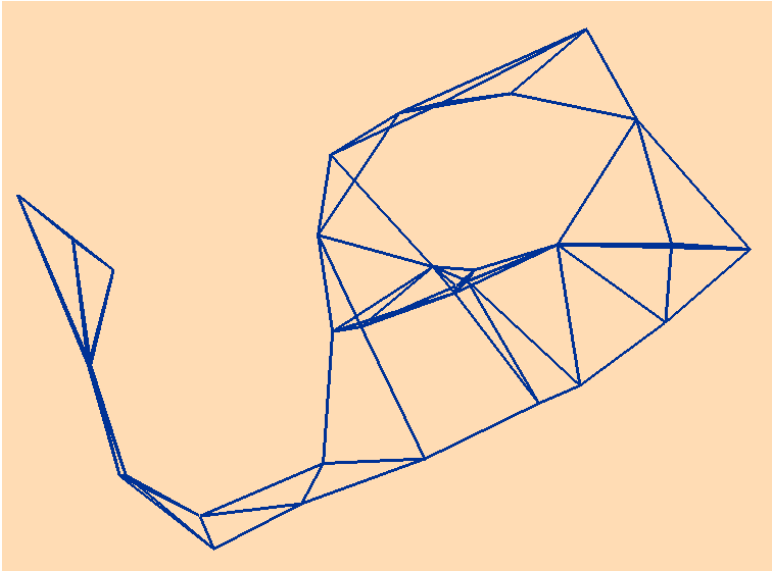


Figure 7-14: GreenBuilding Node Configuration
A top-down map view of the node configuration and adjacencies for the GreenBuilding data set. The average baseline was 15.61 meters.

	Per Image	Per Node	Total
Nodes	—	—	30
Pairs	—	—	80
Images	—	23	695
Lines	237	5,498	164,945
Points	257	5,967	179,030
VPs	0.35	3.3	5

Table 7-4: GreenBuilding Data Size

	Per Node	Total
VP Hough	0.11 s	0 03 m
VP EM	2.93 s	1 28 m
Rotation EM	—	0 18 m
Total	3.04 s	1 49 m

	Per Pair	Total
Baseline Hough	6.2 s	8 16 m
Baseline MCEM	42.5 s	56 20 m
Global Opt	—	0 21 m
Total	48.7 s	1 05 h

Table 7-5: GreenBuilding Computation Times

	Average	Maximum
VP Angle Error	0.047°	0.11°
VP Bound	0.092°	0.52°
Rot Offset	2.95°	6.83°
Rot Bound	0.067°	0.12°
Trans Offset	2.86 m	5.97 m
Trans Bound	4.5 cm	8.1 cm

	Average	Maximum	Std. Dev.
3-D Ray Error	10.2 cm	18.5 cm	5.3 cm
2-D Epi Error	2.21 pixel	4.17 pixel	1.43 pixel

Table 7-6: GreenBuilding Error Assessment

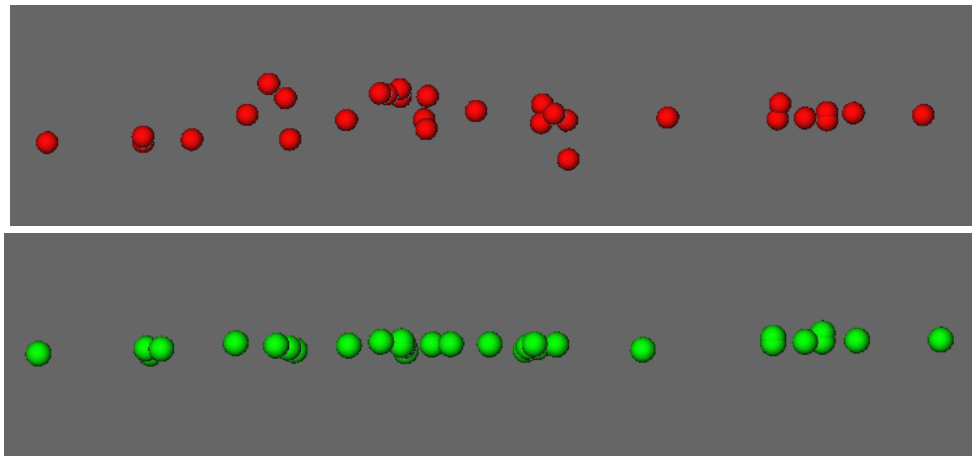


Figure 7-15: GreenBuilding Data Corrections
(a) A side view (looking parallel to the ground) of the node topology before pose refinement. All nodes were acquired at roughly the same height above the ground; note, however, the large variation in initially estimated camera heights. (b) After refinement, much of the height variation has been corrected.

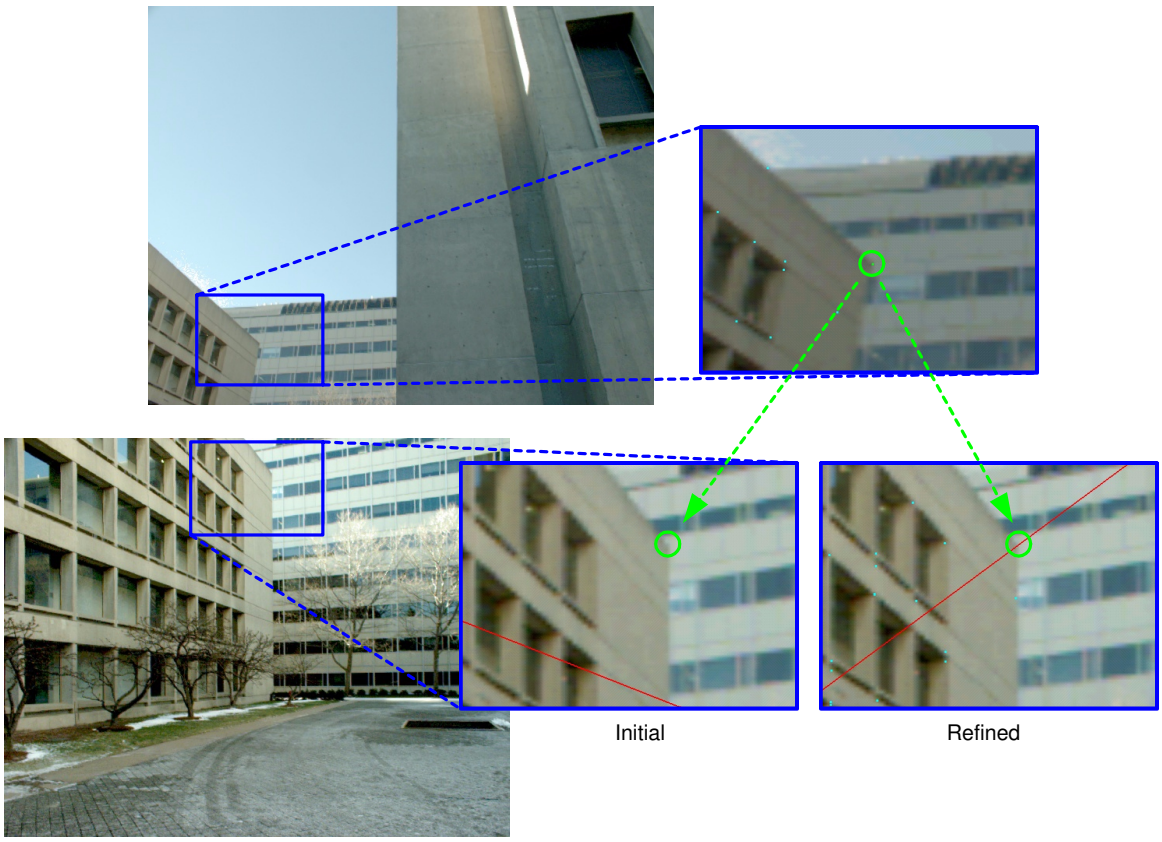


Figure 7-16: GreenBuilding Epipolar Geometry Comparison
Epipolar geometry inferred from initial cameras is compared to that inferred from the refined pose. Significant pose error is corrected by the automatic registration technique.

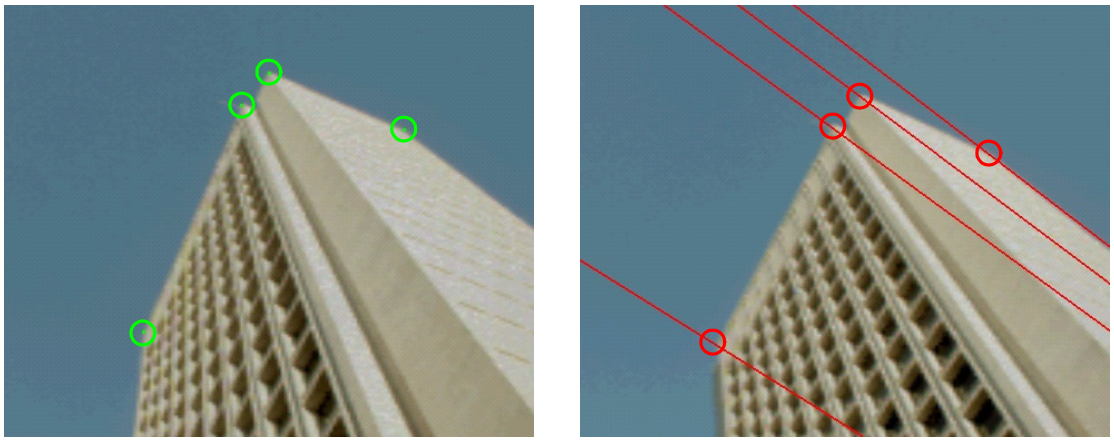


Figure 7-17: GreenBuilding Epipolar Geometry
Epipolar lines after registration are consistent to within a few pixels, even for distant 3-D points. Error in initial pose is substantial; the same lines inferred from the this pose fail to intersect the image.

7.2.3 AmesCourt Data Set (100 nodes)

The AmesCourt set spans an area of 315 by 380 meters, representing a larger portion of geography and a larger number of camera sites. Of the 100 nodes in this set, 95 were registered successfully. Numerical results and illustrations are shown below.

Initial pose was corrected by 5.59° and 6.18 m, achieving average consistency of 0.095°, 5.7 cm, and 3.88 pixels. Errors did not exceed 0.21°, 8.8 cm, or 5.02 pixels.

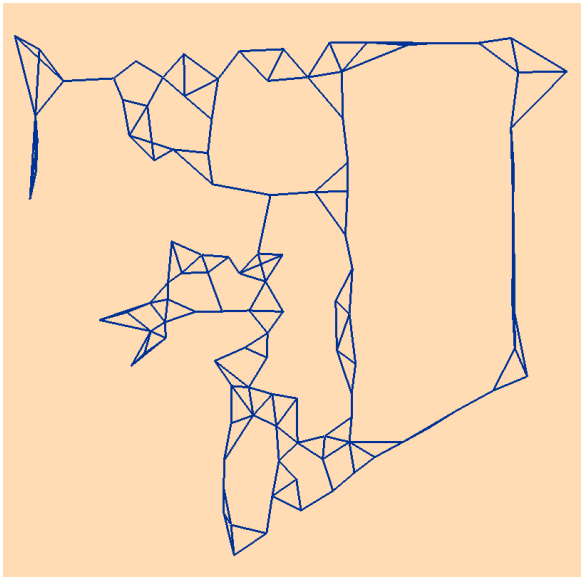


Figure 7-18: AmesCourt Node Configuration
A top-down map view of the node configuration and adjacencies for the AmesCourt data set. The average baseline was 23.53 meters.

	Per Image	Per Node	Total
Nodes	—	—	100
Pairs	—	—	232
Images	—	20	2,000
Lines	228	4,562	456,246
Points	257	4, 132	413,254
VPs	0.43	3.2	8

Table 7-7: AmesCourt Data Size

	Per Node	Total
VP Hough	0.09 s	10 s
VP EM	2.55 s	4 m 22 s
Rotation EM	—	33 s
Total	2.64 s	5 m 05 s

	Per Pair	Total
Baseline Hough	7.8 s	30 m 10 s
Baseline MCEM	52.6 s	3 h 24 m
Global Opt	—	1 m 04 s
Total	60.4 s	3 h 55 m

Table 7-8: AmesCourt Computation Times

	Average	Maximum
VP Angle Error	0.043°	0.09°
VP Bound	0.23°	0.74°
Rot Offset	2.83°	5.59°
Rot Bound	0.095°	0.21°
Trans Offset	3.53 m	6.18 m
Trans Bound	5.7 cm	8.8 cm

	Average	Maximum	Std. Dev.
3-D Ray Error	14.9 cm	20.2 cm	5.6 cm
2-D Epi Error	3.88 pixel	5.02 pixel	2.10 pixel

Table 7-9: AmesCourt Error Assessment

Figure 7-19: AmesCourt Epipolar Geometry
Several point features and corresponding epipolar lines for a typical node pair in the AmesCourt set. There is good agreement between features, suggesting accurately recovered pose.

7.2.4 Omnidirectional Images

There are several advantages to using wide FOV spherical images rather than narrow FOV planar images, one of which is coherence of direction estimates. To illustrate this effect, a particular vanishing point in a single City Project node was estimated using a Hough transform, varying the number of constituent images from the hemispherical tiling described in §2.2.1 and thus varying the field of view. The baseline direction estimate between two adjacent nodes was also examined as a function of the number of constituent images. Transform values are plotted in Figure 7-20.

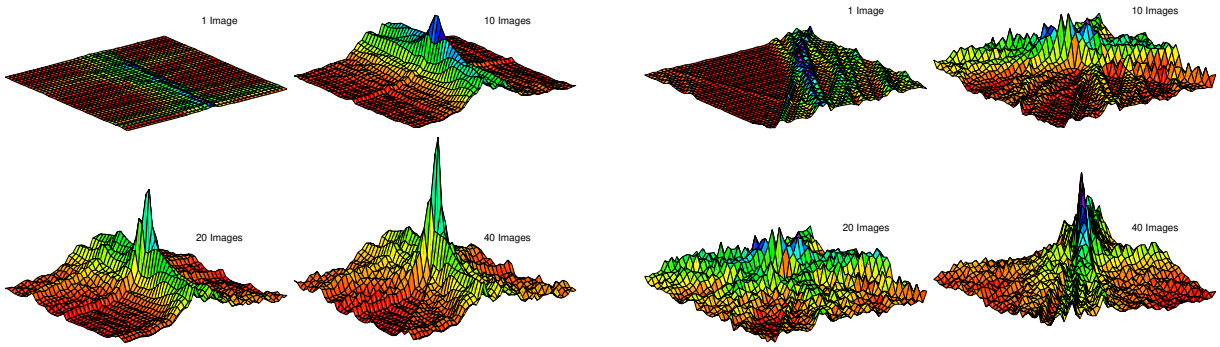


Figure 7-20: Hough Transform Peak Coherence
Illustrations of the dependence of Hough transform peak coherence on field of view for nodes containing 47 images. Peaks are shown for a vanishing point (left) and a baseline direction (right) for varying numbers of images in the hemispherical tiling.

The statistics in Table 7.2.1, Table 7.2.2, and Table 7.2.3 show further evidence of the superior utility of spherical imagery. Vanishing points were estimated from the line features of each individual rectangular image of each node; the number of vanishing points detected in these individual images was on average much lower than the number detected in the aggregate nodes. By definition, a larger field of view "sees" more of the surrounding geometry, and is thus more likely to capture relevant scene structure.

CHAPTER 8

Conclusions

PRECEDING CHAPTERS developed theory and algorithms for robust, automatic recovery of external camera pose, and demonstrated their performance on both simulated and real data sets. This chapter summarizes the material presented thus far, and offers further analysis and insight. §8.1 discusses the experimental results presented in Chapter 7 and reviews the advantages and limitations of the pose recovery system. §8.2 suggests directions for continued research in this area, and §8.3 gives a summary of the thesis.

8.1 Discussion

Since the City Project system described in §2.2 operates sequentially, error tends to accumulate at each stage. Typical images cover roughly 1 milliradian (mrad), or $0.05°$, of view per pixel; approximate error bounds were obtained from authors of the various system stages, and can be summarized as follows. Correction of nonlinear image distortions is accurate to a few pixels, and hemispherical image registration and internal calibration are accurate to roughly 1 mrad. Lines are detected to sub-pixel accuracy, meaning roughly 0.5 mrad of error in projective lines; similarly, intersections are computed to within 0.5 mrad. The overall error in projective feature rays can then be bounded roughly at 5 mrad.

Using this approximate error bound, and under the assumption that no directional entity is accurate to more than about 2 mrad due to inherent internal calibration error, the accuracy of the pose recovery system can be assessed from experiments run on simulated data in §7.1. The rotational pose error bound is approximately $0.05°$, and the translational error bound is approximately 1 centimeter (cm).

Without careful surveying of ground-truth 3-D measurements, it is difficult to quantitatively judge the true system performance on real data. Uncertainty bounds obtained in §7.2 imply slightly lower precision than theoretically expected (roughly $0.1°$ and 5 centimeters), though the bounds

themselves are somewhat conservative. Qualitative metrics such as angles between vanishing points and individually-examined epipolar geometry suggest that the pose recovery system produces orientations and positions sufficiently accurate for metric scene reconstruction. The next two sections summarize other notable properties of the system.

8.1.1 System Benefits

The pose recovery system developed in this thesis addresses and overcomes many shortcomings of previous approaches to camera registration. As evidenced by the results in Chapter 7, the system is robust, handling significant occlusion and exhibiting low sensitivity to outliers; in addition, it can correct substantial error in initial pose. Use of gradient-based features and hemispherical imagery reduces sensitivity to view changes (e.g. illumination and perspective effects). Fusing thousands of noisy features into a small number of more accurate summarizing entities increases both robustness and efficiency.

The system is also scalable. Orientation estimation is $\mathcal{O}(n)$ in the number of cameras and line features per camera; position estimation is $\mathcal{O}(n)$ in the number of cameras (assuming a constant number of neighbors) and $\mathcal{O}(n^2)$ in the number of point features per camera. In practice, position recovery is closer to linear in the number of features because of the geometric constraints on correspondence described in §6.1.2.

For a data set consisting of 100 nodes and 230 node adjacencies, it would take a human operator 10 hours to specify the minimum number of correspondences necessary to register the cameras (i.e. 5 per pair), assuming that identification and correspondence requires roughly 30 seconds per feature. The automated system required approximately 4 hours to recover the camera pose in such a data set and incorporated thousands of features per pair, making it both faster and more stable than manual techniques.

Unlike projective methods, the system recovers metric Euclidean orientation and position, and unlike typical structure from motion methods, explicit correspondence is neither required nor produced. The system also demonstrates end-to-end integration and *automatically* obtains globally accurate pose estimates, distributing error equally throughout the configuration. Models of uncertainty are formulated and incorporated to describe all geometric entities stochastically, and to perform all estimation tasks using probabilistic inference.

8.1.2 System Limitations

Of course, the system does fall somewhat short of ideal. One limitation is the requirement of approximate initial pose; correspondence cannot generally be established among completely arbitrary, unordered images. In essence, all vision techniques impose this requirement, which hinders the development of a truly generalized vision system. Another limitation is that the methods assume perfect internal calibration, when in reality calibration is accurate only to within a few pixels and most generally not known at all. Other techniques exist which estimate pose without internal calibration (see §2.1.5), though such techniques cannot recover metric extrinsic pose or structure.

The translation recovery technique relies on local (pair-wise) motion estimates and correspondence; this method is somewhat unstable when pose configurations are degenerate (see §6.4.1), and also does not enforce any notion of multi-camera feature consistency. Improvements could include three-camera rather than two-camera local motion estimation, which would stabilize the

pose configuration by triangulating the topology and providing greater match redundancy. The pair-wise technique is also $\mathcal{O}(n^2)$ in the number of features, though sparse matrices and match culling can reduce this to $\mathcal{O}(n)$.

The techniques are limited to scenes containing sufficient regular structure (i.e. line parallelism) and thus cannot be applied to more general scenes. For example, it is impossible to recover accurate orientation in natural environments, or in urban environments consisting only of domed or cylindrical buildings, using these methods.

Because no ground-truth scene geometry is initially available, it is also currently impossible to establish which cameras truly view overlapping geometry. Determination of camera adjacency presented in §4.4 is an overly simplistic approximation, and has several failings; there are many situations in which cameras may be spatially adjacent, but may not in fact view any common scene structure (Figure 8-1).

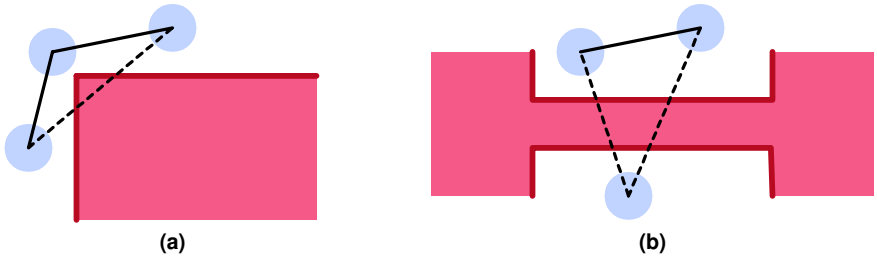


Figure 8-1: Limitations of Adjacency Computation
Examples in which adjacency does not imply coherence of view, shown in 2-D. (a) Cameras at a building corner view little or no common geometry. (b) Cameras on opposite sides of a narrow building are not likely to have common scene structure in view.

Due to curvature of the Earth's surface, the scope of orientation estimation is inherently limited to relatively small geographic areas (Figure 8-2). The Earth's radius is approximately $6,400$ kilometers; thus, for example, directions of vertical lines further apart than roughly 23 kilometers will differ by more than $0.2°$. Vanishing point correspondence and rotation techniques that attempt to match such lines under the assumption that they are truly parallel will fail or suffer from limited accuracy.

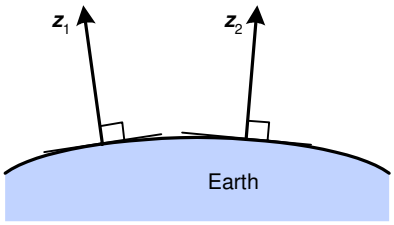


Figure 8-2: Earth Curvature
"Vertical" is defined as perpendicular to the Earth's surface; vertical directions separated by a large distance are thus not parallel to each other.

Several less fundamental problems were encountered in practice. One is that the system relies on "good" feature sets, and thus exhibits sensitivity to the specific algorithms used to extract

features. Parameters of these algorithms must be adjusted so that all salient features are detected, especially for two-camera translational alignment which requires a large degree of feature overlap. Rotational registration is less sensitive, since correspondence is established between *classes* of lines (vanishing points) rather than individual lines.

Drift and impulsive noise in Argus also presented difficulties in baseline estimation. Cameras within a few meters of each other were sometimes displaced by up to 8 meters from their true positions, causing wildly inaccurate initial baseline direction estimates. This problem has yet to be resolved, though presumably it is a practical issue involving the acquisition platform and not a theoretical limitation of these methods, which assume sufficiently accurate initial pose is available.

8.2 Future Work

By no means does this thesis represent an exhaustive treatment of robust pose recovery; nor does the final system represent the sole path traveled in course of this work. Many other possible avenues exist, some of which were partially investigated as part of this work, and some of which yet remain to be explored. This section outlines a few of these extensions and alternative methods.

8.2.1 Optimizations

While many system components were designed and implemented with computational efficiency in mind, this was not the main focus for the overall system. Perhaps the most straightforward improvements that can be made thus deal with performance—there are several ways in which the current system can be optimized.

One method for improving performance is to simply compile and run the code on a faster computer; most data was processed on a 250 MHz machine, which is somewhat slow by modern standards. Another is to exploit the fact that although many tasks in the system are sequential, relying on output from preceding stages, many also perform sets of independent operations. Running time can then be drastically reduced through high-level parallelization. For example, vanishing points can be estimated independently for each camera, and baseline directions can be estimated independently for each relevant camera pair. As shown in the previous chapter, these two tasks consume nearly all CPU time required for pose recovery.

In §6.3.6 it was mentioned that the $M \times N$ match matrices for baseline estimation are sparse, consisting of at most $\min(M, N)$ non-zero entries. The matrix of global position constraints in §6.4.1 is also quite sparse; the majority of its rows contain only 4 non-zero entries. Efficient sparse routines for manipulation of these matrices can be employed to decrease both memory usage and CPU time.

8.2.2 Improvements to Uncertainty Models

There are several areas for improvement involving uncertainty. In the City Project, Argus produces approximate estimates of position and heading by fusing multiple sensor measurements in a Kalman filter. As a result, covariance matrices are also produced alongside the pose. This information is not currently utilized by any subsequent system stage, but could (and should) be incorporated in the form of prior distributions and uncertainty bounds.

In addition, this pose recovery system produces measures of uncertainty in orientation estimates, represented by parameters of a Bingham distribution on $\mathbb{S}^3$, and in position estimates, represented by parameters of a Gaussian distribution on $\mathbb{R}^3$. Geometric elements transformed by these estimates—for example, scene points expressed in camera coordinates via stochastic rotation and translation—accumulate currently unmodeled uncertainty. A formulation for computation of this uncertainty is developed below.

Rotation of unit vectors on the sphere is quite a common operation (e.g. rotation of line directions for use in baseline recovery), as is rotation of Euclidean scene points (e.g. transformation to camera coordinates). The deterministic rotation of a Bingham random variable results in a new Bingham variable with rotated parameter matrix: if $\boldsymbol{x}$ is a Bingham variable with distribution $\mathcal{B}_3(\boldsymbol{x}; \boldsymbol{M})$, and $\boldsymbol{y} = \boldsymbol{R}\boldsymbol{x}$ where $\boldsymbol{R}$ is a deterministic rotation, then $\boldsymbol{y}$ is also Bingham with distribution $\mathcal{B}_3(\boldsymbol{y}; \boldsymbol{R}\boldsymbol{M}\boldsymbol{R}^\top)$. Similarly, deterministic rotation of a trivariate Gaussian random variable $\mathcal{N}_3(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Lambda})$ results in a new Gaussian $\mathcal{N}_3(\boldsymbol{y}; \boldsymbol{R}\boldsymbol{\mu}, \boldsymbol{R}\boldsymbol{\Lambda}\boldsymbol{R}^\top)$. In either case, the "magnitude" of the uncertainty is unchanged because $\boldsymbol{R}$ is a unitary matrix, only affecting the orientation of the resulting distributions.

Stochastic rotations can be represented as Bingham random variables on the space of unit quaternions, namely the surface of the 4-D hypersphere (§3.4.1). Using a Bayesian approach, the distribution of an uncertain direction or point $\boldsymbol{x}$ transformed by an uncertain rotation $\boldsymbol{q}$ can be formulated as an integral over conditional densities. Specifically, if $\boldsymbol{x}$ is a direction described by a Bingham distribution $\mathcal{B}_3(\boldsymbol{x}; \boldsymbol{M})$, then given a particular sample value of quaternion $\boldsymbol{q}$, the density of $\boldsymbol{x}$ is given by

$$p(\boldsymbol{x}|\boldsymbol{q}) = \mathcal{B}_3(\boldsymbol{x}; \boldsymbol{R}(\boldsymbol{q})\boldsymbol{M}\boldsymbol{R}^\top(\boldsymbol{q})) \tag{8-1}$$

where $R(\boldsymbol{q})$ is the 3×3 rotation matrix induced by the quaternion $\boldsymbol{q}$. Then the Bayesian estimate for the distribution of $\boldsymbol{x}$ is by the integral of (8-1) with respect to the quaternion,

$$\begin{aligned} p(\boldsymbol{x}) &= \int_{\mathbb{S}^3} p(\boldsymbol{x}|\boldsymbol{q}) p(\boldsymbol{q}) \mathrm{d}\boldsymbol{q} \\ &= \int_{\mathbb{S}^3} \mathcal{B}_3(\boldsymbol{x}; \boldsymbol{R}(\boldsymbol{q})\boldsymbol{M}\boldsymbol{R}(\boldsymbol{q})^\top) \mathcal{B}_4(\boldsymbol{q}; \boldsymbol{M}_{\boldsymbol{q}}) \mathrm{d}\boldsymbol{q}. \end{aligned} \tag{8-2}$$

Similar arguments can be applied to the Euclidean case, in which the resulting Bayesian integral is

$$\int_{\mathbb{R}^3} \mathcal{N}_3(\boldsymbol{x}; \boldsymbol{R}(\boldsymbol{q})\boldsymbol{\mu}, \boldsymbol{R}(\boldsymbol{q})\boldsymbol{\Lambda}\boldsymbol{R}(\boldsymbol{q})^\top) \mathcal{B}_4(\boldsymbol{q}; \boldsymbol{M}_{\boldsymbol{q}}) \mathrm{d}\boldsymbol{q}. \tag{8-3}$$

Analytic evaluation of these expressions seems unlikely, though approximation methods (e.g. linearizations) may exist which produce random variables of Bingham or Gaussian form.

Translation occurs in Euclidean 3-D space, and does not affect directional quantities; uncertainty in Bingham random variables thus remains unchanged after probabilistic translation. Stochastic 3-D points $\boldsymbol{x}$, however, are affected. Assuming that the translation $\boldsymbol{t}$ is represented by a trivariate distribution $\mathcal{N}_3(\boldsymbol{t}; \boldsymbol{\mu}_{\boldsymbol{t}}, \boldsymbol{\Lambda}_{\boldsymbol{t}})$, and that $\boldsymbol{x}$ and $\boldsymbol{t}$ are independent, the distribution of $\boldsymbol{y} \equiv \boldsymbol{x} + \boldsymbol{t}$ can be represented simply as $\mathcal{N}_3(\boldsymbol{y}; \boldsymbol{\mu}_{\boldsymbol{x}} + \boldsymbol{\mu}_{\boldsymbol{t}}, \boldsymbol{\Lambda}_{\boldsymbol{x}} + \boldsymbol{\Lambda}_{\boldsymbol{t}})$, since a linear combination of Gaussians is also Gaussian.

8.2.3 Internal Calibration

All methods presented as part of this thesis rely on accurately known internal camera parameters. In truth, these parameters are imprecise, and associated uncertainty is not explicitly modeled. Rather, it is accounted for implicitly along with all other unmodeled error sources by the Bingham densities resulting from various inference tasks. Explicit uncertainty models could further improve pose estimation and subsequent processing.

Various authors have also proposed methods for internal calibration solely from vanishing points [Ech89, CT90, WT91, LZ98], some including error analysis [Kan92]. The problem is inherently underconstrained, so the one common assumption of all existing techniques is that metric 3-D angles between the vanishing points are known *a priori*. This assumption is sensible in the context of the City Project, in which rectilinear scene structures are common.

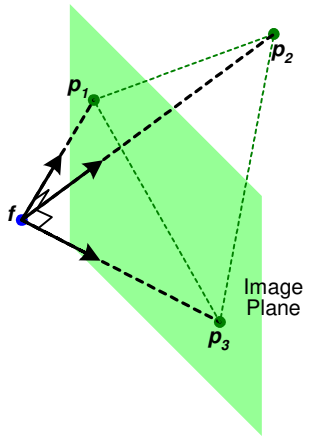


Figure 8-3: Internal Parameters from Vanishing Points
Orthogonal vanishing points constrain the position of the focal point relative to the image plane. In particular, rays from the focal point to vanishing point projections must be orthogonal.

The position of the optical center $\boldsymbol{f}$ expressed in image coordinates can be obtained analytically using vanishing points estimated from three mutually orthogonal line sets, shown in Figure 8-3. If an arbitrary optical center is initially assumed (e.g. $-\hat{\boldsymbol{z}}$), then image observations can be transformed into projective points and the estimation methods of §5.3 can be applied. Let $\boldsymbol{p}_1$, $\boldsymbol{p}_2$, and $\boldsymbol{p}_3$ denote 3-D points on each of three orthogonal vanishing point rays; then the nonlinear constraint equations are

$$\begin{aligned} (\boldsymbol{f}-\boldsymbol{p}_1)\cdot(\boldsymbol{f}-\boldsymbol{p}_2) &= 0 \\ (\boldsymbol{f}-\boldsymbol{p}_2)\cdot(\boldsymbol{f}-\boldsymbol{p}_3) &= 0 \\ (\boldsymbol{f}-\boldsymbol{p}_3)\cdot(\boldsymbol{f}-\boldsymbol{p}_1) &= 0, \end{aligned} \tag{8-4}$$

which can be solved in closed form [Ant96]. There are in fact two solutions, one that places the focal point in front of the image plane and one behind, corresponding to the inverting and non-inverting pinhole models of Figure 1-2. Orthogonality is not a strict requirement; any configuration suffices in which angles between all vanishing points are known, but non-zero dot products slightly

change the form of (8-4). Vanishing points at infinity (i.e. parallel to the image) prevent unique solution.

A hemispherical image, or node, in the City Project consists of a set of images acquired at different rotations around a single optical center. Internal parameters could be estimated per node by fusing several estimates of $\boldsymbol{f}$ from each of the node's constituent rectangular images. Further, if the parameters are assumed not to vary from one node to another, estimates from multiple nodes could be fused.

8.2.4 Exclusive Use of Line Features

It is well established that line features are inherently more reliable than point features [TK92]. A given line consists of many observations (namely all visible points on the line) rather than a single observation, making estimation of its parameters more robust, and also allowing the same line to be detected across multiple views even when it is partially occluded. Point features derived from line intersections capture some of this robustness, but occlusion is still more likely. Points are also lower in the feature hierarchy mentioned in §1.3, so are more susceptible to error accumulation.

Line features are thus preferable to point features for SFM and other 3-D vision applications. Several authors have shown that lines alone can be used to fully determine geometry and pose [TK92]. In the context of this work, line features are further preferable because their associated 3-D directions are determined during rotational pose alignment, leaving only one degree of freedom per observation (namely its depth along a particular 3-D view ray).

The space of line correspondences can be reduced somewhat by exploiting the 3-D direction classification obtained from rotational pose estimation. Lines from different groups (i.e. in different 3-D directions) cannot possibly match the same 3-D geometry or each other, so feature matching can be performed on each group independently. If infinite lines rather than finite segments are used as features, then lines in a particular direction $\boldsymbol{d}$ cannot be used to distinguish camera motions parallel to $\boldsymbol{d}$; this suggests a decoupling of position estimation into components, each corresponding to a particular known 3-D direction.

Specifically, all lines in a given direction $\boldsymbol{d}_j$ and all cameras can be projected onto the plane orthogonal to $\boldsymbol{d}_j$ in a 2-D structure from motion formulation, reducing degrees of freedom to two per camera [TKA91]. Line features then become planar point features, and observation rays have only one degree of freedom (namely their orientation θ in the plane). Correspondence can be further constrained in this context by using the approximately known camera pose, which limits searches to small planar regions.

Several avenues for estimation of camera position using lines alone were explored as part of this thesis. One method attempted to establish correspondence by searching for high-incidence regions in the plane. If a given 2-D point feature (i.e. 3-D line projected onto the plane) is observed by multiple cameras, then in the absence of noise, all observation rays should intersect precisely at that point. Thus, the planar surface was discretized into cells, and every relevant ray (corresponding to all line features parallel to $\boldsymbol{d}_j$) was extruded from each camera center and accumulated. This method did not work well in practice, because discretization caused artificially high incidence in regions near the cameras. Also, because features were not sufficiently persistent across views and because the of initial camera misregistration, true incidence peaks were difficult to distinguish from spurious peaks.

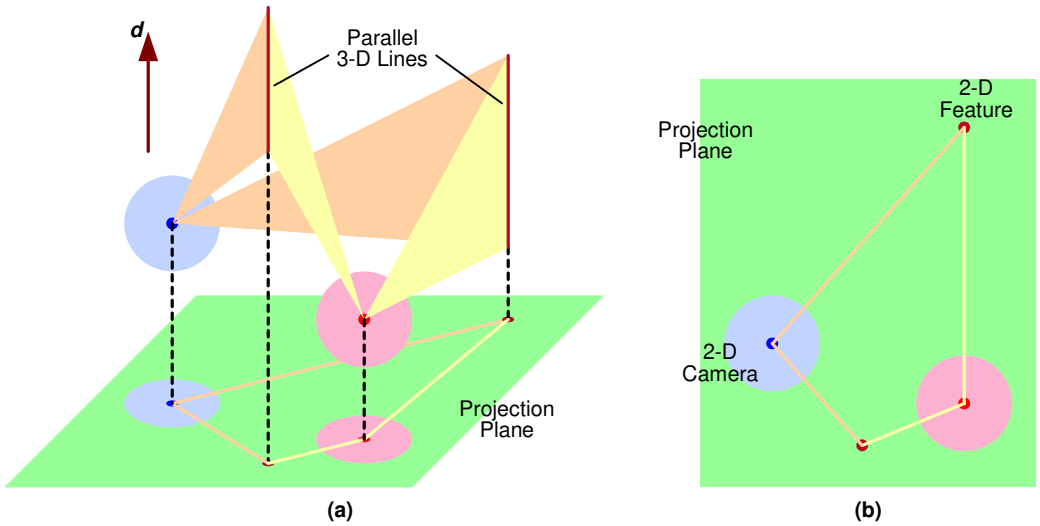


Figure 8-4: Two Dimensional Structure from Motion
(a) A configuration in which two cameras view two parallel 3-D line features. Motion in the direction of the lines cannot be determined, so the entire configuration can be projected onto a plane orthogonal to this direction. (b) The projected configuration.

Another more promising method was partially developed, similar in spirit to the position recovery techniques of Chapter 6 in which local translation estimates were assembled to deduce a global pose configuration. It can be shown that at least three cameras are necessary to uniquely determine pose configurations (up to arbitrary scale) in planar SFM problems [TKA91]; thus three-way rather than two-way correspondences were considered. Correspondence triplets allow local registration of camera triplets, resulting in more stable global camera configurations.

The proposed technique used a local three-camera method similar to the discretized two-camera translation technique of §6.2.4. If arbitrary global translation and scale are imposed, then each camera triplet in the plane has three degrees of freedom, shown in Figure 8-5: the first camera is held fixed, the second camera is unit distance from the first, and the third camera has arbitrary position, so the parameters are (θ, x, y).

Each correct three-way correspondence imposes a single constraint on these parameters via the common intersection of all corresponding observation rays (Figure 8-5). Let θ_1, θ_2, and θ_3 denote the angles formed by the three observation rays in a particular triplet. This correspondence imposes a single constraint of the form

$$\alpha_1 x + \alpha_2 y + \alpha_3 \cos\theta + \alpha_4 \sin\theta = 0 \tag{8-5}$$

where the α_i are trigonometric functions of the θ_i (derivation omitted). This constraint can be used in a Hough transform with 3-D parameter space $\theta \in [-\pi, \pi]$, $x \in \mathcal{R}$, and $y \in \mathcal{R}$; the parameter space can also be restricted if approximate pose is available. Constraint surfaces are nonlinear, but the proximity function of §4.1.6 can be used to compute accumulation values. An example constraint surface is shown in Figure 8-6.

Correct correspondence is not known; however, the accumulation of constraint surfaces for *all possible* match triples should result in a strong transform peak at the correct parameter values, just as in the two-camera case of §6.2.4.

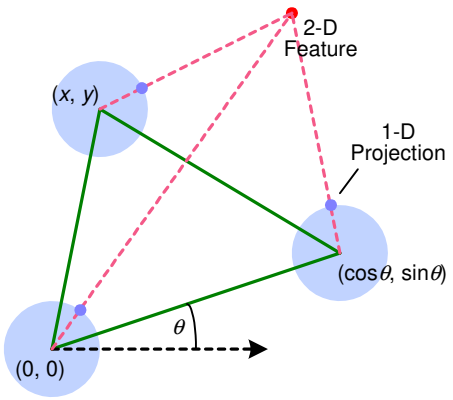


Figure 8-5: Camera Triplet Geometry
Configurations of three cameras in 2-D can be summarized by three parameters. One camera is taken to be the origin, and the second is taken to be unit distance from the first. The position of the third camera is expressed as a 2-D translation.

This formulation has two main drawbacks. The first is that the computational requirements for a 3-D Hough transform are fairly large; a great deal of memory and time is required for surface accumulation, peak finding, etc. The second is that enumeration of all possible three-way matches is $\mathcal{O}(n^3)$ in the number of features, which is prohibitively high even for moderately large feature sets.

8.2.5 Domain-Specific Constraints

In order to maintain generality, few domain-specific constraints are imposed in this thesis. If automatic pose recovery techniques are to be applied exclusively to urban environments, however, several simplifying assumptions can be made.

One such assumption is that environments consist mainly of vertical (orthogonal to the ground) and horizontal (parallel to the ground) line directions. In practice, prominent vanishing points are seldom observed in other directions; some examples include chain-link fences, ramped parking structures, and slanted rooftops, but these are not generally consistent across views. Constraining line directions simplifies vanishing point estimation considerably; for example, every image can be assumed to view vertical lines, and detection of the remaining line directions then reduces to a one-dimensional problem (i.e. estimation of horizontal line azimuths).

Another simplification is to assume that environments consist mainly of planar surfaces, and further, that these surfaces are vertically oriented. Urban scenes often consist of large vertical façades, which can serve as low-dimensional features in the recovery of camera position. One of the attractive properties of rotational pose estimation is that the process requires a relatively small number of features, as opposed to thousands of individual lines and points. Analogous formulations may be applicable to position estimation: if point features could be segmented into groups, where each group represents a particular vertical plane, then these *classes* of points, rather than individual points, could be used as features. Planes have only three degrees of freedom, of which two (namely the orientation) can be inferred immediately: since 3-D line directions are known, and since point features represent intersections of lines, the normal to a plane containing a

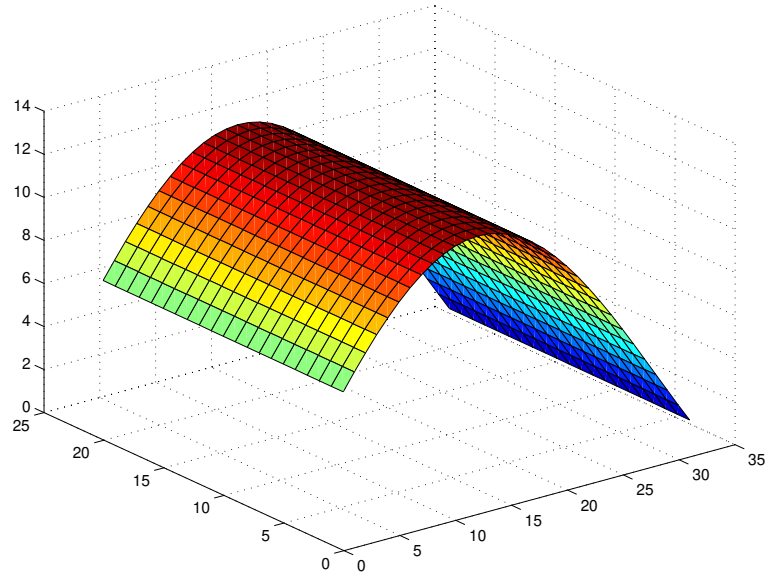


Figure 8-6: Three-Way Match Constraint Surface
Each three-way correspondence in the 3-D parameter space of camera triples induces a 2-D surface. An example of such a surface is depicted.

given point can be taken as the cross product of the directions of the constituent 3-D lines (Figure 8-7).

Another attractive property of rotational pose estimation is that each camera is individually rotated to scene-relative entities (namely 3-D line directions), thus simultaneously producing accurate descriptions of these entities and unbiased pose estimates. If planar surfaces were parameterized as above, it would be possible to hypothesize their 3-D locations and align each camera to these scene-relative entities in an iterative scheme similar to that of §5.5. This would result in a globally-consistent, unbiased set of camera positions, as well as estimates of significant planar geometry.

8.2.6 Correspondence Propagation

The translational registration techniques presented in this work operate only on two-camera correspondence. If multi-camera correspondence were available, then global bundle adjustment or other standard SFM algorithms could be applied to further stabilize the pose estimates.

Although deterministic correspondence is never established by these techniques, the probabilistic match matrices $\boldsymbol{W}$ from §6.2.3 can be heuristically transformed into binary matrices; for example, any entry w_{ij} greater than a threshold value can be assumed to imply an unambiguous correspondence. The set of all such correspondences can then be enumerated for each relevant camera pair.

For consistent matching across multiple cameras, each point feature must be labeled with a unique identifier, such as a sequential camera number and feature number. This allows the point to be identified as the same entity across multiple camera pairs. Data structures and methods for determination of multi-camera correspondence from two-camera correspondence can grow to be

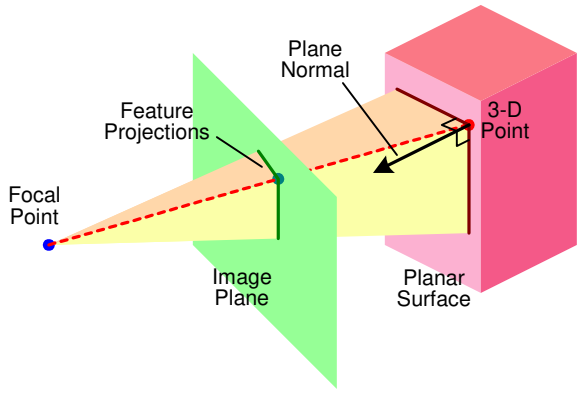


Figure 8-7: Plane Orientations
The normal to a 3-D planar surface can be computed as the vector orthogonal to two lines in the plane with known directions. If these lines intersect at a point and the intersection is detected in the image, the plane normal can be obtained from the cross product of the 3-D directions of the point's constituent line segments.

quite complex; one brute-force approach is similar to that of vanishing point correspondence in §5.5.3, where a global list of unique point features and their associated image observations evolves as the camera adjacency graph is traversed.

In particular, the constituent features of each binary correspondence in a given arc of the adjacency graph are enumerated and added to a global list. The graph is then traversed, for example in depth-first order, and further correspondending features are appended as they are encountered. Problems that arise on first examination, besides complexity of implementation and scale issues, include redundant features (i.e. separate global features which correspond to the same 3-D entity) and lack of feature coherence across multiple views. The sheer number of correspondences, however, might provide enough redundancy to produce solid pose estimates.

8.2.7 Extension to Real-Time

All techniques developed in this work operate offline; that is, they assume all data to be immediately available and estimate camera pose as a post-process. Recently there has been considerable interest in online vision systems, such as augmented reality and autonomous navigation, motivating examination of this work in the context of real-time applications.

In this context, it is convenient to view images as having been obtained from a single camera at sequential times rather than from multiple cameras in a fixed configuration. Images are also ordered, so that even if approximate pose or inertial estimates are unavailable, the strong assumption of small inter-image camera motion can be imposed.

Small baselines would facilitate local correspondence of vanishing points and provide strong geometric constraints on two-camera point matching, so pose could be recovered accurately over short image sequences. However, without simultaneous incorporation of all pose information or unbiased knowledge of the overall motion, the system would suffer from drift and error propagation artifacts characteristic of all purely local SFM techniques. An extended Kalman filter that computes incremental updates to the camera's orientation and position could reduce drift, but

would drastically alter the uncertainty formulation, and still not necessarily produce metric camera alignment.

Real-time implementations involve tradeoffs between estimation accuracy and speed. Using a sequence of discretized and continuous techniques has the advantage that it provides a natural boundary between the two extremes; estimation can be performed quickly (though approximately) for real-time applications using the Hough transform alone. In particular, assuming that lines and points could be detected in real time, vanishing points and baseline directions, which require the most significant computation, could be established quickly, at the expense of accuracy and tight error bounds.

8.2.8 Topological Analysis

A final improvement to these techniques involves further analysis of the initial pose configuration. Currently, no automatic high-level partitioning of the global configuration occurs; cameras are manually segmented into meaningful groups (e.g. by spatial proximity), since registration and other tasks become simpler and faster when performed on smaller subsets of cameras. The graph of camera topology can be examined to automatically perform geographic segmentation, and to detect "disconnected" components—that is, sub-graphs of cameras not adjacent to any other cameras.

Examination of the camera configuration could also produce likely locations of major structures in the scene. Assuming sufficient camera density, large regions of "empty space" signify occupancy by significantly large geometry (e.g. buildings); this is illustrated for a real pose configuration in Figure 8-8. If these regions can be identified, then determination of camera adjacency can be greatly improved. For example, no adjacency would be created for cameras on opposite sides of a structure (i.e. whose arc passes through occupied space); connected paths around the objects could also be created.

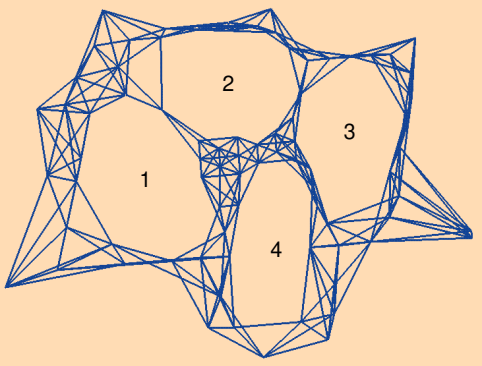


Figure 8-8: Topological Holes
Regions of empty space in the camera configuration—i.e. large contiguous regions that contain no cameras—are likely to contain prominent 3-D structure. Here, the approximate locations of four buildings are evident.

8.3 Summary

This dissertation has presented a novel extrinsic pose recovery system able to automatically register a large number of cameras. The system relies solely on discrete image features and stochastic geometric inference to produce accurate pose estimates and bounds on their uncertainty. The system makes use of discretized techniques for robust initialization and formulation of prior distributions, as well as continuous parameter estimation in a probabilistic framework for precision and to circumvent the need for explicit correspondence. The main assumptions required for pose recovery are reliable intrinsic calibration, approximately known position and orientation, images viewing parallel line sets, and sufficient overlap of viewed geometry.

Earlier chapters reviewed the fundamentals of image projection and geometric models, and situated this thesis both with respect to its immediate motivations (the City Project) and in the context of the larger body of existing vision research. Subsequent chapters described the main core of the work, namely the sequential recovery of camera orientations and positions, while highlighting novel aspects and contributions. The final chapters assessed the performance of the pose recovery system both stage-by-stage on simulated data and end-to-end on real data, showing it to be quite accurate and robust.

Uncertainty models and stochastic inference were used extensively in this work in order to incorporate as much information as possible in parameter estimation tasks, and to provide and propagate confidence measures for recovered quantities. The Bingham density on $\mathbb{S}^2$ was used to describe all projective entities, while the Bingham density on $\mathbb{S}^3$ described rotational uncertainty. General projective inference techniques for the fusion of uncertain measurements were developed, for example to estimate projective line features from image gradients.

Novel interpretations, implementations, and applications of the Hough transform were introduced as part of this work. The transform was shown to be a powerful, general tool for robust parameter estimation and optimization problems, especially when combined with more accurate non-discretized approaches. A reliable peak detection method was presented, as was a proximity function that compensates for data uncertainty and aliasing effects. Discretization of the surface of the unit sphere was achieved by sampling three faces of a unit cube, and adjacency computation across faces was facilitated by a padding technique.

The Hough transform, in combination with a Bingham mixture model and an expectation maximization algorithm, was used to detect vanishing points from 2-D line observations. Accurate estimates were produced per camera via projective data fusion, then aligned across cameras to produce globally consistent rotational pose. A classical two-camera rotational registration technique was extended in three ways: first, it was interpreted as an inference problem on the 4-D hypersphere of unit quaternions with deterministic samples; second, uncertainty in the samples themselves was formulated and incorporated by Bayesian integration; and third, the method was generalized to handle an arbitrary number of cameras. A bootstrapping technique for determination of local vanishing point correspondence was used to initialize a nested EM algorithm that alternately estimated probabilistic correspondence, scene-relative 3-D line directions, and stochastic camera orientations.

Orientations were used to construct a simple model for two-camera epipolar geometry. In this framework, inference of motion direction was shown to involve projective relationships identical to those of vanishing points. The space of two-camera feature correspondences was reduced significantly by unique geometric constraints involving known 3-D line directions and bounds on

camera positions. Approximate baseline estimates in the form of projective directions were estimated using a novel Hough transform approach that simultaneously considered all relevant correspondences. An EM algorithm was then used to alternately refine the baseline and determine the distribution over correspondence sets. This distribution was obtained using a Markov chain Monte Carlo method, which introduced a state space and several novel state perturbations to account for occlusions and outlier features.

Stochastic baseline directions were assembled in a linear least-squares formulation to determine the most consistent set of camera positions. Several soft constraints were imposed to compensate for degeneracies in camera topology that would otherwise prevent unique solutions. Projective uncertainty in the baselines was incorporated into the global optimization, and Euclidean uncertainty in the final camera positions was also extracted. Finally, a 3-D to 3-D registration technique aligned the resulting camera configuration with metric coordinates.

It is important in any engineering research to experimentally verify theoretical results. Therefore, a major concern of this work, besides providing a solid theoretical foundation, was to ensure that efficient and practical implementation of these techniques was possible, and to provide an assessment of this implementation. The larger context of the City Project also motivated end-to-end considerations and system integration. An array of different experiments was designed and used to characterize the system's behavior and to confirm that accurate pose could be recovered. High-level behavior of the system as part of the larger project has yet to be characterized; this behavior, as well as generalization to new types of data sets, are the subjects of ongoing investigation.

Metric pose recovery is a highly computational task, involving primarily quantitative rather than qualitative reasoning and requiring precision in the description of both the pose itself and its reliability. The task is thus somewhat ill-suited for biological vision systems, but falls naturally into the domain of computational machines—given, of course, that an appropriate set of algorithmic tools is available. Although fully autonomous vision systems continue to elude the efforts of modern research, it is the author's sincere hope that this work will provide a solid stepping stone for future research, and help to pave the way toward generalized machine vision.

APPENDIX A

Quaternions

ROTATION OF VECTORS IN THREE DIMENSIONS is a common operation, in most cases involving the application of a 3×3 linear transformation $\boldsymbol{R}$. Rotation parameters do not lie in linear spaces, however; implicit constraints on the matrix $\boldsymbol{R}$, namely that all its columns are orthogonal and of unit length, make parameterizations highly nonlinear. Many other representations have therefore been devised for convenient manipulations of rotational quantities in various contexts.

The *unit quaternion* (or simply *quaternion*) $\boldsymbol{q}$ is possibly the most elegant such representation, and, in the context of this thesis, the most convenient. The following sections describe various manipulations of quaternions in more detail.

A.1 Definition

Quaternions consist of four parameters,

$$\boldsymbol{q} = (w, x, y, z)^\top \tag{A-1}$$

subject to the constraint that $\|\boldsymbol{q}\| = 1$, and thus can be envisioned as vectors in $\mathbb{R}^4$ lying on the surface of the unit hypersphere $\mathbb{S}^3$. They can also be described as complex numbers with a scalar real part w and a vector imaginary part $\boldsymbol{v} \in \mathbb{R}^3$, namely

$$\boldsymbol{q} = (w, \boldsymbol{v})^\top, \qquad \boldsymbol{v} = (x, y, z)^\top. \tag{A-2}$$

Quaternions are quite general in that they can also represent scalar values (i.e. $\boldsymbol{v} = \boldsymbol{0}$) and 3-D vectors (i.e. $w = 0$).

A.2 Conversions

Conversion from unit quaternions to other rotational representations is relatively straightforward. For example, to determine the angle of rotation θ and the axis $\boldsymbol{d}$ about which the rotation operates, the following relationships can be used:

$$w = \cos 2\theta \tag{A-3}$$

$$\boldsymbol{v} = \boldsymbol{d} \sin 2\theta; \tag{A-4}$$

thus,

$$\theta = \frac{1}{2} \cos^{-1} w \tag{A-5}$$

$$\boldsymbol{d} = \frac{\boldsymbol{v}}{\sin 2\theta}. \tag{A-6}$$

An orthonormal rotation matrix can also be constructed according to

$$\boldsymbol{R}(\boldsymbol{q}) = \begin{pmatrix} w^2 + x^2 - y^2 - z^2 & 2(xy - wz) & 2(xz + wy) \\ 2(xy + wz) & w^2 - x^2 + y^2 - z^2 & 2(yz - wx) \\ 2(xz - wy) & 2(yz + wx) & w^2 - x^2 - y^2 + z^2 \end{pmatrix} \tag{A-7}$$

A.3 Transformations

The conjugate (i.e. inverse) of a quaternion, using the notation of (A-2), is defined simply as

$$\bar{\boldsymbol{q}} = (w, -\boldsymbol{v}), \tag{A-8}$$

just as with complex numbers. Its magnitude can then be written as

$$\|\boldsymbol{q}\|^2 = \boldsymbol{q} \cdot \bar{\boldsymbol{q}}, \tag{A-9}$$

with the dot product defined in the usual way.

Application of a quaternion $\boldsymbol{q}_1$ to another quaternion $\boldsymbol{q}_2$ (that is, concatenation of two rotations) results in another quaternion $\boldsymbol{r}$, and can be achieved using a product operator:

$$\boldsymbol{r} = \boldsymbol{q_1} \times \boldsymbol{q_2} = ([w_1 w_2 - \boldsymbol{v}_1 \cdot \boldsymbol{v}_2], [w_1 \boldsymbol{v}_2 + w_2 \boldsymbol{v}_1 + \boldsymbol{v}_1 \times \boldsymbol{v}_2])^\top. \tag{A-10}$$

Note that this definition exhibits the expected behavior for scalar-scalar and scalar-vector multiplication, as well as for 3-D vector cross products. The operator (A-10) can also be applied using a 4×4 matrix:

$$\boldsymbol{r} = \begin{pmatrix} w_1 & -x_1 & -y_1 & -z_1 \\ x_1 & w_1 & -z_1 & y_1 \\ y_1 & z_1 & w_1 & -x_1 \\ z_1 & -y_1 & x_1 & w_1 \end{pmatrix} \boldsymbol{q}_2. \tag{A-11}$$

The product operator is compatible with magnitude, so that

$$\|\boldsymbol{q}_1 \times \boldsymbol{q}_2\| = \|\boldsymbol{q}_1\| \cdot \|\boldsymbol{q}_2\|. \tag{A-12}$$

Finally, rotation of a vector $\boldsymbol{d}$ by a quaternion $\boldsymbol{q}$ is given by the quadratic relation

$$\boldsymbol{R}(\boldsymbol{q})\boldsymbol{d} = \boldsymbol{q} \times \boldsymbol{d} \times \bar{\boldsymbol{q}}, \tag{A-13}$$

with $\boldsymbol{R}(\boldsymbol{q})$ defined as in (A-7).

A.4 Comparison of Relative Rotations

When assessing registration results, it is often necessary to directly measure the difference between two orientations. There are several methods for such comparisons; one is to compute the Euclidean angle between the quaternions $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$:

$$\theta_e = \cos^{-1}(\boldsymbol{q}_1 \cdot \boldsymbol{q}_2). \tag{A-14}$$

Since this metric has no physical meaning, however, a preferable method is to measure the angle required to rotate the coordinate frame of $\boldsymbol{q}_1$ to that of $\boldsymbol{q}_2$. The quaternions can be interpreted as rotations that take the principal axes in scene coordinates to a new coordinate system, and are thus specified relative to the same rotational reference. A new relative rotation can then be formed as

$$\boldsymbol{q}_{1\to 2} = \boldsymbol{q}_2 \times \bar{\boldsymbol{q}}_1, \tag{A-15}$$

and the rotation discrepancy angle θ_e can then be found using (A-5).

Two relative rotations $\boldsymbol{q}$ and $\boldsymbol{r}$ can themselves be compared by direct examination of the respective angles of rotation $\theta_{\boldsymbol{q}}$ and $\theta_{\boldsymbol{r}}$ from (A-5). Only the angles are necessary because relative rotations are by definition specified in arbitrary rotational reference frames; thus the axes of rotation are irrelevant and need not be compared.

A.5 Optimal Deterministic Rotation

In §5.4.1, a method was described for optimal estimation of a quaternion $\boldsymbol{q}$ that minimizes an objective function (5-10) given a set of corresponding 3-D directions $\boldsymbol{v}_j^{\mathcal{A}}$ and $\boldsymbol{v}_j^{\mathcal{B}}$. This section derives the result in more detail.

Using quaternion notation, minimization of (5-10) is equivalent to finding

$$\text{argmin}_{\boldsymbol{q}} \sum_{j=1}^{J} \|\boldsymbol{v}_j^{\mathcal{A}} - \boldsymbol{q} \times \boldsymbol{v}_j^{\mathcal{B}} \times \bar{\boldsymbol{q}}\|^2 \tag{A-16}$$

$$= \text{argmin}_{\boldsymbol{q}} \sum_{j=1}^{J} \|\boldsymbol{v}_j^{\mathcal{A}} - \boldsymbol{q} \times \boldsymbol{v}_j^{\mathcal{B}} \times \bar{\boldsymbol{q}}\|^2 \|\boldsymbol{q}\|^2, \tag{A-17}$$

since multiplication by one does not affect the minimization. Using the compatibility stated in (A-12) and the definition of quaternion magnitude in (A-9), this can be written as

$$\text{argmin}_{\boldsymbol{q}} \sum_{j=1}^{J} \|\boldsymbol{v}_j^{\mathcal{A}} \times \boldsymbol{q} - \boldsymbol{q} \times \boldsymbol{v}_j^{\mathcal{B}} \times \bar{\boldsymbol{q}} \times \boldsymbol{q}\|^2$$

$$= \text{argmin}_{\boldsymbol{q}} \sum_{j=1}^{J} \|\boldsymbol{v}_j^{\mathcal{A}} \times \boldsymbol{q} - \boldsymbol{q} \times \boldsymbol{v}_j^{\mathcal{B}}\|^2. \tag{A-18}$$

Now, using the linear definition of the product (A-11), the inner quantity can be written as a single 4×4 matrix $\boldsymbol{A}_j$ acting on $\boldsymbol{q}$, namely

$$\begin{aligned} & \text{argmin}_{\boldsymbol{q}} \sum_{j=1}^{J} \|\boldsymbol{A}_j \boldsymbol{q}\|^2 \\ = \; & \text{argmin}_{\boldsymbol{q}} \sum_{j=1}^{J} \boldsymbol{q}^\top \boldsymbol{A}_j^\top \boldsymbol{A}_j \boldsymbol{q} \\ = \; & \text{argmin}_{\boldsymbol{q}} [\boldsymbol{q} \boldsymbol{A} \boldsymbol{q}] \end{aligned} \tag{A-19}$$

which is identical to (5-12). It can be shown that this quantity can be minimized by choosing the eigenvector of $\boldsymbol{A}$ corresponding to the minimum eigenvalue.

The matrices $\boldsymbol{A}_j$ are linear functions of the corresponding vectors $\boldsymbol{v}_j^{\mathcal{A}}$ and $\boldsymbol{v}_j^{\mathcal{B}}$, and from (A-11) are expressed as

$$\boldsymbol{A}_j = \begin{pmatrix} 0 & (x_2 - x_1) & (y_2 - y_1) & (z_2 - z_1) \\ (x_1 - x_2) & 0 & (-z_1 - z_2) & (y_1 + y_2) \\ (y_1 - y_2) & (z_1 + z_2) & 0 & (-x_1 - x_2) \\ (z_1 - z_2) & (-y_1 - y_2) & (x_1 + x_2) & 0 \end{pmatrix} \tag{A-20}$$

where

$$\boldsymbol{v}_j^{\mathcal{A}} = (x_1, y_1, z_1)^\top, \qquad \boldsymbol{v}_j^{\mathcal{B}} = (x_2, y_2, z_2)^\top. \tag{A-21}$$

The same result can be obtained by taking second derivatives of (A-16) with respect to $\boldsymbol{q}$.

A final note is that, in order for $\boldsymbol{A}_j^\top \boldsymbol{A}_j$ to be a valid second moment matrix, its eigenvalues must sum to one. Thus, it must be normalized by its trace, which in this case is 8.

A.6 Incorporation of Uncertainty

If $\boldsymbol{v}_j^{\mathcal{A}}$ and $\boldsymbol{v}_j^{\mathcal{B}}$ represent not deterministic samples but random variables with bipolar Bingham distributions, computation of the appropriate distribution on the correspondence quaternion $\boldsymbol{q}_j$ is not so straightforward as in the previous section. This computation requires evaluation of the Bayesian integral (5-13); however, a closed-form expression for (5-13) seems unlikely. Instead, the distribution of $\boldsymbol{q}_j$ is assumed to be Bingham on $\mathbb{S}^3$ and a different integral is evaluated. The basic idea is to first find what amounts to a sample second moment matrix $\boldsymbol{S}_j$, then transform this matrix to a 4×4 Bingham parameter matrix $\boldsymbol{M}_j$ using the method of §B.2.

This second moment, rather than the parameter matrix, is computed as a weighted average over all possible samples of $\boldsymbol{v}_j^{\mathcal{A}}$ and $\boldsymbol{v}_j^{\mathcal{B}}$. The samples are assumed independent, though they can *not* be treated as axial quantities because rotations can only be computed for directed vectors; rotations from antipodal points would effectively cancel each other out in the average. Therefore, the original $\boldsymbol{v}$ are used to form new variables $\tilde{\boldsymbol{v}}$, each of which has an asymmetric, directed probability distribution whose mode is aligned with the correctly signed polar axis of $\boldsymbol{v}$, and whose covariance is equal to that of the original antipodally symmetric Bingham distribution. A mean vector $\langle \tilde{\boldsymbol{v}} \rangle$ (as

defined in §3.1.2) can be computed via integration of this new distribution; the mean vector is also aligned with $\boldsymbol{v}$, but its magnitude may be smaller than one.

The desired second moment matrix is then given by

$$\boldsymbol{S}_j = \int_{\mathbb{S}^2}\int_{\mathbb{S}^2} \boldsymbol{S}_j^0 p(\tilde{\boldsymbol{v}}_j^{\mathcal{A}}) p(\tilde{\boldsymbol{v}}_j^{\mathcal{B}}) \mathrm{d}\tilde{\boldsymbol{v}}_j^{\mathcal{A}} \mathrm{d}\tilde{\boldsymbol{v}}_j^{\mathcal{B}}. \tag{A-22}$$

where $\boldsymbol{S}_j^0$, the matrix $\boldsymbol{A}_j^\top \boldsymbol{A}_j$ for particular sample values of the constituent directions, can be expanded into an expression exclusively involving the cross terms x_1y_2, x_2y_1, y_1z_2, y_2z_1, x_1z_2, and x_2z_1. Evaluation of (A-22) can be performed component by component—that is, each entry of $\boldsymbol{S}_j$ can be evaluated separately. Further, each component can be split into a sum of terms of the form

$$\int_{\mathbb{S}^2}\int_{\mathbb{S}^2} t p(\tilde{\boldsymbol{v}}_j^{\mathcal{A}}) p(\tilde{\boldsymbol{v}}_j^{\mathcal{B}}) \mathrm{d}\tilde{\boldsymbol{v}}_j^{\mathcal{A}} \mathrm{d}\tilde{\boldsymbol{v}}_j^{\mathcal{B}}, \tag{A-23}$$

where t is a cross term as described above. Due to the independence of constituent variables $\tilde{\boldsymbol{v}}_j^{\mathcal{A}}$ and $\tilde{\boldsymbol{v}}_j^{\mathcal{B}}$, these terms simplify still further; for example, for $t = x_1y_2$,

$$\int_{\mathbb{S}^2}\int_{\mathbb{S}^2} x_1y_2 p(\tilde{\boldsymbol{v}}_j^{\mathcal{A}}) p(\tilde{\boldsymbol{v}}_j^{\mathcal{B}}) \mathrm{d}\tilde{\boldsymbol{v}}_j^{\mathcal{A}} \mathrm{d}\tilde{\boldsymbol{v}}_j^{\mathcal{B}} = \int_{\mathbb{S}^2} x_1 p(\tilde{\boldsymbol{v}}_j^{\mathcal{A}}) \mathrm{d}\tilde{\boldsymbol{v}}_j^{\mathcal{A}} \int_{\mathbb{S}^2} y_2 p(\tilde{\boldsymbol{v}}_j^{\mathcal{B}}) \mathrm{d}\tilde{\boldsymbol{v}}_j^{\mathcal{B}}. \tag{A-24}$$

This is just the product of the $\hat{\boldsymbol{x}}$ component of the mean of $\tilde{\boldsymbol{v}}_j^{\mathcal{A}}$ with the $\hat{\boldsymbol{y}}$ component of the mean of $\tilde{\boldsymbol{v}}_j^{\mathcal{B}}$.

Let $\langle\tilde{\boldsymbol{v}}\rangle = k(x, y, z)^\top$, where $0 \le k \le 1$, and let $c = (k_1k_2)^{-1}$. Then the final expression for $\boldsymbol{S}_j$ is

$$\boldsymbol{S}_j = \frac{1}{c}\begin{pmatrix} (c - x_1x_2 - y_1y_2 - z_1z_2) & (y_1z_2 - y_2z_1) \\ (y_1z_2 - y_2z_1) & (c - x_1x_2 + y_1y_2 + z_1z_2) \\ (x_2z_1 - x_1z_2) & -(x_1y_2 + x_2y_1) \\ (x_1y_2 - x_2y_1) & -(x_1z_2 + x_2z_1) \end{pmatrix} \cdots$$

$$\cdots \begin{pmatrix} (x_2z_1 - x_1z_2) & (x_1y_2 - x_2y_1) \\ -(x_1y_2 + x_2y_1) & -(x_1z_2 + x_2z_1) \\ (c + x_1x_2 - y_1y_2 + z_1z_2) & -(y_1z_2 + y_2z_1) \\ -(y_1z_2 + y_2z_1) & (c + x_1x_2 + y_1y_2 - z_1z_2) \end{pmatrix}, \tag{A-25}$$

which is identical in form to the deterministic $\boldsymbol{A}_j^\top \boldsymbol{A}_j$, except that each of the two constituent directions has been scaled by the length of its mean.

APPENDIX B

Bingham's Distribution

A FLEXIBLE PROBABILITY DENSITY FUNCTION for describing random variables on the unit sphere was proposed by Bingham [Bin74] and subsequently analyzed in various contexts [Mar72, Wat83, Ken87, CW90]. In this thesis, the density function is used extensively to model uncertainty in projective quantities such as random directions and random rotations. The following sections discuss some theory behind Bingham's distribution and addresses a few issues that arise in practical applications.

B.1 Formulation

Exponential distributions possess many attractive properties that make them ideal for describing measurement noise and performing inference tasks. In particular, the Gaussian distribution $\mathcal{N}_n(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Lambda})$ is used almost exlusively for modeling uncertainty when nothing more specific is known about a given noise process. However, the sample space of $\mathcal{N}_n(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Lambda})$ is the whole of $\mathbb{R}^n$, which limits its applicability in problems where the sample space is bounded or nonlinear.

Conditioning $\mathcal{N}_n(\boldsymbol{x}; \mathbf{0}, \boldsymbol{\Lambda})$ on the event that $\|\boldsymbol{x}\| = 1$, which restricts vectors $\boldsymbol{x}$ to lie on the surface of the unit hypersphere $\mathbb{S}^{n-1}$, results in Bingham's distribution $\mathcal{B}_n(\boldsymbol{x}; \boldsymbol{M})$. Recall from

§3.1.3 that conditional densities must be normalized, so that

$$\begin{aligned}\mathcal{B}_n(\boldsymbol{x};\boldsymbol{M}) &= \frac{\mathcal{N}_n(\boldsymbol{x};\boldsymbol{0},\boldsymbol{\Lambda})}{P(\|\boldsymbol{x}\|=1)} \\ &= \frac{\mathcal{N}_n(\boldsymbol{x};\boldsymbol{0},\boldsymbol{\Lambda})}{\int_{\mathcal{S}^{n-1}}\mathcal{N}_n(\boldsymbol{x};\boldsymbol{0},\boldsymbol{\Lambda})\mathrm{d}\boldsymbol{x}} \\ &= \frac{1}{c}\exp\left(-\frac{1}{2}\boldsymbol{x}^\top\boldsymbol{\Lambda}^{-1}\boldsymbol{x}\right) \\ &= \frac{1}{c}\exp(\boldsymbol{x}^\top\boldsymbol{M}\boldsymbol{x}) \end{aligned} \tag{B-1}$$

where c is a normalizing constant and $\boldsymbol{M} = -\frac{1}{2}\boldsymbol{\Lambda}^{-1}$. The parameter matrix $\boldsymbol{M}$ can be diagonalized as $\boldsymbol{M} = \boldsymbol{U}\boldsymbol{\kappa}\boldsymbol{U}^\top$, where $\boldsymbol{U}$ is a rotation matrix describing the distribution's orientation, and $\boldsymbol{\kappa}$ is a diagonal matrix of "shape" parameters.

Essentially, (B-1) represents the "intersection" of a zero-mean Gaussian density with the unit hypersphere in $\mathbb{R}^n$. Because of the conventions concerning κ_i in §3.3.2, the parameter matrix $\boldsymbol{M}$ has at least one eigenvalue equal to zero; thus the corresponding Gaussian density has at least one mode with infinite variance. $\boldsymbol{M}$ can be viewed as an information matrix, because it is related to the inverse of a covariance; aggregation of parameter matrices for data fusion as in (3-34) is thus analogous to information pooling [Riv84].

Bingham's distribution can describe a wide variety of shapes with elliptic isoprobability contours on the hypersphere. On $\mathbb{S}^2$, shapes include uniform, bipolar, and equatorial contours with various symmetries and asymmetries (see Figure 3-2) and with arbitrary orientations. The distribution is also closed under rotations, meaning that for a given rotation matrix $\boldsymbol{R}$, the transformation $\boldsymbol{y} = \boldsymbol{R}\boldsymbol{x}$ results in a new, rotated Bingham distribution $\mathcal{B}_n(\boldsymbol{y};\boldsymbol{R}\boldsymbol{M}\boldsymbol{R}^\top)$. To see this, note that $\boldsymbol{R}$ is invertible and unitary (i.e. its norm is one), and also that $\boldsymbol{x} = \boldsymbol{R}^\top\boldsymbol{y}$; substitution into (B-1) then gives

$$\begin{aligned}&\frac{1}{c}\exp(\boldsymbol{x}^\top\boldsymbol{M}\boldsymbol{x}) \\ =\ &\frac{1}{c}\exp(\boldsymbol{y}^\top\boldsymbol{R}\boldsymbol{M}\boldsymbol{R}^\top\boldsymbol{y}) \\ =\ &\mathcal{B}_n(\boldsymbol{y};\boldsymbol{R}\boldsymbol{M}\boldsymbol{R}^\top)\end{aligned} \tag{B-2}$$

B.2 Parameter Computation

The coefficient c in (B-1) is a function of the concentration parameters $\boldsymbol{\kappa}$. In particular, for spherical quantities ($n = 3$),

$$c(\boldsymbol{\kappa}) = {}_1F_1(\frac{1}{2};\frac{3}{2};\boldsymbol{\kappa}) \tag{B-3}$$

where ${}_1F_1(\cdot;\cdot;\cdot)$ represents the confluent hypergeometric function of matrix argument [Her55]. This function can be expanded in an infinite series:

$$c(\boldsymbol{\kappa}) = \sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\sum_{k=0}^{\infty}\frac{\Gamma(i+\frac{1}{2})\Gamma(j+\frac{1}{2})\Gamma(k+\frac{1}{2})}{\Gamma(i+j+k+\frac{3}{2})}\frac{\kappa_1^i\kappa_2^j\kappa_3^k}{i!j!k!} \tag{B-4}$$

However, this series converges slowly for large values of the concentration parameters, and is thus not of much practical use. Other asymptotic approximations have therefore been derived, e.g. by Kent who proposes three different expansions for three exhaustive conditions on the concentration parameters and provides efficient implementations of these expansions [Ken87, Amo74].

As mentioned in §3.3.2, various authors have shown that maximum likelihood estimates of $\boldsymbol{U}$ and $\boldsymbol{\kappa}$ may be obtained from a set k of samples $\boldsymbol{x}_i$ based on the sample second moment matrix $\boldsymbol{S}$ alone, where

$$\boldsymbol{S} = \frac{1}{k}\sum_{i=0}^{k} \boldsymbol{x}_i \boldsymbol{x}_i^\top. \tag{B-5}$$

The ML estimate of $\boldsymbol{U}$ is the matrix of eigenvectors of $\boldsymbol{S}$, and the ML estimate of $\boldsymbol{\kappa}$ is obtained by solving equations of the form

$$\frac{\partial \log \kappa_i}{\partial c(\boldsymbol{\kappa})} = \lambda_i \tag{B-6}$$

where λ_i is the eigenvalue of $\boldsymbol{S}$ corresponding to the eigenvector $\boldsymbol{u}_i$. Kent demonstrates efficient approximations for computing the κ_i from the λ_i and vice versa on $\mathbb{S}^2$ [Ken87].

B.3 Confidence Limits

In evaluating data fusion and other inference techniques, it is necessary to determine certainty or *confidence* bounds on the distribution estimates—for example, to specify a region of the parameter space inside which points have 95% likelihood of belonging to the distribution. This allows for data thresholding based on the likelihood that a given point was drawn from the distribution, and also allows bounds to be placed on the uncertainty. Such regions are usually specified in terms of a percentage $1-\alpha$ of the cumulative distribution function (CDF) of the χ^2 probability density, which describes deviation probabilities for Gaussian distributions [PTVF92]. This CDF is denoted $\chi^2(\alpha|d)$, where $1-\alpha$ (for $0 \leq \alpha \leq 1$) represents the confidence and d is the number of degrees of freedom.

Computation of approximate confidence regions on $\mathbb{S}^2$ for various Bingham parameter estimates have been proposed [Bin74, CW90]. Since quantities on this manifold are essentially two-dimensional, the χ^2 function is evaluated with two degrees of freedom. For example, in estimating the modal axis $\boldsymbol{u}_1$ (where $\boldsymbol{u}_i$ is a single column of the orientation matrix $\boldsymbol{U}$) of an equatorial distribution from a set of k data points, the vectors $\boldsymbol{x}$ satisfying

$$\boldsymbol{x}^\top (\boldsymbol{u}_2 \quad \boldsymbol{u}_3) \begin{pmatrix} (\kappa_1-\kappa_2)(\lambda_1-\lambda 2) & 0 \\ 0 & \kappa_1(\lambda_1-\lambda_3) \end{pmatrix} (\boldsymbol{u}_2 \quad \boldsymbol{u}_3)^\top \boldsymbol{x} \leq \frac{\chi^2(\alpha|2)}{2k} \tag{B-7}$$

have approximately $1-\alpha$ confidence. This bound corresponds to an elliptic region on the surface of $\mathbb{S}^2$.

BIBLIOGRAPHY

[AF84] Yasuo Amemiya and Wayne A. Fuller. Estimation for the multivariate errors-in-variables model with estimated error covariance matrix. *Annals of Statistics*, 12(2):497–509, June 1984.

[Amo74] D. E. Amos. Computation of modified Bessel functions and their ratios. *Mathematics of Computation*, 28(125):239–251, January 1974.

[Ant96] Matthew E. Antone. Synthesis of navigable 3-D environments from human-augmented image data. Master's thesis, Massachusetts Institute of Technology, 1996.

[AP95] A. Azarbayezani and A. Pentland. Recursive estimation of motion, structure, and focal length. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(6):562–575, June 1995.

[AP96] A. Azarbayezani and A. Pentland. Real-time self-calibrating stereo person tracking using 3-D shape estimation from blob features. In *Proceedings of ICPR*, pages 627–632, 1996.

[ARS00] A. Adam, E. Rivlin, and I. Shimshoni. ROR: Rejection of outliers by rotations in stereo matching. In *Proceedings of CVPR*, volume 1, pages 2–9, June 2000.

[AT00a] Matthew E. Antone and Seth Teller. Automatic recovery of camera positions in urban scenes. Technical Report MIT-LCS-814, Massachusetts Institute of Technology Laboratory for Computer Science, December 2000.

[AT00b] Matthew E. Antone and Seth Teller. Automatic recovery of relative camera rotations for urban scenes. In *Proceedings of CVPR*, volume 2, pages 282–289, June 2000.

[Ati92] M. Atiquzzaman. Multiresolution Hough transform—an efficient method of detecting patterns in images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(11):1090–1095, November 1992.

[BA87] J. R. Bergen and E. H. Adelson. Hierarchical, computationally efficient motion estimation algorithm. *Journal of the Optical Society of America A*, 4(35), 1987.

[Bar83] Stephen T. Barnard. Methods for interpreting perspective images. *Artificial Intelligence*, 21:435–462, 1983.

[BB95] Shawn Becker and V. Michael Bove. Semiautomatic 3-D model extraction from uncalibrated 2-D camera views. In *Proceedings of SPIE Image Synthesis*, volume 2410, pages 447–461, February 1995.

[Ber79] Rudolph Beran. Exponential models for directional data. *Annals of Statistics*, 7(6):1162–1178, November 1979.

[BHR86] J. B. Burns, A. R. Hanson, and E. M. Riseman. Extracting straight lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4(8):425–455, 1986.

[Bin74] Christopher Bingham. An antipodally symmetric distribution on the sphere. *Annals of Statistics*, 2(6):1201–1225, November 1974.

[Bis95] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford, 1995.

[BKY97] P. N. Belhumeur, D. J. Kriegman, and A. L. Yuille. The bas-relief ambiguity. In *Proceedings of CVPR*, pages 1060–1066, 1997.

[BM92] P. J. Besl and H. D. McKay. A method for registration of 3D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, February 1992.

[BN98] S. Baker and S. K. Nayar. A theory of catadioptric image formation. In *Proceedings of ICCV*, pages 35–42, January 1998.

[BT00] Michael C. Bosse and Seth Teller. A high-resolution geo-referenced pose camera. Manuscript, 2000.

[Can86] John F. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, November 1986.

[CC91] Subhasis Chaudhuri and Shankar Chatterjee. Robust estimation of 3-D motion parameters in presence of correspondence mismatches. In *Proceedings of Asilomar Conference on Signals, Systems and Computers*, pages 1195–1199, November 1991.

[CDS00] X. Chen, J. Davis, and P. Slusallek. Wide area camera calibration using virtual calibration objects. In *Proceedings of CVPR*, volume 2, pages 520–527, June 2000.

[Cha89] Ted Chang. Spherical regression with errors in variables. *Annals of Statistics*, 17(1):293–306, March 1989.

[CMT98] Satyan Coorg, Neel Master, and Seth Teller. Acquisition of a large pose-mosaic dataset. In *Proceedings of CVPR*, pages 872–878, June 1998.

[Col93] Robert T. Collins. *Model Acquisition using Stochastic Projective Geometry*. PhD thesis, University of Massachusetts, September 1993.

[Coo98] Satyan Coorg. *Pose Imagery and Automated 3-D Modeling of Urban Environments*. PhD thesis, Massachusetts Institute of Technology, September 1998.

[CR00a] Haili Chui and Anand Rangarajan. A feature registration framework using mixture models. In *Proceedings of IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, pages 190–197, 2000.

[CR00b] Haili Chui and Anand Rangarajan. A new algorithm for non-rigid point matching. In *Proceedings of CVPR*, volume 2, pages 44–51, June 2000.

[CT90] Bruno Caprile and Vincent Torre. Using vanishing points for camera calibration. *International Journal of Computer Vision*, 4(2):127–140, March 1990.

[CT97] George T. Chou and Seth Teller. Multi-image correspondence using geometric and structural constraints. In *Proceedings of the Image Understanding Workshop*, pages 869–874, May 1997.

[CT99] Satyan Coorg and Seth Teller. Extracting textured vertical façades from controlled close-range imagery. In *Proceedings of CVPR*, pages 625–632, June 1999.

[CV98] D. Chetverikov and J. Verestoy. Tracking feature points: A new algorithm. In *Proceedings of ICPR*, volume 2, pages 1436–1438, 1998.

[CW90] Robert T. Collins and R. Weiss. Vanishing point calculation as statistical inference on the unit sphere. In *Proceedings of ICCV*, pages 400–403, December 1990.

[CZ98] D. Capel and A. Zisserman. Automated mosaicing with super-resolution zoom. In *Proceedings of CVPR*, pages 885–891, June 1998.

[CZZF97] G. Csurka, C. Zeller, Z. Zhang, and O. Faugeras. Characterizing the uncertainty of the fundamental matrix. *Computer Vision and Image Understanding*, 68(1):18–36, October 1997.

[dBvKOS91] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, Berlin, 1991.

[DC87] M. Davidian and R. J. Carroll. Variance function estimation. *Journal of the American Statistical Association*, 82(400):1079–1091, December 1987.

[Del00] Frank Dellaert. Addressing the correspondence problem: A Markov chain Monte Carlo approach. Technical Report CMU-RI-TR-00-11, Carnegie Mellon University School of Computer Science, January 2000.

[DLR77] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[Dra67] A. W. Drake. *Fundamentals of Applied Probability Theory*. McGraw-Hill Series in Probability and Statistics. McGraw-Hill, New York, NY, 1967.

[DSTT00] Frank Dellaert, Steven M. Seitz, Charles E. Thorpe, and Sebastian Thrun. Structure from motion without correspondence. In *Proceedings of CVPR*, volume 2, pages 557–564, June 2000.

[DTM96] Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *Proceedings of SIGGRAPH*, pages 11–20, 1996.

[Ech89] Tomio Echigo. A camera calibration technique using sets of parallel lines. In *Proceedings of International Workshop on Industrial Applications of Machine Intelligence and Vision*, pages 151–156, April 1989.

[FA98] Cornelia Fermüller and Yiannis Aloimonos. Ambiguity in structure from motion: Sphere versus plane. *International Journal of Computer Vision*, 28(2):137–154, 1998.

[FAB+00] C. Fermüller, Y. Aloimonos, P. Baker, R. Pless, J. Neumann, and B. Stuart. Multi-camera networks: Eyes from eyes. In *Proceedings of IEEE Workshop on Omnidirectional Vision*, pages 11–18, June 2000.

[Fau93] Olivier Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, Cambridge, MA, 1993.

[FB81] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[FL94] P. Fua and Y. G. Leclerc. Registration without correspondence. In *Proceedings of CVPR*, pages 121–128, June 1994.

[FvDFH90] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, Reading, MA, second edition, 1990.

[FZ98] A. W. Fitzgibbon and A. Zisserman. Automatic camera recovery for closed or open image sequences. In *Proceedings of ECCV*, pages 311–326, June 1998.

[Ger99] Neil Gershenfeld. *The Nature of Mathematical Modeling*. Cambridge University Press, New York, NY, 1999.

[GGSC96] Steven Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael Cohen. The lumigraph. In *Proceedings of SIGGRAPH*, pages 43–54, August 1996.

[GH90] W. E. L. Grimson and D. P. Huttenlocher. On the sensitivity of the Hough transform for object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(3):225–274, March 1990.

[GL80] Gene H. Golub and Charles F. Van Loan. An analysis of the total least squares problem. *SIAM Journal on Numerical Analysis*, 17(6):883–893, December 1980.

[GLT99] S. Gobert, T. Laurencot, and I. Tannous. Design of an integrated system for 3D site reconstruction, control and exploitation from multi-sensor imagery. In *ISPRS Workshop on 3D Geospatial Data Production*, pages 108–120, April 1999.

[Har97] Richard I. Hartley. In defence of the 8-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593, June 1997.

[Her55] Carl S. Herz. Bessel functions of matrix argument. *The Annals of Mathematics, Second Series*, 61(3):474–523, May 1955.

[Her95] L. K. Herman. The history, definition and pecularities of the Earth centered inertial (ECI) coordinate frame and the scales that measure time. In *Proceedings of Aerospace Applications*, pages 233–263, February 1995.

[Hor86] Berthold K. P. Horn. *Robot Vision*. MIT Press, Cambridge, MA, 1986.

[Hor87] Berthold K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642, April 1987.

[Hor91] Berthold K. P. Horn. Relative orientation revisited. *Journal of the Optical Society of America A*, 8(10):1630–1638, October 1991.

[Hor00] Berthold K. P. Horn. Projective geometry considered harmful. Manuscript, 2000.

[Hou62] P. V. C. Hough. A method and means for recognizing complex patterns. U. S. Patent No. 3,069,654, 1962.

[HS81] Berthold. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 16(1–3):185–203, August 1981.

[HS88] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Proceedings of the Alvey Vision Conference*, pages 147–152, 1988.

[HZ00] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, 2000.

[IMG00] Aaron Isaksen, Leonard McMillan, and Steven J. Gortler. Dynamically reparameterized light fields. In *Proceedings of SIGGRAPH*, pages 297–306, July 2000.

[JM79] P. E. Jupp and K. V. Mardia. Maximum likelihood estimators for the matrix von Mises-Fisher and Bingham distributions. *Annals of Statistics*, 7(3):599–606, May 1979.

[Kan92] Kenichi Kanatani. Statistical analysis of focal-length calibration using vanishing points. *IEEE Transactions on Robotics and Automation*, 8(6):767–775, December 1992.

[Kan94] Kenichi Kanatani. Analysis of 3-D rotation fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):543–549, May 1994.

[Ken82] John T. Kent. The Fisher-Bingham distribution on the sphere. *Journal of the Royal Statistical Society, Series B*, 44(1):71–80, 1982.

[Ken87] John T. Kent. Asymptotic expansions for the Bingham distribution. *Applied Statistics*, 36(2):139–144, 1987.

[KGV83] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[KH98] Fredrik Kahl and Anders Heyden. Robust self-calibration and Euclidean reconstruction via affine approximation. In *Proceedings of ICPR*, 1998.

[KM63] M. G. Kendall and P. A. P. Moran. *Geometrical Probability*, volume 10 of *Griffin's Statistical Monographs & Courses*. Charles Griffin & Co., Ltd., London, 1963.

[KMT00] Adam Kropp, Neel Master, and Seth Teller. Acquiring and rendering high-resolution spherical mosaics. In *Proceedings of IEEE Workshop on Omnidirectional Vision*, pages 47–53, June 2000.

[KN95] Yoshihiko Kimuro and Tadashi Nagata. Image processing on an omni-directional view using a spherical hexagonal pyramid: Vanishing points extraction and hexagonal chain coding. In *Proceedings of International Conference on Intelligent Robots and Systems*, volume 3, pages 356–361, 1995.

[KTT99] A. A. Kassim, T. Tan, and K. H. Tan. A comparative study of efficient generalised Hough transform techniques. *Image and Vision Computing*, 17:737–748, 1999.

[LCZ99] D. Liebowitz, A. Criminisi, and A. Zisserman. Creating architectural models from images. In *Proceedings of Eurographics*, volume 18, pages 39–50, September 1999.

[LF97] Q. T. Luong and O. Faugeras. Camera calibration, scene motion, and structure recovery from point correspondences and fundamental matrices. *International Journal of Computer Vision*, 22(3):261–289, 1997.

[LH81] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, September 1981.

[Lim90] Jae S. Lim. *Two-Dimensional Signal and Image Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1990.

[LLF98] Y. H. Leclerc, Q. T. Luong, and P. Fua. Self-consistency: A novel approach to characterizing the accuracy and reliability of point correspondence algorithms. In *Proceedings of the Image Understanding Workshop*, pages 793–807, November 1998.

[LM96] John C. H. Leung and Gerard F. McLean. Vanishing point matching. In *Proceedings of ICIP*, volume 2, pages 305–308, 1996.

[LMD95] Mi-Suen Lee, Gerard Medioni, and Rachid Deriche. Structure and motion from a sparse set of views. In *Proceedings of the International Symposium on Computer Vision*, pages 73–78, November 1995.

[LMLK94] Evelyne Lutton, Henri Maître, and Jaime Lopez-Krahe. Contribution to the determination of vanishing points using Hough transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(4):430–438, April 1994.

[LPC+00] Marc Levoy, Kari Pulli, Brian Curless, Szymon Rusinkiewicz, Dave Koller, Lucas Pereira, Matt Ginzton, Sean Anderson, James Davis, Jeremy Ginsberg, Jonathan Shade, and Duane Fulk. The digital Michelangelo project: 3d scanning of large statues. In *Proceedings of SIGGRAPH*, pages 131–144, 2000.

[LTM99] R. Lenz, Linh Viet Tran, and P. Meer. Moment based normalization of color images. In *IEEE Workshop on Multimedia Signal Processing*, pages 103–108, September 1999.

[LZ98] D. Liebowitz and A. Zisserman. Metric rectification for perspective images of planes. In *Proceedings of CVPR*, pages 482–488, June 1998.

[MA84] M. J. Magee and J. K. Aggarwal. Determining vanishing points from perspective images. *Computer Vision, Graphics and Image Processing*, 26(2):256–267, May 1984.

[Mar72] K. V. Mardia. *Statistics of Directional Data*. Academic Press, London, 1972.

[MB95] Leonard McMillan and Gary Bishop. Plenoptic modeling: An image-based rendering system. In *Proceedings of SIGGRAPH*, pages 39–46, 1995.

[MD89] John Moody and Christian Darken. Fast learning in networks of locally-tuned processing units. *Neural Computation*, 1:289–303, 1989.

[Mel99] J. P. Mellor. Reconstructing built geometry from large sets of calibrated images. Technical Report MIT-AITR-1674, Massachusetts Institute of Technology Artificial Intelligence Laboratory, October 1999.

[MK95] G. F. McLean and D. Kotturi. Vanishing point detection by line clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(11):1090–1095, November 1995.

[MM00] Bogdan Matei and Peter Meer. A general method for errors-in-variables problems in computer vision. In *Proceedings of CVPR*, volume 2, pages 18–25, June 2000.

[MRR+53] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[MSKS00] Yi Ma, Stefano Soatto, Jana Kosecka, and Shankar S. Sastry. Euclidean reconstruction and reprojection up to subgroups. *International Journal of Computer Vision*, 38(3):217–227, 2000.

[MZ92] Joseph L. Mundy and Andrew Zisserman, editors. *Geometric Invariance in Computer Vision*. MIT Press, Cambridge, MA, 1992.

[NFH00] O. Nestares, D. J. Fleet, and D. J. Heeger. Likelihood functions and confidence bounds for total-least-squares problems. In *Proceedings of CVPR*, volume 1, pages 523–530, June 2000.

[Nie94] Yves Nievergelt. Total least squares: State-of-the-art regression in numerical analysis. *SIAM Review*, 36(2):258–264, June 1994.

[NP88] Wayne Niblack and Dragutin Petkovic. On improving the accuracy of the Hough transform: Theory, simulation, and experiments. In *Proceedings of CVPR*, pages 574–579, 1988.

[NRK98] P. J. Narayanan, Peter W. Rander, and Takeo Kanade. Constructing virtual worlds using dense stereo. In *Proceedings of ICCV*, pages 3–10, January 1998.

[NS94] Keith E. Nicewarner and A. C. Sanderson. A general representation for orientation uncertainty using random unit quaternions. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 2, pages 1161–1168, May 1994.

[PIK94] J. Princen, J. Illingworth, and J. Kittler. Hypothesis testing: A framework for analyzing and optimizing Hough transform performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(4):329–341, April 1994.

[PK94] C. J. Poelman and Takeo Kanade. A paraperspective factorization method for shape and recovery. In *Proceedings of ECCV*, pages 97–108, May 1994.

[Pre84] Michael J. Prentice. A distribution-free method of interval estimation for unsigned directional data. *Biometrika*, 71(1):147–154, April 1984.

[Pre86] Michael J. Prentice. Orientation statistics without parametric assumptions. *Journal of the Royal Statistical Society, Series B*, 48(2):214–222, 1986.

[Pre89] Michael J. Prentice. Spherical regression on matched pairs of orientation statistics. *Journal of the Royal Statistical Society, Series B*, 51(2):241–248, 1989.

[PTVF92] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C*. Cambridge University Press, New York, NY, second edition, 1992.

[PZ98] Philip Pritchett and Andrew Zisserman. Matching and reconstruction from widely separated views. In *Proceedings of Workshop on 3-D Structure from Multiple Images of Large-Scale Environments*, pages 78–92, June 1998.

[QM88] Long Quan and Roger Mohr. Matching perspective images using geometric constraints and perceptual grouping. In *Proceedings of ICCV*, pages 679–684, 1988.

[RCD99] Anand Rangarajan, Haili Chui, and James S. Duncan. Rigid point feature registration using mutual information. *Medical Image Analysis*, 4:1–17, 1999.

[Riv84] Louis-Paul Rivest. On the information matrix for symmetric distributions on the unit hypersphere. *Annals of Statistics*, 12(3):1085–1089, September 1984.

[SA00] I. Stamos and P. E. Allen. 3-D model construction using range and image data. In *Proceedings of CVPR*, volume 1, pages 531–536, June 2000.

[SABH93] Rolf Schuster, Nirwan Ansari, and Ali Bani-Hashemi. Steering a robot with vanishing points. *IEEE Transactions on Robotics and Automation*, 9(4):491–498, August 1993.

[Sar00] R. Sara. Accurate natural surface reconstruction from polynocular stereo. In *Proceedings of NATO Advanced Research Workshop on Confluence of Computer Vision and Computer Graphics*, pages 69–86, 2000.

[SB98] Stefano Soatto and Roger Brockett. Optimal structure from motion: Local ambiguities and global estimates. In *Proceedings of CVPR*, pages 282–288, 1998.

[SHS98] Harry S. Shum, Mei Han, and Richard Szeliski. Interactive construction of 3D models from panoramic image mosaics. In *Proceedings of CVPR*, pages 427–433, 1998.

[Shu99] Jefferey A. Shufelt. Performance evaluation and analysis of vanishing point detection techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(3):282–288, March 1999.

[Sin64] Richard Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *Annals of Mathematical Statistics*, 35(2):876–879, June 1964.

[Sin67] Richard Sinkhorn. Diagonal equivalence to matrices with prescribed row and column sums. *American Mathematical Monthly*, 74(4):402–405, April 1967.

[SK94] Richard Szeliski and S. B. Kang. Recovering 3D shape and motion from image streams using nonlinear least squares. *Journal of Visual Communication and Image Representation*, 5(1):10–28, March 1994.

[SKS95] R. Szeliski, Sing Bing Kang, and Heung-Yeung Shum. A parallel feature tracker for extended image sequences. In *Proceedings of International Symposium on Computer Vision*, pages 241–246, 1995.

[Soa97] Stefano Soatto. 3-D structure from visual motion: Modeling, representation and observability. *Automatica*, 33(7):1287–1312, 1997.

[Sol78] H. Solomon. *Geometric Probability*, volume 28 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. SIAM, Philadelphia, PA, 1978.

[SS90] G. W. Stewart and Ji-Guang Sun. *Matrix Perturbation Theory*. Academic Press, London, 1990.

[SS97] Richard Szeliski and H. Y. Shum. Creating full view panoramic image mosaics and texture-mapped models. In *Proceedings of SIGGRAPH*, pages 251–258, August 1997.

[Ste85] Leonard A. Stefanski. The effects of measurement error on parameter estimation. *Biometrika*, 72(3):583–592, December 1985.

[Ste90] G. W. Stewart. Stochastic perturbation theory. *SIAM Review*, 32(4):579–610, December 1990.

[Ste98] Gideon P. Stein. Tracking from multiple view points: Self-calibration of space and time. In *Proceedings of CVPR*, volume 1, pages 521–527, November 1998.

[STI90] Li Shigang, Saburo Tsuji, and Masakazu Imai. Determining of camera rotation from vanishing points of lines on horizontal planes. In *Proceedings of ICCV*, pages 499–502, 1990.

[Sto91] Jorge Stolfi. *Oriented Projective Geometry: A Framework for Geometric Computations*. Academic Press, San Diego, CA, 1991.

[Str88] Gilbert Strang. *Linear Algebra and its Applications*. Academic Press, New York, NY, 1988.

[TD99] M. F. Thorpe and P. M. Duxbury, editors. *Rigidity Theory and Applications*. Plenum Press, New York, NY, 1999.

[Tel98] Seth Teller. Toward urban model acquisition from geo-located images. In *Proceedings of Pacific Graphics*, pages 45–51, October 1998.

[TK92] Camillo J. Taylor and David J. Kriegman. Structure and motion from line segments in multiple images. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1615–1620, May 1992.

[TKA91] Camillo J. Taylor, David J. Kriegman, and P. Anandan. Structure and motion in two dimensions from multiple images: A least squares approach. In *Proceedings of the IEEE Workshop on Visual Motion*, pages 242–248, October 1991.

[TML99] Chi-Keung Tang, Gérard Medioni, and Mi-Suen Lee. Epipolar geometry estimation by tensor voting in 8D. In *Proceedings of ICCV*, volume 1, pages 502–509, 1999.

[TPG97] Tinne Tuytelaars, Marc Proesmans, and Luc Van Gool. The cascaded Hough transform. In *Proceedings of ICIP*, volume 2, pages 736–739, 1997.

[Tsa87] Roger Y. Tsai. A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, 1987.

[Tyl87] David E. Tyler. Statistical analysis for the angular central Gaussian distribution on the sphere. *Biometrika*, 74(3):579–589, September 1987.

[TZ00] P. H. S. Torr and A. Zisserman. Mlesac: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78:138–156, 2000.

[Wat83] G. S. Watson. *Statistics on Spheres*. John Wiley and Sons, New York, NY, 1983.

[Wel97] William Wells. Statistical approaches to feature-based object recognition. *International Journal of Computer Vision*, 21(1/2):63–98, January 1997.

[Wil94] Richard P. Wildes. Singularities of the visual motion field: 3D rotation or 3D translation. In *Proceedings of CVIP*, pages 633–636, 1994.

[WT91] L. L. Wang and W. H. Tsai. Camera calibration by vanishing lines for 3-D computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:370–376, 1991.

[YC92] G. S. Young and R. Chellappa. Statistical analysis of inherent ambiguities in recovering 3-D motion from a noisy flow field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(10):995–1013, October 1992.

[Zha98] Zhengyou Zhang. Determining the epipolar geometry and its uncertainty: A review. *International Journal of Computer Vision*, 27(2):161–195, 1998.

[ZMI00] Lihi Zelnik-Manor and Michal Irani. Multi-frame estimation of planar motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1105–1116, October 2000.