Die approbierte Originalversion dieser Diplom-/Masterarbeit ist an der Hauptbibliothek der Technischen Universität Wien aufgestellt (http://www.ub.tuwien.ac.at).

The approved original version of this diploma or master thesis is available at the main library of the Vienna University of Technology (http://www.ub.tuwien.ac.at/englweb/).

FAKULTÄT
FÜR !NFORMATIK

Faculty of Informatics

Extracting Vanishing Points across Multiple Views

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Computergraphik/Digitale Bildverarbeitung

eingereicht von

Michael Hornáček, BA
Matrikelnummer 0848108

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung
Betreuer: Univ.-Prof. Dipl.-Ing. Dr. Werner Purgathofer
Mitwirkung: Dipl.-Ing. Dr. Robert F. Tobler

Wien, 28.9.2010

(Unterschrift Verfasser)

(Unterschrift Betreuer)

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.ac.at

FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Extracting Vanishing Points across Multiple Views

MASTER'S THESIS

in partial fulfillment of the requirements for the degree of

Master of Science

within the framework of the study program

Computer Graphics and Digital Image Processing

submitted by

Michael Hornáček, BA
Matriculation Number 0848108

at the
Faculty of Informatics of the Vienna University of Technology

Supervision
Supervisor: Univ.-Prof. Dipl.-Ing. Dr. Werner Purgathofer
Collaborator: Dipl.-Ing. Dr. Robert F. Tobler

Vienna, 28.9.2010

(Signature of the Author)

(Signature of the Supervisor)

Erklärung zur Verfassung der Arbeit

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Michael Hornáček
Breite Gasse 3/4
A-1070 Wien

Wien, 28.9.2010

Dedication

To the ineffable Ellen Moore, Gerry LaValley and Neil Cameron, ere whose acquaintance I thought and spake in but hollow noises…

Acknowledgements

I would like to express my gratitude to my thesis supervisor, Professor Werner Purgathofer of the Vienna University of Technology, and to Robert F. Tobler of the VRVis[1] Research Center in Vienna, Austria, who enthusiastically offered suggestions on technical matters and took the time to read through several drafts.

I would like to thank everybody who lent me a hand in one way or another at VRVis, where I had the privilege of producing this thesis; specifically, I owe a debt of gratitude to Anton Fuhrmann, Christian Luksch, Stefan Maierhofer, Janne Mihola, Przemysław Musialski, Andreas Reichinger, Irene Reisner-Kollmann and Michael Schwärzler. Andi helped me with translating my abstract and summary into German. Christian and Janne contributed ideas that led to the tripod fitting approach presented in this thesis. Irene's help with respect to relative camera pose estimation was of the sort that I almost certainly would not have finished my thesis on time without it.

Roman Pflugfelder of the Austrian Institute of Technology (AIT) fortuitously attended my opening talk and provided invaluable pointers on the direction in which to take my work. Without his advice my thesis might well not have turned out the way it did.

I would also like to thank my parents, Jana and Peter, and my endlessly loving fiancée Clio, for their ongoing support and encouragement.

Finally, I would like to gratefully acknowledge the generous financial support of the VRVis Research Center. In this matter I owe thanks primarily to Stefan Maierhofer, who granted me gainful employment at VRVis without having had a clear idea of how or what I might contribute to his group.

Lest anyone should believe otherwise, if this thesis is at all worth reading it is because I stand on the shoulders of giants; any errors or omissions, however, are to be attributed strictly to my person alone.

---

[1] `http://www.vrvis.at/`

Kurzfassung

Die Erkenntnis, dass wir Linien, von denen wir wissen, dass sie tatsächlich im Raum parallel sind, als Linien wahrnehmen, die scheinbar zu einem gemeinsamen Fluchtpunkt konvergieren, hat zu Techniken geführt, mit denen Künstler einen glaubwürdigen Eindruck von Perspektive vermitteln können. Dies führte später auch zu Ansätzen, mit denen die zugrundeliegende Geometrie von Bildern – oder in der Tat auch von Gemälden mit korrekter Perspektive – extrahiert werden kann.

In dieser Arbeit beschäftigen wir uns mit der Extraktion von Fluchtpunkten mit dem Ziel, die Rekonstruktion urbaner Szenen zu vereinfachen. Im Gegensatz zu den meisten Methoden zur Extraktion von Fluchtpunkten, extrahiert die unsere eine Konstellation von Fluchtpunkten über mehrere Ansichten hinweg, anstatt nur in einem einzigen Bild. Durch das Verwenden eines starken Orthogonalitätskriteriums in jeder Ansicht, einer optimalen Berechnung von Segmentschnittpunkten und einem neuartigen Dreibein-Ausrichtungsverfahren, erlaubt unser Ansatz die Extraktion von Ergebnissen, die eine nahe Approximation der dominanten drei paarweise orthogonalen Orientierungen typischer urbaner Szenen darstellen. Dementsprechend kann unser Ansatz als wesentliche Verfeinerung der Methode von Sinha et al. bezeichnet werden.

Abstract

The realization that we see lines we know to be parallel in space as lines that appear to converge in a corresponding vanishing point has led to techniques employed by artists to render a credible impression of perspective. More recently, it has also led to techniques for recovering information embedded in images—or, indeed, in paintings that feature correct perspective—concerning the geometry of their underlying scene.

In this thesis, we explore the extraction of vanishing points in the aim of facilitating the reconstruction of urban scenes. In departure from most vanishing point extraction methods, ours extracts a constellation of vanishing points across multiple views rather than in a single image alone. By making use of a strong orthogonality criterion per view, optimal segment intersection estimation and a novel tripod fitting technique, our approach allows for the extraction of results that correspond closely to the dominant three pairwise-orthogonal orientations of a typical urban scene. Accordingly, ours can fairly be described as a material refinement of the approach proposed in Sinha et al.

Contents

List of Figures xiii

List of Algorithms xv

1 Introduction 1
- 1.1 Motivation and Objective 1
- 1.2 Organization 2
- 1.3 Notational Conventions 2

2 Related Work 5
- 2.1 Extraction Techniques 5
  - 2.1.1 Estimation 8
- 2.2 Application to Scene Reconstruction 8

3 The Geometry of Vanishing Points 11
- 3.1 A Suitable Geometric Framework 11
  - 3.1.1 Homogeneous Coordinates 12
- 3.2 The Projective Plane $\mathbb{P}^2$ 14
  - 3.2.1 Incidence, Collinearity and Concurrence 15
  - 3.2.2 Duality of Points and Lines 16
  - 3.2.3 Projective Transformations of $\mathbb{P}^2$ 16
- 3.3 The Projective Space $\mathbb{P}^3$ 18
- 3.4 Image Formation 18
  - 3.4.1 The Thin Lens Camera 19
  - 3.4.2 The Pinhole Camera 20
  - 3.4.3 The Finite Projective Camera 21
  - 3.4.4 Mappings between Planes 23
  - 3.4.5 Forward Projection 23
  - 3.4.6 Back-Projection 24
- 3.5 Vanishing Points 25
- 3.6 Vanishing Lines 26
- 3.7 Synopsis 27

4 Implementation 29
- 4.1 Processing Pipeline 29
- 4.2 Extracting a Constellation in a Single View 30
- 4.3 Optimizing across k Views of a Scene 34

5 Evaluation **37**
5.1 Remarks on Complexity 37
5.2 Results 38

6 Conclusion **47**
6.1 Recommendations 47

7 Summary **49**

8 Zusammenfassung **51**

A The Single-View Approach of Rother **53**
A.1 Accumulation Step 53
A.2 Search Step 56

B The Multiple-View Approach of Sinha et al. **59**
B.1 Algorithm 59

C Random Sample Consensus **61**
C.1 Framework 61

D Singular Value Decomposition **63**
D.1 Formulation 63
D.2 Minimizing the Quantity $\|\mathtt{Ax}\|^2$ over $\mathbf{x}$ 64
D.3 Orthogonalizing a Square Matrix 64

Bibliography **65**

List of Figures

1.1 Vanishing points in da Vinci's *The Last Supper* 2
1.2 Vanishing Points and Corresponding Scene Orientations 3

2.1 The Gaussian sphere 6
2.2 Scene reconstruction of Vermeer's *The Music Lesson* 9
2.3 The scene reconstruction approach of Werner and Zisserman 10
2.4 The scene reconstruction pipeline of Sinha et al. 10

3.1 One-, two- and three-point perspective 12
3.2 Dürer's string method 13
3.3 The projective plane $\mathbb{P}^2$ 14
3.4 The thin lens camera model 19
3.5 The pinhole camera model 20
3.6 The frontal pinhole camera model 21
3.7 Mappings between planes 24
3.8 How vanishing points arise 25
3.9 Vanishing points and vanishing lines 28

4.1 The processing pipeline of our approach 29
4.2 The optimal line $\hat{\mathbf{l}}_i$ through $\mathbf{v}$ with respect to the segment s_i 30
4.3 Maximum likelihood intersection estimation 32
4.4 Our orthogonality criterion 33

5.1 The `acv` data set with inlier segments w.r.t. the best-fit tripod 39
5.2 The `ares` data set with inlier segments w.r.t. the best-fit tripod 40
5.3 The `techgate` data set with inlier segments w.r.t. the best-fit tripod . 41
5.4 Antipodal unit direction vectors and best-fit tripod per data set. 42
5.5 A best-fit tripod viewed from different poses 42
5.6 Inlier proportions (`acv`) 43
5.7 Inlier proportions (`ares`) 43
5.8 Inlier proportions (`techgate`) 44
5.9 Cumulative inlier error relative to inlier count (`acv`) 44
5.10 Cumulative inlier error relative to inlier count (`ares`) 45
5.11 Cumulative inlier error relative to inlier count (`techgate`) 45

A.1 Rother's distance function $d(\mathbf{v}, s)$ 54
A.2 Rother's distance function $d(l, s)$ 55

List of Algorithms

1 Extracting a Constellation of Vanishing Points in a Single View 34
2 Fitting a Tripod with Pairwise-Orthogonal Axes across k Views . . . 36
3 Computing an Additional Scene Orientation 48
4 Rother's Accumulation Step 55
5 Rother's Search Step 57
6 The Multiple-View Approach of Sinha et al. 60
7 The RANSAC Framework 62

Chapter 1

Introduction

1.1 Motivation and Objective

In casting a glance at a scene, we are not surprised to see that lines we know from experience to be parallel appear to converge in a single point. The realization that this occurs has led to techniques employed by artists to render a credible impression of perspective, as da Vinci famously did in his fifteenth-century mural, *The Last Supper*[1] (cf. Figure 1.1). It has also led to more recent techniques for recovering information embedded in images—or, indeed, in paintings that feature correct perspective—concerning the geometry of their underlying scene. These techniques can provide constraints for scene reconstruction, partial camera calibration and the navigation of robots and autonomous vehicles. In this regard, a sizeable literature has arisen since the late 1970's, offering a litany of algorithms for extracting and employing knowledge of vanishing points.

In this thesis, we explore the extraction of vanishing points in the aim of facilitating the reconstruction of urban scenes. Real-world urban scenes tend to feature a predominance of scene lines corresponding to the pairwise-orthogonal axes of a 3-dimensional Euclidean coordinate frame; accordingly, we shall have in mind a scene that indeed features a predominance of such lines when referring to what we call a *typical* urban scene. It is on account of the geometry of image formation that a set of lines parallel in the scene—that is, that share a single orientation in space—project to lines in the image plane that converge in a corresponding vanishing point. Under known camera geometry, we can map that vanishing point back to a ray through the camera center that likewise shares that same orientation (cf. Figure 1.2). Accordingly, if we are able to compute the vanishing points corresponding to the scene's dominant three pairwise-orthogonal line orientations, we have in our possession normal vectors corresponding closely to each of the scene's dominant three pairwise-orthogonal plane orientations.

In departure from most vanishing point extraction methods, ours extracts a constellation of vanishing points across *multiple views* rather than in a single image alone. By making use of a strong orthogonality criterion per view, optimal segment intersection estimation and a novel tripod fitting technique, our approach allows for the extraction of results that correspond closely to the dominant three pairwise-orthogonal orientations of a typical urban scene. Accordingly, ours can fairly be described as a material refinement of the approach proposed in Sinha et al. [38].

---

[1] See `http://www.haltadefinizione.com/` for an image of da Vinci's *The Last Supper* with a resolution of 16 *billion* pixels.

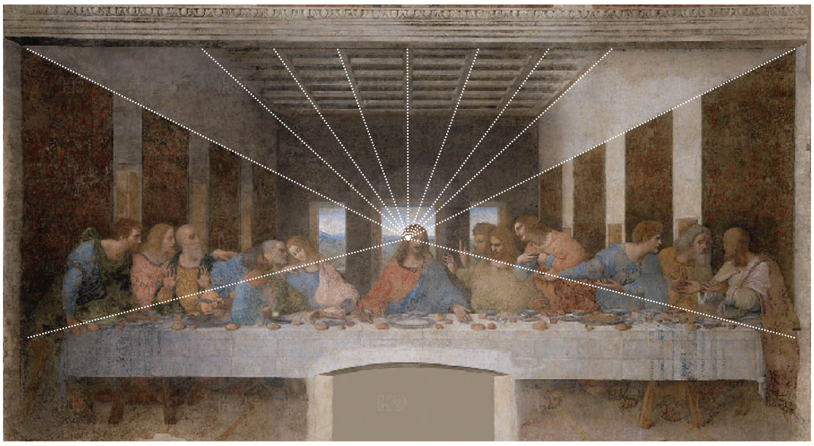


Figure 1.1: Leonardo da Vinci's fifteenth-century mural, *The Last Supper*. Note that the superimposed segments—representing the projection of a set of lines we understand to be parallel in the scene that da Vinci depicts—converge in a vanishing point in the canvas. Image © HAL9000 S.r.l. - Haltadefinizione [14].

1.2 Organization

We begin with an overview of related work in the field in Chapter 2, where we discuss extraction techniques and their application to scene reconstruction. In Chapter 3, we give an introduction to the geometry of vanishing points, which is intended to serve as a self-contained primer to the subject for anybody familiar with basic vector algebra. In Chapter 4, we discuss the multiple-view approach implemented within the framework of this master's thesis. Finally, we provide an evaluation of our algorithm in Chapter 5 and close the thesis in Chapter 6 with a conclusion, in which we include recommendations with respect to the integration of our approach with a larger urban scene reconstruction system. The appendices serve to explain or motivate techniques central to our approach but that do not fit thematically with Chapter 3.

1.3 Notational Conventions

We have tried to follow the notational conventions that Hartley and Zisserman employ in their widely cited canonical text, *Multiple View Geometry in Computer Vision* [15]. We do so because the text is widely recognized as one of the principal authoritative sources on the geometry of image formation across multiple views, and because it was unequivocally the main source used in penning this master's thesis. Accordingly, we represent vectors in boldface, e.g., $\mathbf{b}$. Also, rather than specify points using coordinate notation (a, b, c), we write them as column vectors $(a, b, c)^\top$. Starting in Section 3.4 of Chapter 3, we represent points in world coordinates $\mathbf{X} \sim (\mathrm{X}, \mathrm{Y}, \mathrm{Z})^\top$ with upper-case letters and their projections $\mathbf{x} \sim (x, y, z)^\top$, with letters in lower case. We also attempt, wherever possible, to name and present our vectors and matrices in a manner consistent with the text; e.g., we thus denote the line at infinity as $\mathbf{l}_\infty \sim (0, 0, 1)^\top$. In noteworthy departure from their conventions, however, we use the similarity relation $\sim$ to indicate that two vectors in $\mathbb{P}^n$ are equal to within a non-zero scalar k, $\mathbf{x} \sim \mathbf{x}' \Leftrightarrow \exists k \neq 0 : \mathbf{x} = k\mathbf{x}'$, rather than by 'overloading' the equality relation $=$ as they do.

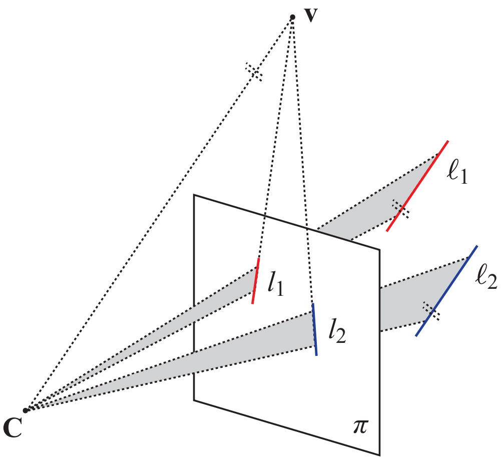


Figure 1.2: The projections $l_1, l_2 \subset \pi$ of two lines ℓ_1, ℓ_2 in space converge in a corresponding vanishing point $\mathbf{v}$ in the image plane π. Note that under known camera geometry, the lines ℓ_1, ℓ_2 in space thus have the same orientation as the ray extending from the camera center $\mathbf{C}$ through $\mathbf{v}$. We call that ray the back-projection of $\mathbf{v}$ with respect to the given camera.

Chapter 2

Related Work

The literature on the extraction of vanishing points dates back to the late 1970's and straddles the fields of photogrammetry, computer vision and robotics. As we mentioned in Chapter 1, knowledge of vanishing points has been put to use in scene reconstruction, partial camera calibration and the navigation of robots and autonomous vehicles. Since our focus is on scene reconstruction, however, we direct our attention to extraction approaches accordingly.[1] We proceed by first examining the progression of relevant vanishing point extraction techniques proposed over the years in the literature. Next, we consider how vanishing points have been employed in facilitating scene reconstruction.

2.1 Extraction Techniques

Extraction techniques tend to involve what amount to an *accumulation* (or *grouping*) *step* followed by an *estimation* (or *search*) *step*, perhaps repeated for some number of iterations. In the accumulation step, line segments—which are typically obtained in a pre-processing step (cf. Guru et al. [13], Rosin and West [33] or Burns et al. [5])—are grouped according to the condition that they come close enough to sharing a common point of intersection, which we interpret as a candidate vanishing point. In the estimation step, one or more optima are chosen from among the results of the accumulation step, and a subsequent re-estimation of those optima is often computed vis-à-vis their respective inlier segments. As we confirm in our own experiments (cf. Chapter 5), small errors in those segments can lead to material inaccuracies in vanishing point estimates. Accordingly, the fact that we extract line segments from quantized noisy images makes developing an accurate and robust extraction technique a challenge, and—some four decades after the first approaches were published—the extraction of vanishing points consequently remains an active field of research.

Tessellating the Gaussian Sphere. The Euclidean unit sphere S^2 centered on the camera center $\mathbf{C} \in \mathbb{R}^3$ is (locally) topologically equivalent to the corresponding camera's image plane π (cf. Figure 2.1). Under known camera geometry, points in space and their projection onto π map two-to-one to antipodal points on this sphere,

---

[1] Extraction techniques employed in the navigation of robots and autonomous vehicles place real-time performance over quality, and are thus categorically ill-suited for our purposes. This is not true in general of techniques used in partial camera calibration.

and lines in space and their projection onto π map one-to-one to great circles. The intersection of two lines thus corresponds to the antipodal intersection of two great circles. One extraction strategy in the literature involves tessellating this sphere—also known as the Gaussian[2] sphere—and tallying the number of great circles that pass through each accumulation cell, where maxima are assumed to represent the vanishing points corresponding to dominant scene orientations. Note that this amounts to mapping to a Hough[3] space (cf. Hough [16] and Duda et al. [10]).

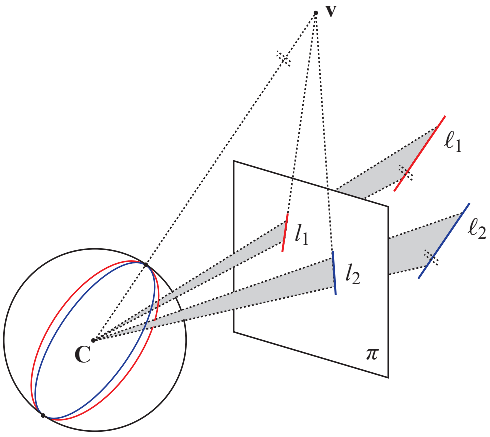


Figure 2.1: The Gaussian sphere, centered on the camera center $\mathbf{C}$. Under known camera geometry, a line ℓ and its projection l in the image plane π correspond to the same great circle on the Gaussian sphere, and the intersection of two lines $l_1, l_2 \subset \pi$ correspond to the intersection of their great circles. Note that the lines ℓ_1, ℓ_2 in space have the same orientation as the ray from $\mathbf{C}$ through their corresponding vanishing point $\mathbf{v}$. Note also, however, that since we do not assume that camera geometry is known, the assumed location of $\mathbf{C}$ with respect to π is at best a good guess, and the accuracy of that guess influences through which cells great circles pass.

Barnard [3] was the first to have availed himself of the Gaussian sphere as an accumulation space for extracting vanishing points. Quan and Mohr [32] improve upon Barnard's approach by carrying out a hierarchical sampling of their Hough space—thereby reducing the likelihood that veridical vanishing points fall on cell boundaries and go undetected—and by making use of a better tessellation. They observe that the quality of results obtained using their approach depends on how close the assumed focal length is to the veridical one; indeed, this is true of all approaches that rely on tessellating the Gaussian sphere, and results in fact depend more broadly on how closely the assumed location of $\mathbf{C}$ with respect to π corresponds to true camera geometry. Lutton et al. [26] first extract candidate vanishing points using a related Hough approach and subsequently use a second Hough approach to choose three vanishing points assumed to correspond closely to the scene's dominant three pairwise-orthogonal scene orientations. They discuss the influence of poor assumptions vis-à-vis camera geometry on

---

[2] Named after the German mathematician Johann Carl Friedrich Gauß (1777-1855), a Gaussian surface—of which a Gaussian sphere is a special case—is, according to its original meaning, a closed surface within the framework of Gauss' law of electric flux, which describes the relationship between the net electric flux through that closed surface and the charge it encloses.

[3] Note that much like the correct pronunciation of Lord Byron's *Don Juan* is /dɒn dʒuːən/, the correct pronunciation of Hough is /hʌf/.

the performance of their algorithm at greater length. Shufelt [37] observes that spurious maxima on the Gaussian sphere can arise both on account of weak perspective effects, and on account of textural effects leading to segments that do not correspond to dominant scene orientations. Accordingly, he introduces one Gaussian sphere technique that incorporates *a priori* knowledge about the geometry of objects of interest, and another that incorporates edge precision estimates in the aim of compensating for the influence of segments that arise from textural effects.

An advantage of using the Gaussian sphere as an accumulation space is that it allows for the unbounded image space to be mapped to a bounded space—thereby constraining the search space—and for infinite and finite vanishing points to be treated in the same manner. One disadvantage of approaches that rely on Hough transforms is that results depend on the chosen quantization (cf. Grimson et al. [12]). Another disadvantage involves the need to make guesses relating to camera geometry. A third disadvantage—observed in Rother [34]—is that the mapping of lines in the image to great circles on the Gaussian sphere does not preserve relative distances, which—as pointed out in Pflugfelder [31]—is a consequence of Girard's theorem.

The Intersection Constraint. For three lines in the image plane to be the projection of lines parallel in space, the normals of their interpretation planes must (ideally) be coplanar (cf. Section 3.1.1 of Chapter 3). This fact motivates van den Heuvel's [40] introduction of an *intersection constraint* for triplets of image segments, which states that three extracted image segments s_1, s_2, s_3, with corresponding interpretation plane normals $\mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3$, share a common vanishing point if the magnitude of $\det(\mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3)$ is below a tight threshold. Given n image segments and the $\binom{n}{3}$ possible triplets of interpretation plane normals, van den Heuvel rejects all triplets that do not satisfy his intersection constraint. He then carries out a clustering step over the remaining clusters, with clusters themselves constrained such that each triplet of interpretation plane normals they respectively contain satisfy the intersection constraint. Roughly speaking, the largest cluster is then chosen to correspond to the first vanishing point; another two are subsequently extracted, constrained to be collectively close to pairwise-orthogonal with the orientation estimated from the first vanishing point. Van den Heuvel thus finds three vanishing points assumed to correspond closely to the underlying scene's dominant three pairwise-orthogonal orientations without using the Hough transform. Note, however, that he too assumes that at least a good guess of the focal length is available.

The Image Plane as Accumulation Space. Magee and Aggarwal [28] compute the intersections of all $\binom{n}{2}$ pairs of lines through image segments and cluster them on the unit sphere. Rother [34] presents an approach that likewise operates over the set of all such intersections, but instead uses a voting scheme coupled with single-view constraints on camera geometry (cf. Liebowitz and Zisserman [24]). Part of Rother's contribution is a distance function $d(\mathbf{v}, s)$ for determining the extent to which an image line segment s corresponds to a given (candidate) vanishing point $\mathbf{v}$. Although we do not ourselves do so, the method upon we base our vanishing point extraction approach—namely, that of Sinha et al. [38]—makes use of this distance function. In the interest of completeness, we provide a more thorough summary of Rother's algorithm in Appendix A.

Expectation Maximization. Košecká and Zhang [20] cast the problem of extracting the vanishing points corresponding to the scene's dominant three pairwise-orthogonal orientations in terms of an expectation maximization (EM) framework. Pflugfelder [31] introduces his own EM framework, and integrates segment information over a video stream for a static camera. Advantages of using a video stream include greater robustness to single-frame sensor noise and the ability to incorporate additional dynamic information that may appear in the scene, due for instance to human activity or changes in lighting conditions. In both approaches, the extracted vanishing points are used to carry out partial camera calibration.

Extraction across Multiple Views. Werner et al. [41] present a multiple-view approach for extracting the dominant three pairwise-orthogonal orientations across k available uncalibrated views of the scene. They begin by computing vanishing points per view assumed to correspond closely to the scene's dominant three pairwise-orthogonal orientations, using RANSAC (cf. Appendix C). Next, they proceed to match those vanishing points combinatorially across the k views. Finally, they estimate the corresponding orientations in space by minimizing the reprojection error with respect to each corresponding vanishing point's inlier segments.

Sinha et al. [38] begin by computing up to n candidate vanishing points across each of the k available calibrated views of the scene. They then map each candidate's back-projection to a point on the unit sphere, and cluster over those points in the aim of obtaining three clusters corresponding closely to the dominant three pairwise-orthogonal orientations of the scene. Since it is upon their approach that we base ours, we provide a more detailed summary of their technique in Appendix B.

2.1.1 Estimation

Given a set $\mathcal{S}_{\mathbf{v}}$ of image segments determined to be inliers of a candidate vanishing point $\mathbf{v} \in \mathbb{P}^2$, Caprile and Torre [6] re-estimate $\mathbf{v}$ by computing a weighted mean of the intersections of the lines $\mathbf{l} \in \mathbb{P}^2$ corresponding to the segments $s \in \mathcal{S}_{\mathbf{v}}$. A more accurate approach involves fitting a point $\mathbf{v} \in \mathbb{P}^2$ to the set of lines $\mathbf{l} \in \mathbb{P}^2$ corresponding to the segments in $\mathcal{S}_{\mathbf{v}}$ by minimizing with respect to point-line incidence (cf. Collins and Weiss [8], Cipolla and Boyer [7]), which can be solved using the SVD (cf. Appendix D). An approach that produces potentially even better intersection estimations is the maximum likelihood (ML) intersection estimation technique of Liebowitz [22], which can be solved using an implementation of a non-linear least squares solver such as Levenberg-Marquardt (cf. Lourakis [25]). Pflugfelder [31] gives a comparison of the SVD and ML techniques with a mean approach. We discuss the SVD and ML intersection estimation techniques at greater length in Section 4.2 of Chapter 4.

2.2 Application to Scene Reconstruction

Feature Point-based Reconstruction. Rother [35] presents a system for using vanishing points extracted per view of a typical urban scene to initialize a feature point-based algorithm for the estimation of relative camera pose and calibration parameters. The vanishing points are assumed to correspond closely to pairwise-orthogonal scene orientations, and are constrained to be so in a vein that follows from the camera and orthogonality criteria of Rother [34]. Having computed feature point matches across views, he simultaneously recovers camera geometry and generates a sparse point cloud of the scene.

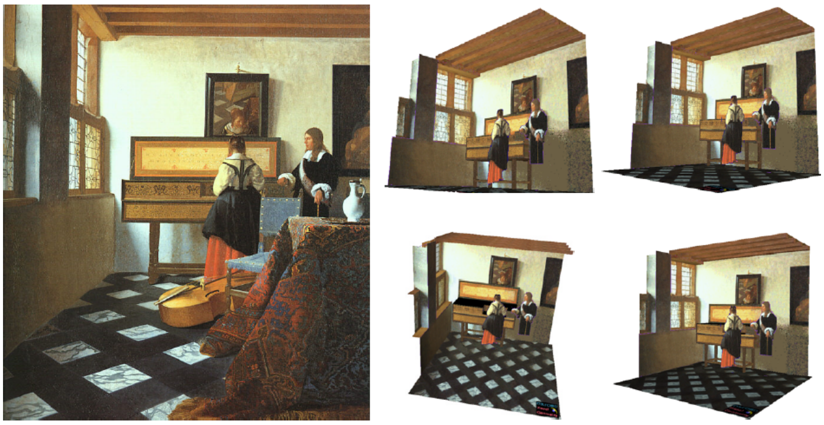


Figure 2.2: Scene reconstruction of Jan Vermeer's oil painting, *The Music Lesson* (1662-65), aided with knowledge of vanishing points. Figure from Criminisi [9].

Model-based Reconstruction. Criminisi [9] discusses the geometry of scene reconstruction aided with knowledge of vanishing points in detail. Using vanishing points extracted manually, he succeeds in producing several compelling reconstructions[4] of the scenes that paintings in correct perspective depict (cf. Figure 2.2).

Werner and Zisserman [41] extract the scene's three dominant pairwise-orthogonal scene orientations from vanishing points computed across k available views. Next, they sweep planes along those dominant orientations in the aim of generating a model of the scene (cf. Figure 2.3). Sinha et al. [38] use orientations extracted from per-view vanishing points estimates to provide 'snapping directions' to the user in addition to candidate plane orientations (cf. Figure 2.4).

---

[4] See `http://www.robots.ox.ac.uk/~vgg/projects/SingleView/` for VRML models and videos of such single-view reconstructions.

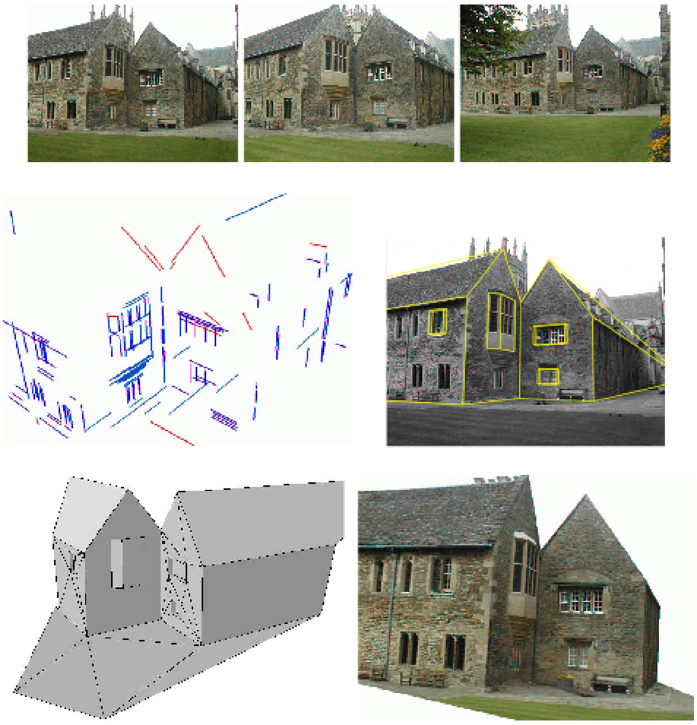
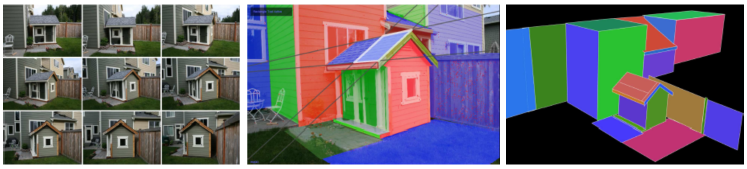
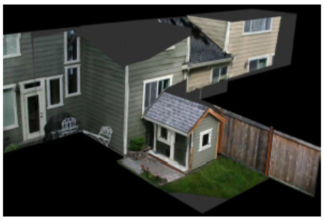


Figure 2.3: The urban scene reconstruction approach of Werner and Zisserman [41], which uses knowledge of vanishing points to guide a plane sweeping technique.



(a) Input photographs. (b) 2D sketching interface. (c) 3D geometric model.



(d) Texture-mapped model.

Figure 2.4: The urban scene reconstruction pipeline of Sinha et al. [38], which uses vanishing points extracted across multiple views to compute plane orientations and so-called ‘snapping directions’ corresponding to principal orientations of the scene.

Chapter 3

The Geometry of Vanishing Points

Adde parvum parvo magnus acervus erit.

—Ovid

The notion that a cube is composed of three sets of respectively parallel and mutually pairwise-orthogonal edges follows necessarily from the definition of a cube. Even so, regardless of the pose from which we observe a cube, we see that lines through the edges of at least one of those three sets invariably appear to converge in a corresponding point—called a finite[1] *vanishing point*—as illustrated in Figure 3.1. This would perhaps seem a contradiction, but for the fact that it follows necessarily from the manner in which light is projected through the lens of a human eye onto the eye's retina, or the lens system of a camera onto its photosensitive surface.

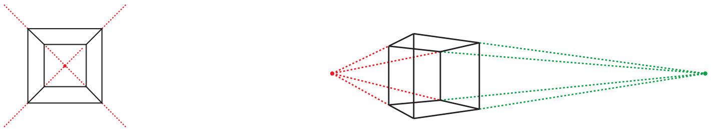
We begin our discussion of the geometry of vanishing points with an examination of the foundations of projective geometry, which provides a suitable framework for discussing how vanishing points arise. We then explore the process of image formation, which accounts for why they arise. Finally, we discuss vanishing points and vanishing *lines* in more detail, and conclude our discussion with a synopsis.

3.1 A Suitable Geometric Framework

The geometry of rigid bodies usually lends itself to adequate description within the framework of *Euclidean* geometry; in Euclidean geometry, we can measure the sides of objects, we can compute the angle between intersecting lines, and we can describe any two lines as parallel if they lie in the same plane and never meet. However rigid bodies—and with them whatever other kinds of bodies—are, with respect to how we *see* them, better served by *projective* geometry. Indeed, one of the shortcomings of Euclidean geometry in the plane is that provision must be made for two classes of line pairs: those that intersect in a point and those that—on account of being parallel—do not. Projective geometry does away with this distinction by elegantly augmenting the

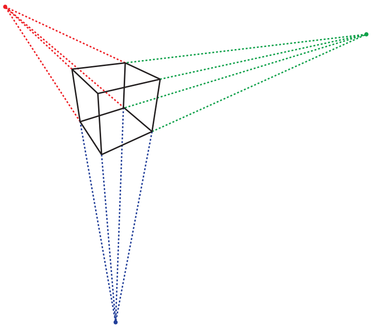
---

[1] We shall see that an *infinite* vanishing point is the point of intersection of the projection of lines parallel in space that are at the same time parallel to the image plane.



(a) One-point perspective.

(b) Two-point perspective.



(c) Three-point perspective.

Figure 3.1: A cube depicted in one-, two- and three-point perspective, terms borrowed from descriptive geometry. The number of points refer to the number of finite vanishing points in the corresponding view.

Euclidean plane with ‘ideal points’ that serve as the points of intersection of lines parallel in the plane. Moreover, projective geometry allows us to neatly model the central projection that underlies the image formation process, and which accounts for why vanishing points arise. We shall see that projective geometry thus offers a convenient formalism for our study of the geometry of vanishing points.

Several texts offer an excellent introduction to the geometric foundations that underlie the geometry of vanishing points. Among them, there figure Hartley and Zisserman [15], Ma et al. [27] and Springer [39]. This chapter—diagrams and matrices included—is based primarily on the expositions given in the first two. For a scholarly treatment of the techniques employed over the history of European art to render an impression of perspective (cf. Figure 3.2), refer to Andersen [1] or Kemp [18].

3.1.1 Homogeneous Coordinates

Points in $\mathbb{P}^n$. In Euclidean geometry, we represent points as n-dimensional ordered tuples $(x_1, \ldots, x_n)^\top \in \mathbb{R}^n$ called Euclidean coordinates. We can augment the Euclidean space $\mathbb{R}^n$ to the *projective space* $\mathbb{P}^n$ by representing all points in $\mathbb{R}^n$ as homogeneous $(n+1)$-dimensional vectors $(x_1, \ldots, x_n, 1)^\top \in \mathbb{P}^n = \mathbb{R}^{n+1} \setminus \{\mathbf{0}\}$.[2] We declare that a vector $(x_1, \ldots, x_n, 1)^\top \in \mathbb{P}^n$ and any vector $(kx_1, \ldots, kx_n, k)^\top \in \mathbb{P}^n$ for $k \neq 0$ represent the same point; that is, they belong to the selfsame equivalence class, since we are at all times accordingly free to scale one into the other. We indicate that two vectors $\mathbf{x}, \mathbf{x}' \in \mathbb{P}^2$ are equal to within a non-zero scalar factor k by using the notation $\mathbf{x} \sim \mathbf{x}' \Leftrightarrow \exists k \neq 0 : \mathbf{x} = k\mathbf{x}'$. In order to take the homogeneous vector $(kx_1, \ldots, kx_n, k)^\top \in \mathbb{P}^n, k \neq 0$, to its representation in inhomogeneous Euclidean

---

[2] We omit the vector $\mathbf{0} \in \mathbb{R}^{n+1}$ from $\mathbb{P}^n$ because—as we shall see—it represents neither a point nor an orientation of $\mathbb{R}^n$.

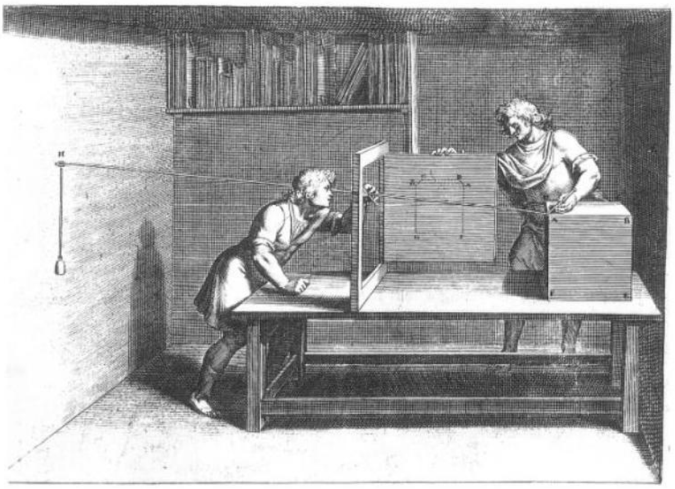


Figure 3.2: Illustration by French engineer Salomon de Caus (1612) of Albrecht Dürer's string method for producing a perspective composition. The point H on the wall is the center of projection (or *eye point*, as it is called in perspective drawing). Image reproduced from Andersen [1].

coordinates, we return all but the last coordinate, each of which we divide by k, giving $(x_1/k, \ldots, x_n/k)^\top \in \mathbb{R}^n$. For brevity, we shall accordingly often refer to such vectors $\mathbf{x}$ simply as points, even if they are in fact vectors that *represent* points.

All homogeneous vectors of $\mathbb{P}^2$ scaled such that $x_3 = 1$ lie in the plane $x_3 = 1$. We may think of the plane $x_3 = 1$ as an embedding of the Euclidean plane $\mathbb{R}^2$ in $\mathbb{P}^2$, given by the unit translation of the Euclidean plane $\mathbb{R}^2$ along the positive x_3-axis of the 3-dimensional Euclidean coordinate frame. Accordingly, we call the vector space $\mathbb{P}^2$ the *projective plane* (cf. Figure 3.3).

Points at Infinity in $\mathbb{P}^n$. Points in $\mathbb{P}^n$ with coordinates $(x_1, \ldots, x_n, 0)^\top$ are the *points at infinity* (or *infinite points*); in inhomogeneous Euclidean coordinates, we represent a point at infinity with a vector $(x_1/0, \ldots, x_n/0)^\top \in \mathbb{R}^n$, and we think of it accordingly as a point infinitely distant from the origin of the coordinate frame in the direction $(x_1, \ldots, x_n)^\top \in \mathbb{R}^n$. Since points at infinity thus have no real counterpart in $\mathbb{R}^n$, we also term them *ideal points*. Note, however, that infinite points are but ordinary points in $\mathbb{P}^n$. In addition to all points of $\mathbb{R}^n$, the projective space $\mathbb{P}^n$ thus contains points—namely, the ideal points—not present in $\mathbb{R}^n$.

Hyperplanes in $\mathbb{P}^n$. Let us consider the general form equation of a line $l \subset \mathbb{R}^2$ in the Euclidean plane,

$$ax_1 + bx_2 + c = 0. \tag{3.1}$$

Rewriting Equation (3.1) as the scalar product of two vectors,

$$(a, b, c)^\top (x_1, x_2, 1) = 0, \tag{3.2}$$

reveals an *incidence relation* between the homogeneous vector of a 2-dimensional point $\mathbf{x} \sim (x_1, x_2, 1)^\top \in \mathbb{P}^2$ and a second vector $\mathbf{l} \sim (a, b, c)^\top$, where we qualify two

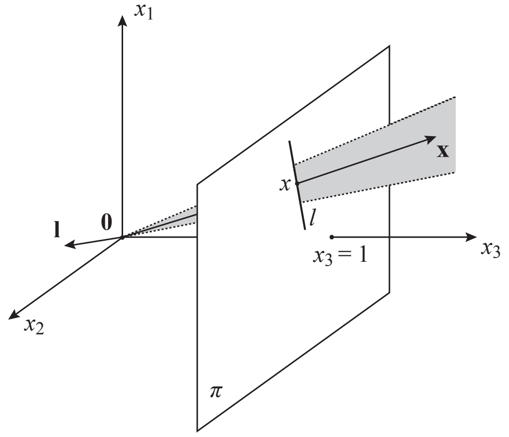


Figure 3.3: The projective plane $\mathbb{P}^2$. The vector $\mathbf{l} \in \mathbb{P}^3$ is the homogeneous normal vector of the plane through the origin that intersects the plane $x_3 = 1$ in the line $l \subset \mathbb{R}^2$, called the interpretation plane corresponding to l. A point $x \in \mathbb{R}^2$ is given in by the vector $\mathbf{x} \in \mathbb{P}^2$ through x, and thus $x \in l$ only if $\mathbf{l}^\top \mathbf{x} = 0$.

vectors as *incident* if they are orthogonal with respect to one another. Since scaling the vector $\mathbf{l}$ by a non-zero scalar has no effect on its incidence with $\mathbf{x}$, the vector $\mathbf{l}$—like the vector $\mathbf{x}$—is itself homogeneous. Interpreting the vector $\mathbf{l}$ as a normal vector of a plane through the origin of the coordinate frame—a plane we term the projective *interpretation plane* of the line l—we see that all vectors $(x_1, x_2, x_3)^\top \in \mathbb{P}^2$ that lie in that plane satisfy Equation (3.2). Rescaling all such incident homogeneous vectors such that $x_3 = 1$, we thus arrive at the set of all points that form the desired line l in the plane $x_3 = 1$. Geometrically, l is thus given by the intersection of the plane $x_3 = 1$ with the interpretation plane corresponding to $\mathbf{l}$. Accordingly, we understand the vector $\mathbf{l} \sim (a, b, c)^\top \in \mathbb{P}^2$ to *represent* the line $l \subset \mathbb{R}^2$ in question, and shall—again, in the service of brevity— often refer to such vectors $\mathbf{l}$ simply as lines. Analogously, we can extend the incidence relation in Equation (3.2) to n-dimensional points and hyperplanes through the origin of the coordinate frame of $\mathbb{P}^n$.

Hyperplanes at Infinity in $\mathbb{P}^n$. The line $\mathbf{l}_\infty \sim (0, 0, 1)^\top \in \mathbb{P}^2$, termed *line at infinity* (or the *ideal line*), is the line incident with all 2-dimensional ideal points $(x_1, x_2, 0)$, since $(0, 0, 1)^\top (x_1, x_2, 0) = 0$ for all x_1, x_2. The ideal points of $\mathbb{P}^2$ thus all lie in the plane $x_3 = 0$, which is the interpretation plane that corresponds to $\mathbf{l}_\infty$. Note that this plane is parallel to the plane $x_3 = 1$, which it consequently does not intersect (except, so to speak, 'at infinity'). In $\mathbb{P}^3$, we speak of the *plane at infinity* $\boldsymbol{\pi}_\infty \sim (0, 0, 0, 1)^\top$, which is the plane incident with all 3-dimensional ideal points $(\mathrm{X}_1, \mathrm{X}_2, \mathrm{X}_3, 0)^\top$ of $\mathbb{P}^3$. Analogously, we can extend the notion to n dimensions.

3.2 The Projective Plane $\mathbb{P}^2$

In our forthcoming discussion of image formation in Section 3.4, the projective plane $\mathbb{P}^2$ serves as the image plane of our camera model. In this section, we discuss point-line incidence in $\mathbb{P}^2$ and see how homogeneous coordinates allow us to neatly express the intersection of lines and the join of points in terms of the vector product. We determine that the vector $\mathbf{x} \sim (x_1, x_2, 0) \in \mathbb{P}^2$ that represents the intersection at infinity of two parallel lines $l, l' \subset \mathbb{R}^2$ has the selfsame orientation as l and l'. Accordingly, the points at infinity of $\mathbb{P}^2$ serve to represent the totality of orientations of the projective plane. We also explore the planar projective transformations, which characteristically preserve point-line incidence but do not in general guarantee that parallel lines be mapped to parallel lines. Projective transformations will, again, be of interest in Section 3.4, since the projection of a plane in space onto an image plane reduces to precisely a projective transformation.

3.2.1 Incidence, Collinearity and Concurrence

Two vectors are incident when their scalar product is zero, and a point $\mathbf{x} \in \mathbb{P}^2$ lies on a line $\mathbf{l} \in \mathbb{P}^2$ only if the vectors $\mathbf{x}$ and $\mathbf{l}$ are incident. Another way to think of point-line incidence is that the point $\mathbf{x}$ lies on a line $\mathbf{l}$ only if the vector $\mathbf{x}$ lies in the plane through the origin of $\mathbb{R}^3$ whose normal is the vector $\mathbf{l}$, recalling that we are at all times free to scale the vector $\mathbf{x} \sim (x_1, x_2, x_3)^\top$ such that $x_3 = 1$. Accordingly, all vectors corresponding to collinear points or concurrent lines are, respectively, coplanar.

The Line Joining Two Points. A consequence of the homogeneous representation of points is that the line $\mathbf{l} \in \mathbb{P}^2$ joining two points $\mathbf{x}, \mathbf{x}' \in \mathbb{P}^2$ is $\mathbf{x} \times \mathbf{x}' \sim \mathbf{l}$. This is because the vector $\mathbf{x} \times \mathbf{x}'$ is the unique homogeneous vector that is incident to both $\mathbf{x}$ and $\mathbf{x}'$. Indeed, by the triple scalar product identity, $\mathbf{x}^\top(\mathbf{x} \times \mathbf{x}') = \mathbf{x}'^\top(\mathbf{x} \times \mathbf{x}') = 0$.

The Intersection Point of Two Lines. The intersection of two lines $\mathbf{l}, \mathbf{l}' \in \mathbb{P}^2$ is the point $\mathbf{l} \times \mathbf{l}' \sim \mathbf{x} \in \mathbb{P}^2$. The proof is analogous to the argument given above for the line joining two points in $\mathbb{P}^2$. To see what happens when we compute the intersection of lines parallel in the Euclidean plane, let us consider the lines $\mathbf{l} \sim (a, b, c)^\top$ and $\mathbf{l}' \sim (a, b, c')^\top$. One way to see that the corresponding lines $l, l' \subset \mathbb{R}^2$ in the Euclidean plane are parallel is by observing that their respective slopes are both $-a/b$.[3] Their point of intersection is then

$$\mathbf{x} \sim \mathbf{l} \times \mathbf{l}' \sim \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ a & b & c \\ a & b & c' \end{vmatrix} \sim \begin{pmatrix} b \\ -a \\ 0 \end{pmatrix}, \tag{3.3}$$

which is a point at infinity. This demonstrates that, contrary to the state of affairs in the Euclidean plane, two parallel lines always meet in a point—albeit an ideal point—in the projective plane. Finally, we note that since two lines $l, l' \subset \mathbb{R}^2$ with identical slope $-a/b$ meet in the ideal point $\mathbf{x} \sim (b, -a, 0)^\top$, it follows that the vector $\mathbf{x}$ and the lines l, l' all share the same *orientation*. One way to satisfy ourselves that this is true is to consider that a slope $-a/b$ represents a per-unit displacement in the Euclidean plane by b units in the x-direction, and one of $-a$ units in the y-direction, which amounts to precisely a displacement by the vector $(b, -a, 0)^\top$.

Collinearity of Three Points. Three points $\mathbf{x}, \mathbf{x}', \mathbf{x}'' \in \mathbb{P}^2$ all lie on a line $\mathbf{l} \in \mathbb{P}^2$ if, without loss of generality, the vector that represents the line $\mathbf{l} \sim \mathbf{x}' \times \mathbf{x}''$ joining

---

[3] We recall from gradeschool mathematics that by rewriting the general form equation $ax + by + c = 0$ of a line in slope-intercept form, we obtain $y = -(a/b)x - c/b$.

two of the points is incident with the vector that represents the third, i.e., $\mathbf{l}^\top \mathbf{x} = (\mathbf{x}' \times \mathbf{x}'')^\top \mathbf{x} = 0$. All three vectors $\mathbf{x}, \mathbf{x}', \mathbf{x}''$ must therefore be coplanar. Equivalently, we can articulate this requirement as $\det(\mathbf{x}, \mathbf{x}', \mathbf{x}'') = 0$.[4]

Concurrence of Three Lines. Three lines $\mathbf{l}, \mathbf{l}', \mathbf{l}'' \in \mathbb{P}^2$ are incident with the same point $\mathbf{x} \in \mathbb{P}^2$ (i.e., they are concurrent), when $\det(\mathbf{l}, \mathbf{l}', \mathbf{l}'') = 0$. The proof is analogous to the argument given above for the collinearity of three points in $\mathbb{P}^2$. This is the foundation of the 'intersection constraint' van den Heuvel [40] uses in his single-view vanishing point extraction approach (cf. Chapter 2).

3.2.2 Duality of Points and Lines

In our discussion of incidence, collinearity and concurrence, we have seen that the role of points and lines can be interchanged. Indeed, to every theorem of the projective plane $\mathbb{P}^2$ there exists a dual theorem of $\mathbb{P}^2$ obtained by substituting points for lines and lines for points. This follows from the symmetry of the incidence relation.

3.2.3 Projective Transformations of $\mathbb{P}^2$

Geometrically, a *projective transformation* (synonymously termed a *homography*, a *collineation* or a *projectivity*) of $\mathbb{P}^2$ is an invertible mapping $h : \mathbb{P}^2 \rightarrow \mathbb{P}^2$ that preserves point-line incidence, and thus maps lines to lines (hence the term 'collineation'). Algebraically, a mapping h is a projectivity if and only if there exists a 3×3 invertible matrix $\mathtt{H}$ such that, for any $\mathbf{x} \in \mathbb{P}^2$, it holds that $h(\mathbf{x}) \sim \mathtt{H}\mathbf{x}$. Indeed, if three collinear points $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3 \in \mathbb{P}^2$ lie on a line $\mathbf{l} \in \mathbb{P}^2$, then each $\mathbf{x}'_i \sim \mathtt{H}\mathbf{x}_i, i \in \{1, 2, 3\}$, lies on the line $\mathbf{l}' \sim \mathtt{H}^{-\top}\mathbf{l}$, since $\mathbf{l}'^\top \mathbf{x}'_i = (\mathtt{H}^{-\top}\mathbf{l})^\top \mathtt{H}\mathbf{x}_i = \mathbf{l}^\top \mathtt{H}^{-1}\mathtt{H}\mathbf{x}_i = \mathbf{l}^\top \mathbf{x}_i = 0, i \in \{1, 2, 3\}$. Accordingly, a projectivity h represented by an invertible 3×3 matrix $\mathtt{H}$ transforms a point $\mathbf{x} \in \mathbb{P}^2$ to the point $\mathbf{x}' \sim \mathtt{H}\mathbf{x}$ and a line $\mathbf{l} \in \mathbb{P}^2$ to the line $\mathbf{l}' \sim \mathtt{H}^{-\top}\mathbf{l}$, and point-line incidence is thus preserved. Note that the matrix $\mathtt{H}$ is, again, itself homogeneous, since scaling $\mathtt{H}$ by a non-zero scalar has no effect on the outcome of the corresponding projective transformation.

In the spirit of Klein's Erlangen program [19], a projective transformation is characterized by the geometric properties invariant to it. General projective transformations given by arbitrary invertible 3×3 matrices form a group called the *projective linear group* on three dimensions. All projectivities preserve incidence (and with it collinearity and concurrence) and a measure called the cross ratio. Meanwhile, the projective linear group on three dimensions encompasses a hierarchy of nested subgroups of transformations that feature increasingly specialized invariants in addition to the invariants of their respective encompassing supergroups. Accordingly, the Euclidean transformations are a subgroup of the similarities, the similarities a subgroup of the affinities, and the affinities a subgroup of the general projectivities. In addition to their own specialized invariants, the Euclidean transformations thus preserve all the invariants of the similiarities, the similarities all the invariants of the affinities, and the affinities all the invariants of the projectivities.

With respect to invariance, our focus is on the effect that projectivities have on the line at infinity $\mathbf{l}_\infty$, since the transformation of $\mathbf{l}_\infty$ to a finite line $\mathbf{l}$ accounts for parallel lines being projected to lines that meet in a finite point. For a more detailed

---

[4] Interpreting the determinant of three vectors in $\mathbb{P}^2$ as the volume of the parallelepiped spanned by three vectors in $\mathbb{R}^3$, we correctly arrive at a volume of zero if the three vectors are coplanar.

treatment of the properties invariant to the projective linear group on three dimensions and its subgroups, as well as for an explanation of the cross ratio, refer to Hartley and Zisserman [15].

Euclidean Transformations. The Euclidean transformations (also referred to as the *isometries* or *displacements*) of the plane are the planar rotations, translations and reflections. They preserve length and angle, in addition to the affine invariants, namely ratio of lengths, parallelism, incidence (and with it collinearity and concurrence) and the cross ratio. The form of the general Euclidean transfomation matrix is

$$\mathtt{H}_{\mathrm{E}} \sim \begin{bmatrix} \epsilon\cos\theta & -\sin\theta & t_x \\ \epsilon\sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathtt{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}, \tag{3.4}$$

where $\epsilon = \pm 1$ and $\mathtt{R}$ is an arbitrary 2×2 orthogonal matrix. Like the similarities, the Euclidean transformations map the line at infinity $\mathbf{l}_\infty$ to itself, and—since Euclidean transfomations preserve incidence—points at infinity to points at infinity.

Similarity Transformations. The similarities of the plane encompass uniform scaling in addition to rotations, translations and reflections. Similarities preserve all the properties that affinities preserve, in addition to angle and ratio of lengths. Collectively, similarities thus happen to preserve all the invariants that Euclidean transformations do, except length; i.e., the Euclidean properties, defined up to scale. We term these invariants the *metric* properties. The form of the matrix of a general similarity transformation is

$$\mathtt{H}_{\mathrm{S}} \sim \begin{bmatrix} s\cos\theta & -s\sin\theta & t_x \\ s\sin\theta & s\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s\mathtt{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}, \tag{3.5}$$

where $s \in \mathbb{R}$ and $\mathtt{R}$ is an arbitrary 2×2 orthogonal matrix. Like affinities, Euclidean transformations also map the line at infinity $\mathbf{l}_\infty$ to itself, and consequently points at infinity to points at infinity.

Affine Transformations. In addition to uniform scaling, rotations, translations and reflections, the affinities also feature non-uniform scaling. They preserve the invariants that general projectivities preserve, together with parallelism. The form of the matrix of a general affinity is

$$\mathtt{H}_{\mathrm{A}} \sim \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}. \tag{3.6}$$

The line at infinity $\mathbf{l}_\infty \sim (0, 0, 1)^\top \in \mathbb{P}^2$ is invariant under the affinities (and consequently the similarities and the Euclidean transfomations as well), since

$$\mathtt{H}_{\mathrm{A}}^{-\top}\mathbf{l}_\infty \sim \begin{bmatrix} \mathtt{A}^{-\top} & \mathbf{0} \\ -\mathbf{t}^\top\mathtt{A}^{-\top} & 1 \end{bmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \sim \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \sim \mathbf{l}_\infty. \tag{3.7}$$

Under affinities, points at infinity thus remain at infinity. Note, however, that a point at infinity $\mathbf{x} \sim (x_1, x_2, 0)^\top$ is not mapped to *itself* unless there exists a non-zero scalar k

such that $\mathtt{A}(x_1, x_2)^\top = k(x_1, x_2)^\top$, since

$$\mathbf{x}' \sim \begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 0 \end{pmatrix} \sim \left(\begin{array}{c} \mathtt{A} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ 0 \end{array} \right). \tag{3.8}$$

In other words, for a point at infinity $\mathbf{x} \sim (x_1, x_2, 0)^\top$ to be mapped to itself, the vector $(x_1, x_2)^\top$ must be an eigenvector of the matrix $\mathtt{A}$.

Projective Transformations. The general projectivities encompass all of rotations, translations, reflections, uniform and non-uniform scaling, central projection between planes and all compositions of projectivities. With respect to invariants, general projectivities preserve only incidence (and with it collinearity and concurrence) and the cross ratio. The general projectivity is given by an arbitrary invertible 3×3 matrix

$$\mathtt{H}_\mathtt{P} \sim \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ v_1 & v_2 & v \end{bmatrix} = \begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{v}^\top & v \end{bmatrix}. \tag{3.9}$$

The line at infinity $\mathbf{l}_\infty$ is *not* invariant under general projective transformations, since $\mathtt{H}_\mathtt{P}$ can be any invertible 3×3 matrix. What this amounts to is that $\mathbf{l}_\infty$ is—unless the projectivity is an affinity—transformed to a finite line, and the points at infinity are thus transformed to points[5] incident with this finite line, according to

$$\mathbf{x}' \sim \begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{v}^\top & v \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 0 \end{pmatrix} \sim \left(\begin{array}{c} \mathtt{A} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ v_1 x_1 + v_2 x_2 \end{array} \right). \tag{3.10}$$

3.3 The Projective Space $\mathbb{P}^3$

Like the projective plane $\mathbb{P}^2$ is an augmentation of the Euclidean plane $\mathbb{R}^2$ with the set of ideal points $(x_1, x_2, 0)^\top \in \mathbb{P}^2$, so too is projective 3-space $\mathbb{P}^3$ an augmentation of Euclidean 3-space $\mathbb{R}^3$ with the set of ideal points $(\mathbf{d}^\top, 0)^\top = (\mathrm{X}_1, \mathrm{X}_2, \mathrm{X}_3, 0)^\top \in \mathbb{P}^3$. Moreover, like the points at infinity of $\mathbb{P}^2$ represent the totality of orientations of the projective plane, so too do the points at infinity of $\mathbb{P}^3$ represent the orientations of projective 3-space. Analogously to the fact that the line at infinity $\mathbf{l}_\infty \sim (0, 0, 1)^\top \in \mathbb{P}^2$ contains all orientations of $\mathbb{P}^2$, the plane at infinity $\boldsymbol{\pi}_\infty \sim (0, 0, 0, 1)^\top \in \mathbb{P}^3$ contains all orientations of $\mathbb{P}^3$. A more thorough discussion of projective 3-space is available in Hartley and Zisserman [15]. The facts of projective 3-space we have hereby presented, however, will suffice for the remainder of our discussion.

3.4 Image Formation

Vanishing points arise on account of how image formation works. Image formation is the process of projecting points in 3-dimensional space to a 2-dimensional image plane,

---

[5] We refrain from qualifying the totality of these transformed points as finite for good reason. In particular, all but one of the infinitude of points incident with a finite line are themselves finite; this is because every finite line $(a, b, c)^\top \in \mathbb{P}^2$ is incident with the infinite point $(b, -a, 0)^\top \in \mathbb{P}^2$, which is the mapping of the unique point in projective 3-space for which $v_1 x_1 + v_2 x_2 = 0$.

in our case in the manner done by a typical consumer-level digital camera. Rather than try to account for the totality of physical phenomena that come into play when we take a photograph, we satisfy ourselves with a simplified camera model that allows us to better understand the geometry in which we are interested. The model we choose is called the finite projective camera. We carry out our discussion stepwise, beginning with an examination of the thin lens camera, which we subsequently generalize to the pinhole camera, which we in turn finally generalize to the finite projective camera.

A more thorough but still readable introduction to image formation is given in Ma et al. [27] and Hartley and Zisserman [15], the both of which serve collectively as the main sources for this section. The classic textbook on the physics of image formation is reputably Born and Wolf [4].

3.4.1 The Thin Lens Camera

A typical consumer-level digital camera is composed of a system of one or more lenses used to refract light onto a photosensitive sensor (or surface) such as a CCD chip. The simplest and most specialized model of such a camera is the *thin lens*, which we illustrate in Figure 3.4. Perpendicular to a single ideal double-convex (and consequently symmetric and converging) lens,[6] the *optical axis* (or *principal axis*) crosses the center of the lens at a point called the *optical center* (or *camera center*) $\mathbf{C}$. By definition, rays of light emanating from a point $\mathbf{X}$ on the opposite side cross the lens according to the following refraction rules:

i. the lens refracts incident rays parallel to the optical axis such that they invariably pass through a point on the optical axis called the *focal point* (or *focus*), lying at a distance f called the *focal length* (or *camera constant*) from $\mathbf{C}$;

ii. incident rays passing through the opposite side's focal point (also at a distance f from the lens) are refracted such that they continue parallel to the optical axis;

iii. incident rays passing through $\mathbf{C}$ cross the lens undeflected.

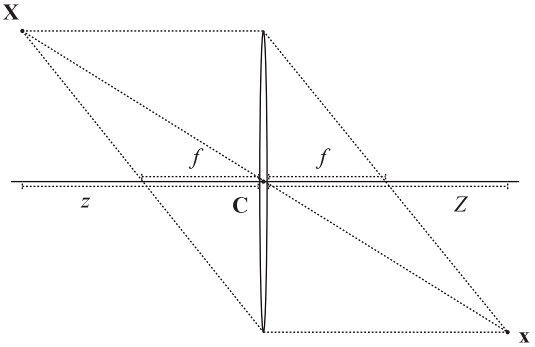


Figure 3.4: The thin lens camera model.

The *image* $\mathbf{x}$ of the point $\mathbf{X}$ lies at the point of intersection of any two rays obtained by the above rules. Note that the projected point $\mathbf{x}$ is upside down with respect to the projecting point $\mathbf{X}$.

---

[6] The English *lens* derives ultimately from the identically spelled Latin word for 'lentil', owing to the lentil-like shape of a double-convex lens.

As an aside, let the point $\mathbf{X}$ lie at a distance Z from the lens, and its image $\mathbf{x}$, at a distance z from the lens on the opposite side. Using similar triangles, we obtain

$$\frac{1}{Z} + \frac{1}{z} = \frac{1}{f}, \tag{3.11}$$

which is the *fundamental equation of the thin lens*.

3.4.2 The Pinhole Camera

As we shrink the aperture of a thin lens camera towards zero, the only rays of light allowed to reach the image plane are—in the limit—those that pass through the optical center. The resulting construction is called a *pinhole camera*, and models a camera that directs light onto its image plane using not a lens, but—like a *camera obscura*—only a tiny aperture, a 'pinhole'.

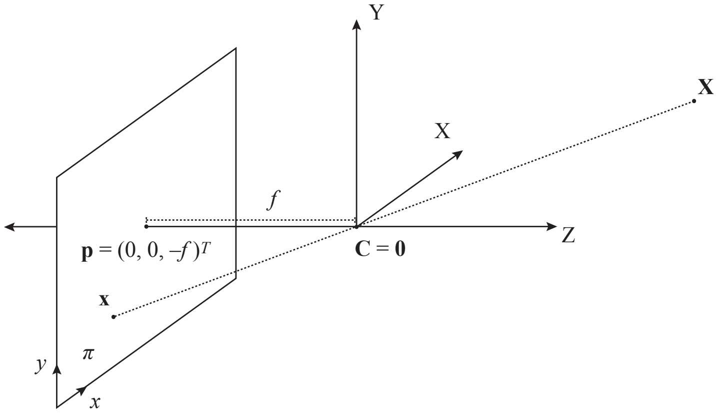


Figure 3.5: The pinhole camera model.

According to the model, a point $\mathbf{X} = (\mathrm{X}, \mathrm{Y}, \mathrm{Z})^\top \in \mathbb{R}^3$ in space projects to the point $\mathbf{x} = (x, y)^\top \in \mathbb{R}^2$ on the image plane π through the optical center $\mathbf{C} \in \mathbb{R}^3$, such that $\mathbf{X} \neq \mathbf{C}$ and $\mathbf{C} \notin \pi$, are related by the central projection. The *central projection* (or *perspective projection*) is a general mathematical formulation[7] of the projection from 3-dimensional space onto a 2-dimensional image plane π through a point $\mathbf{C}$, $\mathbf{C} \notin \pi$, that serves as the *center of projection*. Given a point $\mathbf{X} = (\mathrm{X}, \mathrm{Y}, \mathrm{Z})^\top \in \mathbb{R}^3$, $\mathbf{X} \neq \mathbf{C}$, the projection maps $\mathbf{X}$ to a point $\mathbf{x} \in \pi$ obtained by intersecting the plane π with the line joining $\mathbf{C}$ and $\mathbf{X}$. Assuming that $\mathbf{C}$ lies at the origin of $\mathbb{R}^3$ and that π is the plane $\mathrm{Z} = -f$, the corresponding mapping from $\mathbb{R}^3$ to $\mathbb{R}^2$ (cf. Figure 3.5) is given, again using similar triangles, by

$$(\mathrm{X}, \mathrm{Y}, \mathrm{Z})^\top \mapsto (-f\mathrm{X}/\mathrm{Z}, -f\mathrm{Y}/\mathrm{Z})^\top. \tag{3.12}$$

Note that the projected point $\mathbf{x}$ is—as was the case with the thin lens camera model—upside down with respect to its projecting point $\mathbf{X}$. In order to eliminate this effect, we flip the image plane π according to the mapping $(x, y) \mapsto (-x, -y)$. Doing so

---

[7] Note that usage varies; in the photogrammetry literature, the 'central projection' (or perspective projection) is understood more broadly to conceptually encompass the 'pinhole camera model' of the computer vision literature (cf. Mundy [30]).

is equivalent to placing the image plane $\mathrm{Z} = -f$ on the opposite side of the lens at $\mathrm{Z} = f$, and corresponds to the *frontal pinhole camera* (cf. Figure 3.6) model given by

$$(\mathrm{X}, \mathrm{Y}, \mathrm{Z})^\top \mapsto (f\mathrm{X}/\mathrm{Z}, f\mathrm{Y}/\mathrm{Z})^\top. \tag{3.13}$$

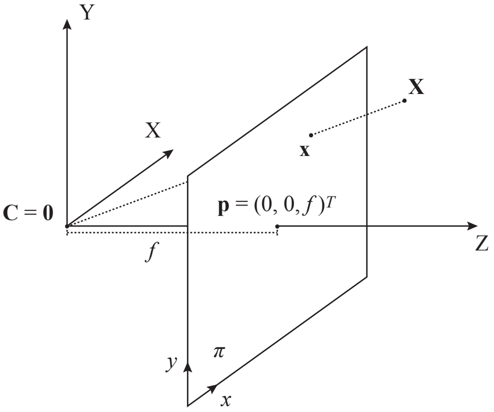


Figure 3.6: The frontal pinhole camera model.

Using instead the homogeneous coordinates of projective 3-space $\mathbb{P}^3$, we can reformulate the mapping in (3.13) as a matrix multiplication,

$$\begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} f\mathrm{X} \\ f\mathrm{Y} \\ \mathrm{Z} \end{pmatrix} = \begin{bmatrix} f & & & 0 \\ & f & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ 1 \end{pmatrix}, \tag{3.14}$$

which expresses the central projection as a *linear* mapping between the respective homogeneous coordinates of a point in space and its projection on the image plane. We call the 3×4 matrix in (3.14) the *camera projection matrix* $\mathtt{P}$, which we can further decompose as

$$\mathtt{P} \sim \begin{bmatrix} f & & \\ & f & \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{3.15}$$

where we call the 3×4 matrix the *standard* (or *canonical*) *projection matrix* and the 3×3 matrix the *camera calibration matrix* $\mathtt{K}$. We express this decomposition more compactly using block notation as

$$\mathtt{P} \sim \mathtt{K}[\mathtt{I} \mid \mathbf{0}]. \tag{3.16}$$

3.4.3 The Finite Projective Camera

Manufacturing defects such as the misalignment of a camera's lens with its photosensitive surface (e.g., a CCD chip) or physical imperfections in its lens system cause the model of an ideal pinhole camera as presented in (3.14) to be ill-suited to adequately modeling the geometry of image formation. Accordingly, we make the appropriate modifications to the camera calibration matrix $\mathtt{K}$ introduced above to obtain the *finite projective camera*. Note that in our discussion of the finite projective camera, we understand a vector $\mathbf{X}$ to be a homogeneous vector in $\mathbb{P}^3$, and a vector $\tilde{\mathbf{X}}$ to be its inhomogeneous counterpart in $\mathbb{R}^3$.

Principal Point. The point of intersection $(p_x, p_y) \in \mathbb{R}^2$ between the optical axis and the image plane π is called the *principal point*. The (frontal) pinhole camera model given above assumes that the principal point lies at the origin $\mathbf{0} \in \mathbb{R}^2$ of the image coordinate frame. If, however, the optical axis is not orthogonal to π, the principal point lies elsewhere in the image plane. In order to account for this effect, we rewrite the mapping in (3.13) as

$$(\mathrm{X}, \mathrm{Y}, \mathrm{Z})^\top \mapsto (f\mathrm{X}/\mathrm{Z} + p_x, f\mathrm{Y}/\mathrm{Z} + p_y)^\top. \tag{3.17}$$

In the homogeneous coordinates of $\mathbb{P}^3$, this mapping is given by

$$\begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} f\mathrm{X} + \mathrm{Z}p_x \\ f\mathrm{Y} + \mathrm{Z}p_y \\ \mathrm{Z} \end{pmatrix} = \begin{bmatrix} f & & p_x & 0 \\ & f & p_y & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ 1 \end{pmatrix}, \tag{3.18}$$

and, accordingly, the 3×3 camera calibration matrix $\mathtt{K}$ becomes

$$\mathtt{K} \sim \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}. \tag{3.19}$$

More compactly, the mapping in (3.18) is

$$\mathbf{x} \sim \mathtt{K}[\mathtt{I} \mid \mathbf{0}]\mathbf{X}_{\mathrm{cam}}, \tag{3.20}$$

where $\mathbf{X}_{\mathrm{cam}} \in \mathbb{P}^3$ is understood to be a point in space given—in what is called the *camera coordinate frame*—with respect to the camera assumed to be located at the origin of $\mathbb{R}^3$ and with its optical axis pointing in the direction of the positive Z-axis.

Pixels in CCD Cameras. So far, we have assumed that the image coordinates are Euclidean coordinates, with equal scale in both axial directions. In order to account for the fact that our image plane is tessellated into pixels, we modify our 3×3 camera calibration matrix $\mathtt{K}$ accordingly, yielding

$$\mathtt{K} \sim \begin{bmatrix} m_x f & & m_x p_x \\ & m_y f & m_y p_y \\ & & 1 \end{bmatrix} = \begin{bmatrix} \alpha_x & & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix}, \tag{3.21}$$

where m_x, m_y give the number of pixels per unit distance along the x- and y-directions in image coordinates. Although omitted for most normal cameras, we may also include a parameter s, which expresses a measure of pixel skew. Accordingly, our matrix $\mathtt{K}$ becomes

$$\mathtt{K} \sim \begin{bmatrix} \alpha_x & s & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix}. \tag{3.22}$$

Camera Pose. The *pose* of a camera refers collectively to its position and to the direction in which it is facing, with respect to the *world coordinate frame*. Given a vector $\tilde{\mathbf{X}} \in \mathbb{R}^3$ representing a point's position in world coordinates and given some finite projective camera $\mathtt{P}$, the same point $\tilde{\mathbf{X}}_{\mathrm{cam}} \in \mathbb{R}^3$ in camera coordinates is related to $\tilde{\mathbf{X}}$ in world coordinates by the Euclidean transformation

$$\tilde{\mathbf{X}}_{\mathrm{cam}} = \mathtt{R}(\tilde{\mathbf{X}} - \tilde{\mathbf{C}}), \tag{3.23}$$

where $\tilde{\mathbf{C}} \in \mathbb{R}^3$ represents the position of the camera center in world coordinates and $\mathtt{R}$ is a 3×3 rotation matrix that gives the orientation of the camera coordinate frame with respect to the world coordinate frame. We can reformulate the transformation in Equation (3.23) in terms of the homogeneous coordinates of $\mathbb{P}^3$ as

$$\mathbf{X}_{\text{cam}} \sim \begin{bmatrix} \mathtt{R} & -\mathtt{R}\tilde{\mathbf{C}} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ 1 \end{pmatrix} = \begin{bmatrix} \mathtt{R} & -\mathtt{R}\tilde{\mathbf{C}} \\ \mathbf{0}^\top & 1 \end{bmatrix} \mathbf{X}, \tag{3.24}$$

where $\mathbf{X}$ is in the world coordinate frame. By substituting the rightmost expression in Equation (3.24) for $\tilde{\mathbf{X}}_{\text{cam}}$ in Equation (3.20), we obtain

$$\mathbf{x} \sim \mathtt{KR}[\mathtt{I} \mid -\tilde{\mathbf{C}}]\mathbf{X}, \tag{3.25}$$

and our final finite projective camera projection matrix $\mathtt{P}$ is accordingly

$$\mathtt{P} \sim \mathtt{KR}[\mathtt{I} \mid -\tilde{\mathbf{C}}] = \mathtt{K}[\mathtt{R} \mid -\mathtt{R}\tilde{\mathbf{C}}] = \mathtt{K}[\mathtt{R} \mid \mathbf{t}], \tag{3.26}$$

where $\mathbf{t} = -\mathtt{R}\tilde{\mathbf{C}}$. The 3×4 matrix $\mathtt{P}$ thus maps 3-dimensional points in the world coordinate frame to 2-dimensional points in the image coordinate frame. Note that the last column $\mathtt{K}\mathbf{t}$ of the matrix $\mathtt{P}$ is the projection of the origin $(0, 0, 0, 1)^\top \in \mathbb{P}^3$ of the world coordinate frame.

3.4.4 Mappings between Planes

In mapping between planes by the central projection, point-line incidence is preserved (cf. Figure 3.7). Accordingly, we can represent any plane-to-plane mapping given by a finite projective camera as a planar projectivity $h : \mathbb{P}^2 \to \mathbb{P}^2$, which we can express using an invertible 3×3 matrix $\mathtt{H}$. The only requirements are that a coordinate system be defined in both planes and that points be represented using homogeneous vectors. Consequently, lines parallel in a projecting plane are projected onto the image plane as lines that meet in a finite point, unless the projectivity is an affinity.

3.4.5 Forward Projection

As we have seen, given the camera matrix $\mathtt{P}$ of a finite projective camera, the corresponding projection of a point in space given by the vector $\mathbf{X} \in \mathbb{P}^3$ to a point in the image plane given by the vector $\mathbf{x} \in \mathbb{P}^2$ is obtained by

$$\mathbf{x} \sim \mathtt{P}\mathbf{X}. \tag{3.27}$$

In the case of infinite points $\mathbf{D} \sim (\mathbf{d}^\top, 0)^\top \in \mathbb{P}^3$, which represent the orientations of projective 3-space, the mapping simplifies to

$$\mathbf{x} \sim \mathtt{P}\mathbf{D} = \mathtt{KR}[\mathtt{I} \mid -\tilde{\mathbf{C}}]\mathbf{D} = [\mathtt{KR} \mid -\mathtt{KR}\tilde{\mathbf{C}}]\mathbf{D} = [\mathtt{M} \mid \mathbf{p}_4]\mathbf{D} \sim \mathtt{M}\mathbf{d}, \tag{3.28}$$

where $\mathtt{M} = \mathtt{KR}$ is an invertible[8] 3×3 matrix and $\mathbf{p}_4 = -\mathtt{M}\tilde{\mathbf{C}}$ is the last column of the matrix $\mathtt{P}$. As we shall see in Section 3.5, $\mathtt{M}\mathbf{d}$ is precisely the vanishing point $\mathbf{v}$ incident with the projection of *every line* ℓ in space that shares the orientation of the vector $\mathbf{D}$.

---

[8] Were the matrix $\mathtt{M}$ non-invertible, then $\mathtt{P} \sim [\mathtt{M} \mid \mathbf{p}_4]$ would represent an *infinite* projective camera (or *affine camera*), which has its camera center at infinity and thus models a parallel projection.

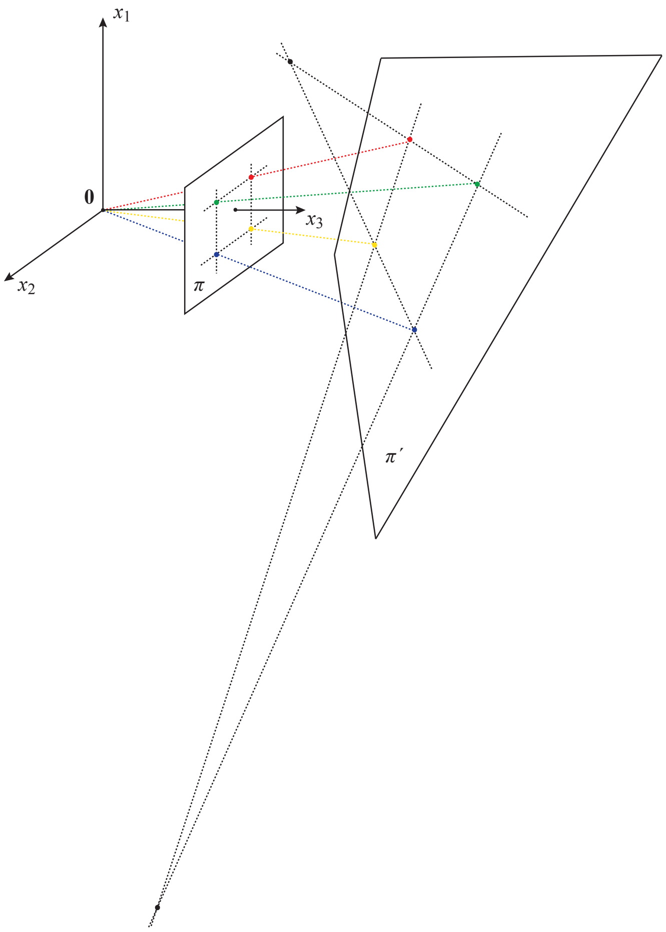


Figure 3.7: A mapping between planes by the central projection preserves point-line incidence. Accordingly, we can represent such a mapping using a planar projectivity.

3.4.6 Back-Projection

Given a vector $\mathbf{x} \in \mathbb{P}^2$ corresponding to a point in the image, its *back-projection* is the set of all points $\mathbf{X} \in \mathbb{P}^3$ in space that $\mathtt{P}$ maps to $\mathbf{x}$. The back-projection is thus given by the ray through the camera center passing through the image point in question. Since we can decompose $\mathtt{P}$ as

$$\mathtt{P} \sim \mathtt{KR}[\mathtt{I} \mid -\tilde{\mathbf{C}}] = \mathtt{M}[\mathtt{I} \mid \mathtt{M}^{-1}\mathbf{p}_4], \tag{3.29}$$

it follows that the camera center $\tilde{\mathbf{C}} \in \mathbb{R}^3$ is given by $\mathtt{M}^{-1}\mathbf{p}_4$, and thus

$$\mathbf{C} \sim \begin{pmatrix} \mathtt{M}^{-1}\mathbf{p}_4 \\ 1 \end{pmatrix}. \tag{3.30}$$

A second point on the ray is given by the ray's intersection with the plane at infinity $\boldsymbol{\pi}_\infty \sim (0,0,0,1)^\top$,

$$\mathbf{D} \sim \begin{pmatrix} \mathtt{M}^{-1}\mathbf{x} \\ 0 \end{pmatrix}. \tag{3.31}$$

Indeed, every point along the ray can be obtained by the parameterization $\mathbf{C} + \lambda\mathbf{D}$, for the appropriate $\lambda \in \mathbb{R}$. Accordingly, a vanishing point $\mathbf{v} \in \mathbb{P}^2$ back-projects to its corresponding orientation in space, since if $\mathbf{v} \sim \mathtt{P}(\mathbf{d}^\top, 0)^\top \sim \mathtt{M}\mathbf{d}$ represents the vanishing point corresponding to the orientation $(\mathbf{d}^\top, 0)^\top$, then the back-projection of $\mathbf{v}$ itself has the orientation $((\mathtt{M}^{-1}\mathtt{M}\mathbf{d})^\top, 0)^\top \sim (\mathbf{d}^\top, 0)^\top$.

3.5 Vanishing Points

We have already seen that every orientation projects to a vanishing point whose back-projection is a ray along the original orientation. Our concern, however, is the relationship between *lines* ℓ in space and vanishing points in the image plane. Given the projection matrix $\mathtt{P}$ of a finite projective camera and the parameterization $\mathbf{X}(\lambda) \sim \mathbf{A} + \lambda\mathbf{D}$ in homogeneous world coordinates of a line ℓ in space such that $\mathbf{D} \sim (\mathbf{d}^\top, 0)^\top$ and, as λ increases, the point $\mathbf{X}(\lambda)$ travels either along or past the camera's image plane, its projection onto the image plane is given by

$$\mathbf{x}(\lambda) \sim \mathtt{P}\mathbf{X}(\lambda) \sim \mathtt{P}\mathbf{A} + \lambda\mathtt{P}\mathbf{D} = \mathtt{P}\mathbf{A} + \lambda[\mathtt{M} \mid \mathbf{p}_4]\begin{pmatrix}\mathbf{d}\\ 0\end{pmatrix} \sim \mathbf{a} + \lambda\mathtt{M}\mathbf{d}. \tag{3.32}$$

The corresponding vanishing point $\mathbf{v} \in \mathbb{P}^2$ is obtained in the limit,

$$\mathbf{v} \sim \lim_{\lambda\to\infty} \mathbf{x}(\lambda) \sim \lim_{\lambda\to\infty} (\mathbf{a} + \lambda\mathtt{M}\mathbf{d}) \sim \lim_{\lambda\to\infty} \lambda\mathtt{M}\mathbf{d} \sim \mathtt{M}\mathbf{d}, \tag{3.33}$$

recalling that the homogeneous vector $\mathtt{M}\mathbf{d} \in \mathbb{P}^3$ is equivalent to the homogeneous vector $\lambda\mathtt{M}\mathbf{d} \in \mathbb{P}^3$, for any scalar $\lambda \neq 0$. The location of a vanishing point in the image plane is thus identical for all lines in space that share the same orientation, since it is only that orientation that plays any role given a fixed camera. By the central projection, the line through the camera center $\mathbf{C}$ and the vanishing point $\mathbf{v}$ on the image plane necessarily has that same orientation as well; consequently, the vanishing point corresponding to an orientation $\mathbf{D}$ is equivalently given by the intersection with the image plane of the unique ray through $\mathbf{C}$ with orientation $\mathbf{D}$ (cf. Figure 3.8).

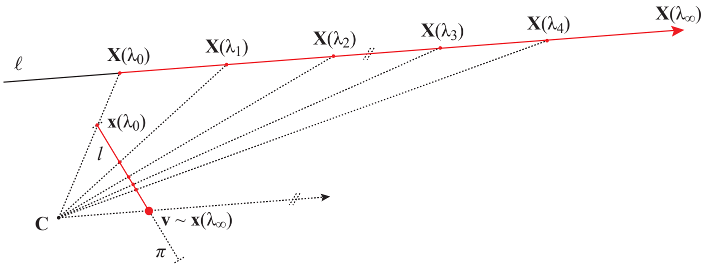


Figure 3.8: The projection of a line $\ell = \{\mathbf{X}(\lambda) \mid \lambda \in \mathbb{R}\}$ in 2-dimensional space to the line $l = \{\mathbf{x}(\lambda) \mid \lambda \in \mathbb{R}, \mathbf{x}(\lambda) \subset \pi\}$ in the 1-dimensional bounded image plane π. The vanishing point $\mathbf{v}$ is obtained at $\mathbf{x}(\lambda)$ as $\lambda \to \infty$; that same point $\mathbf{v}$ is obtained equivalently by intersecting the ray through the camera center $\mathbf{C}$ parallel to ℓ. The same holds for the projection of a line in 3-dimensional space to a 2-dimensional image plane.

Infinite Vanishing Points. Given a 3×4 camera projection matrix $\mathtt{P}$, points $(x, y, 0)^\top \sim \mathtt{P}\mathbf{X}$ arise from the projection of points $\mathbf{X} \in \mathbb{P}^3$ orthogonal to the third row of $\mathtt{P}$. Consequently, that third row of $\mathtt{P}$ is the normal vector of the *principal plane* through the origin of $\mathbb{R}^3$ parallel to the image plane π, since the infinite vanishing points of π are the projections of orientations parallel to π.

Vanishing Points in the Columns of $\mathtt{P}$. The first three columns of the projection matrix $\mathtt{P}$ of a finite projective camera are the respective vanishing points of the orientations in 3-dimensional space corresponding to the X-, Y- and Z-axes of the world coordinate frame. Let $\mathbf{p}_i$ indicate the i^{th} column of $\mathtt{P}$. To take an example of the X-axis, the orientation of the X-axis is given by $(1, 0, 0, 0)^\top \in \mathbb{P}^3$ and thus projects to $\mathbf{p}_1 \sim \mathtt{P}(1, 0, 0, 0)^\top$.

3.6 Vanishing Lines

The intersections at infinity $\mathbf{D} \sim (\mathbf{d}^\top, 0)^\top \in \mathbb{P}^3$ of a set of pairs of lines ℓ parallel in space project to corresponding vanishing points $\mathbf{v} \in \mathbb{P}^2$ incident with a single line $\mathbf{l} \in \mathbb{P}^2$ in the image plane, called a *vanishing line*, if and only if the orientations in space of all such lines ℓ are coplanar (cf. Figure 3.9). Since the mapping between planes by the central projection reduces to a planar projectivity $h : \mathbb{P}^2 \to \mathbb{P}^2$, and since projectivities preserve point-line incidence, a vanishing line is the projection onto the image plane of the vector in $\mathbb{P}^2$ corresponding to a vector in $\mathbb{R}^3$ normal to the plane through the center of projection that contains the totality of the said coplanar orientations. Accordingly, the vanishing line $\mathbf{l}$ of a plane is precisely the line at infinity $\mathbf{l}_\infty$ if and only if the projecting plane is parallel to the image plane. In either case, $\mathbf{l}$ specifies the orientation in world coordinates of the projecting plane.

Affine Planar Rectification. As an aside, let us consider that carrying out the rectification to within an affinity of the projectively distorted image of a plane reduces to identifying the plane's corresponding vanishing line $\mathbf{l}$. If the projecting plane contains at least two distinct pairs of parallel lines, we can compute $\mathbf{l}$ as the join of their two corresponding vanishing points. Once we have identified its vanishing line, we can remove the projective distortion in the plane's image by applying a projective warping specified by a planar projectivity h that maps $\mathbf{l} \sim (l_1, l_2, l_3)^\top$ to its *canonical position* $\mathbf{l}_\infty \sim (0, 0, 1)^\top$. This mapping is given by an invertible 3×3 matrix $\mathtt{H}$,

$$\mathtt{H} \sim \mathtt{H}_\text{A} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_1 & l_2 & l_3 \end{bmatrix}, \tag{3.34}$$

where $\mathtt{H}_\text{A}$ is any planar affinity, since

$$\mathtt{H}^{-\top}\mathbf{l} \sim \mathtt{H}_\text{A}^{-\top} \begin{bmatrix} l_3 & 0 & -l_1 \\ 0 & l_3 & -l_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} l_1 \\ l_2 \\ l_3 \end{pmatrix} \sim \mathtt{H}_\text{A}^{-\top} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \sim \mathbf{l}_\infty, \tag{3.35}$$

recalling that the line at infinity $\mathbf{l}_\infty$ is invariant under the affine transformations of the plane. We call the application of such a projective warping an *affine planar rectification*. Note that in order to rectify the image to within a non-zero scalar factor, we would need to carry out a *metric* planar rectification (cf. Hartley and Zisserman [15]).

3.7 Synopsis

Projective Geometry. Parallel lines in Euclidean geometry never meet in a point; in projective geometry they always do, albeit in points at infinity. The points at infinity $(x, y, 0)^\top \in \mathbb{P}^2$ represent the totality of orientations of the projective plane; the points at infinity $(\mathbf{d}^\top, 0)^\top = (\mathrm{X}, \mathrm{Y}, \mathrm{Z}, 0)^\top \in \mathbb{P}^3$ represent all of the orientations of projective 3-space. In the projective plane, a point $\mathbf{x} \in \mathbb{P}^2$ lies on a line $\mathbf{l} \in \mathbb{P}^2$ only if the vectors $\mathbf{x}, \mathbf{l}$ are incident, i.e., only if $\mathbf{x}^\top \mathbf{l} = 0$; in projective 3-space, we have incidence between points $\mathbf{X} \in \mathbb{P}^3$ and planes $\boldsymbol{\pi} \in \mathbb{P}^3$. In $\mathbb{P}^2$, the unique line $\mathbf{l}$ that joins the points $\mathbf{x}, \mathbf{x}'$ is given by $\mathbf{l} \sim \mathbf{x} \times \mathbf{x}'$; likewise, the point of intersection $\mathbf{x}$ of two lines $\mathbf{l}, \mathbf{l}'$ is given by $\mathbf{x} \sim \mathbf{l} \times \mathbf{l}'$. The vector in $\mathbb{P}^2$ that thus represents the intersection of two lines parallel in the image plane has the same orientation as those two parallel lines. All infinite points of the projective plane are incident with the line at infinity $\mathbf{l}_\infty \sim (0, 0, 1)^\top$. A projectivity $h : \mathbb{P}^2 \to \mathbb{P}^2$ in the projective plane is an invertible mapping that preserves point-line incidence, and we can represent any such h using an invertible 3×3 matrix $\mathtt{H}$. Unless the projectivity h is an affinity, h transforms $\mathbf{l}_\infty$ to a finite line $\mathbf{l}$. Since projectivities preserve point-line incidence, the infinite points are—with a single exception[9]—themselves transformed to finite points, and lines parallel in the plane are accordingly projected such that incidence with these transformed infinite points is preserved.

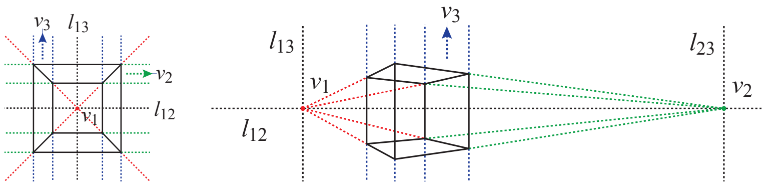
Image Formation. We model image formation using a finite projective camera, which projects points $\mathbf{X} \in \mathbb{P}^3$ in 3-dimensional space to points $\mathbf{x} \in \mathbb{P}^2$ in the 2-dimensional image plane fundamentally via the central projection. Under known camera geometry, we can accordingly determine to which point $\mathbf{x}$ in the image plane a point $\mathbf{X}$ in space projects. Moreover, we can back-project any point $\mathbf{x}$ in the image plane to the ray through the camera center that passes through all points $\mathbf{X}$ in space that project to $\mathbf{x}$. Back-projecting the projection onto the image plane of an infinite point $\mathbf{D} \sim (\mathbf{d}^\top, 0)^\top \in \mathbb{P}^3$ in space yields a ray with the selfsame orientation as the orientation the vector $\mathbf{D}$ itself represents.

Vanishing Points. Vanishing points arise on account of the nature of image formation. Every orientation in space projects to a corresponding vanishing point $\mathbf{v} \in \mathbb{P}^2$ in the image plane, albeit perhaps one at infinity. Projecting any line ℓ in space with orientation $\mathbf{D} \sim (\mathbf{d}^\top, 0)^\top \in \mathbb{P}^3$ onto the image plane and intersecting the image plane with the unique ray through the camera center $\mathbf{C}$ with orientation $\mathbf{D}$ equivalently yields the same vanishing point $\mathbf{v}$. The back-projection of $\mathbf{v}$ yields a ray whose orientation $\mathbf{D}$ is the same as that of the projecting line ℓ. Every point in the image plane is the vanishing point corresponding to a particular orientation in the scene.

Vanishing Lines. Every vanishing line uniquely determines the orientation of a plane in space, and every vanishing line is given by the join of two vanishing points corresponding to lines respectively parallel in that plane. A projecting plane is parallel to the image plane if and only if its corresponding vanishing line $\mathbf{l}$ is the line at infinity $\mathbf{l}_\infty$; only then are lines parallel in the projecting plane projected to lines parallel in the image plane. Affine rectification reduces to transforming $\mathbf{l}$ to $\mathbf{l}_\infty$. The vector $\mathbf{l}$ is the normal vector in $\mathbb{R}^3$ of the projecting plane.

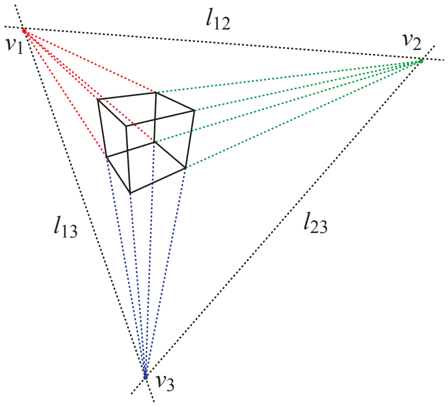
---

[9] We recall (from an earlier footnote) that every finite line $\mathbf{l} \sim (a, b, c)^\top$ is incident with the single infinite point $(b, -a, 0)^\top$.



(a) One finite vanishing point v_1, two at infinity v_2, v_3. The vanishing line l_{23} is at infinity, l_{12} and l_{13} are finite.

(b) Two finite vanishing points v_1, v_2, one at infinity v_3. The corresponding vanishing lines l_{12}, l_{13}, l_{23} are all finite.



(c) Three finite vanishing points v_1, v_2, v_3. The corresponding vanishing lines l_{12}, l_{13}, l_{23} are all finite.

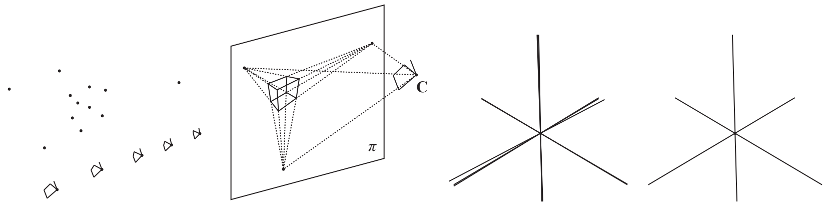
Figure 3.9: Vanishing points and vanishing lines for triplets of pairwise-orthogonal scene orientations, using the cube from Figure 3.1. We depict vanishing points at infinity in the customary manner, using an arrow that specifies a direction of the corresponding orientation. A line $l_{ij} = l_{ji}$ is the vanishing line shared by the vanishing points v_i, v_j. All the vanishing lines in the figure are finite, with the sole exception of the line l_{23} in (a), which corresponds to the line at infinity $\mathbf{l}_\infty$. Note that, for instance, the vanishing line l_{12} of (b) belongs to both the top and bottom planes of the cube and thus uniquely specifies the common orientation of both planes, and all planes with which they are parallel.

Chapter 4

Implementation

4.1 Processing Pipeline

Our system borrows in spirit most heavily from the multiple-view approach for extracting the dominant three pairwise-orthogonal orientations of a typical urban scene proposed in Sinha et al. [38] (cf. Appendix B). As we shall see, however, ours is a material refinement of their approach. We begin with the recovery of camera geometry for each view (cf. Irschara et al. [17]). Across the k available views, we then extract image line segments and compute a single constellation of two or three candidate vanishing points per view, constrained to satisfy an orthogonality criterion and refined with respect to candidate vanishing point inliers determined using an optimal distance measure. We then map the orientations corresponding to those candidate vanishing points to antipodal points on the unit sphere, given by corresponding unit direction vectors. We proceed to extract three pairwise-orthogonal orientations—which we expect to correspond closely with the dominant three pairwise-orthogonal orientations of the underlying urban scene—by fitting a tripod centered at the sphere's origin to those said points. We illustrate the processing pipeline of our approach in Figure 4.1.



(a) Recovery of camera geometry for the k available views.

(b) Extraction of a constellation of two or three candidate vanishing points in a single view, constrained to correspond closely to pairwise-orthogonal scene orientations.

(c) Antipodal unit direction vectors corresponding to the orientations computed from candidate vanishing points extracted across the k views.

(d) The pairwise-orthogonal orientations corresponding to the best-fit tripod.

Figure 4.1: The processing pipeline of our approach.

4.2 Extracting a Constellation in a Single View

Candidate Vanishing Points. Given a view of the scene that we have projectively warped in order to compensate for the effect of radial lens distortion (cf. Hartley and Zisserman [15]) and a set $\mathcal{S}$ of line segments s that we have extracted from that view, we compute candidate vanishing points from the intersections of the image lines $l \subset \mathbb{R}^2$ corresponding to the segments $s \in \mathcal{S}$. We begin by obtaining the homogeneous representation $\mathbf{l} \in \mathbb{P}^2$ of a line l in the image plane corresponding to an extracted image segment s by working out the vector product of the homogeneous endpoints $\mathbf{p}_1, \mathbf{p}_2 \in \mathbb{P}^2$ of s,

$$\mathbf{l} \sim \mathbf{p}_1 \times \mathbf{p}_2.$$

Given the homogenous vectors $\mathbf{l}, \mathbf{l}' \in \mathbb{P}^2$ that represent the two lines $l, l' \subset \mathbb{R}^2$, we compute the intersection of l, l' once again using the vector product,

$$\mathbf{v} \sim \mathbf{l} \times \mathbf{l}',$$

yielding the candidate vanishing point $\mathbf{v} \in \mathbb{P}^2$ corresponding to the segments s, s'. We then normalize $\mathbf{v} = (v_1, v_2, v_3)^\top$ such that $\mathbf{v} = (v_1, v_2, 0)^\top$ if the magnitude of v_3 is not much greater than the machine epsilon, and $\mathbf{v} = (v_1/v_3, v_2/v_3, 1)^\top$ otherwise.

Accumulation. In order to determine which line segments correspond to a given candidate vanishing point, Sinha et al. [38] make use of a distance function $d(\mathbf{v}, s) = \alpha \in [0, \pi/2]$ proposed in Rother [34]. This distance function delivers an angular measure of the 'closeness' of the line segment $s \in \mathcal{S}$ to the vanishing point $\mathbf{v} \in \mathbb{P}^2$, where a smaller angle indicates a better correspondence than a larger one (cf. Figure A.1 of Appendix A). Sinha et al. consider all line segments s for which $d(\mathbf{v}, s) < T_{\mathrm{Roth}}$ to form the set $\mathcal{S}_{\mathbf{v}} \subseteq \mathcal{S}$ of *inliers* of the candidate vanishing point $\mathbf{v}$.

Rother's distance measure, however, is not optimal; as justified in Pflugfelder [31], the error measure of Liebowitz [22] (cf. Figure 4.2) is the only true error measure between a line segment s and a vanishing point $\mathbf{v}$ available in the literature. Consequently, we appeal to the distance measure of Liebowitz rather than to that of Rother.

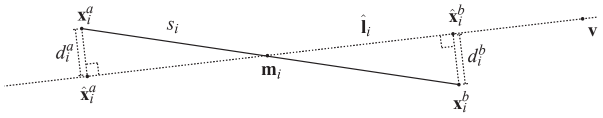


Figure 4.2: The line $\hat{\mathbf{l}}_i = \arg\min_{\mathbf{l}} \mathcal{F}^{(i)}(\mathbf{l})$ is the line through $\mathbf{v}$ that minimizes the error $\mathcal{F}^{(i)}(\mathbf{l}) = d_\perp^2(\mathbf{l}, \mathbf{x}_i^a) + d_\perp^2(\mathbf{l}, \mathbf{x}_i^b) = d_i^a \cdot d_i^a + d_i^b \cdot d_i^b$ with respect to the segment s_i, where $d_\perp^2(\mathbf{l}, \mathbf{x})$ gives the squared Euclidean distance in the plane between the point $\mathbf{x}$ and its projection to the line $\mathbf{l}$. The error $\mathcal{F}^{(i)}(\hat{\mathbf{l}}_i)$ gives the error measure of Liebowitz with respect to a segment s_i and a candidate vanishing point $\mathbf{v}$. Note that $\mathbf{m}_i$ is not necessarily the midpoint of the segment s_i.

Let the vectors $\mathbf{x}_i^a = (x_{i1}^a, x_{i2}^a, 1)^\top, \mathbf{x}_i^b = (x_{i1}^b, x_{i2}^b, 1)^\top \in \mathbb{P}^2$ represent the endpoints of the segment s_i, the vector $\mathbf{v} = (v_1, v_2, v_3)^\top \in \mathbb{P}^2$ represent a candidate vanishing point, and the vector $\mathbf{l} = (l_1, l_2, l_3)^\top \in \mathbb{P}^2$ represent a line that joins $\mathbf{v}$ with some $\mathbf{m}_i = k_i^a \mathbf{x}_i^a + k_i^b \mathbf{x}_i^b \in \mathbb{P}^2$. Given a segment s_i and a line $\mathbf{l}$ through a fixed candidate vanishing point $\mathbf{v}$, we declare that the line's error[1] $\mathcal{F}^{(i)}(\mathbf{l})$ with respect to the segment s_i is given by

$$\begin{aligned}\mathcal{F}^{(i)}(\mathbf{l}) &= d_\perp^2(\mathbf{l}, \mathbf{x}_i^a) + d_\perp^2(\mathbf{l}, \mathbf{x}_i^b) \\ &= d_i^a \cdot d_i^a + d_i^b \cdot d_i^b \\ &= \frac{(\mathbf{x}_i^{a\top}\mathbf{l})^2 + (\mathbf{x}_i^{b\top}\mathbf{l})^2}{l_1^2 + l_2^2},\end{aligned} \tag{4.1}$$

where $d_\perp^2(\mathbf{l}, \mathbf{x})$ gives the squared Euclidean distance between the point $\mathbf{x}$ and its projection to the line $\mathbf{l}$. The line $\hat{\mathbf{l}}_i = \arg\min_{\mathbf{l}} \mathcal{F}^{(i)}(\mathbf{l})$ through $\mathbf{v}$ that minimizes the error with respect to s_i is thus the line

$$\begin{aligned}\hat{\mathbf{l}}_i &= \mathbf{v} \times \mathbf{m}_i \\ &= \mathbf{v} \times (k_i^a \mathbf{x}_i^a + k_i^b \mathbf{x}_i^b) \\ &= \mathbf{v} \times ((\mathbf{x}_i^a + \mathbf{x}_i^b)^\top \mathbf{k}_i),\end{aligned} \tag{4.2}$$

where the vector $\mathbf{k}_i$ specifies the linear combination $\mathbf{m}_i$ of the endpoints $\mathbf{x}_i^a, \mathbf{x}_i^b$ of the segment s_i for which $\mathcal{F}^{(i)}(\hat{\mathbf{l}}_i)$ is minimized. It is this minimized error measure $\mathcal{F}^{(i)}(\hat{\mathbf{l}}_i)$ that we term the distance measure of Liebowitz between a segment s_i and a vanishing point $\mathbf{v}$. Substituting $\hat{\mathbf{l}}_i = \mathbf{v} \times [\mathbf{x}_i^a + \mathbf{x}_i^b]\mathbf{k}_i$ from Equation (4.2) for $\mathbf{l}$ in Equation (4.1), we obtain[2]

$$\mathcal{F}^{(i)}(\mathbf{v} \times ((\mathbf{x}_i^a + \mathbf{x}_i^b)^\top \mathbf{k}_i)) = \frac{\mathbf{k}_i{}^\top \mathbf{k}_i}{\mathbf{k}_i{}^\top \mathtt{A} \mathbf{k}_i}, \tag{4.3}$$

which is minimized when $\mathbf{k}_i$ is the unit eigenvector of $\mathtt{A}$ corresponding to the largest eigenvalue $\lambda_{\max}$ of $\mathtt{A}$, since it follows that $\mathbf{k}_i{}^\top \mathbf{k}_i = 1$ and $\mathbf{k}_i{}^\top \mathtt{A} \mathbf{k}_i = \lambda_{\max} \mathbf{k}_i{}^\top \mathbf{k}_i = \lambda_{\max}$. The matrix $\mathtt{A}$ is given by

$$\mathtt{A} = \frac{1}{\mu} \begin{bmatrix} A_{11} & A_{12} \\ A_{12} & A_{22} \end{bmatrix}, \tag{4.4}$$

where

$$\begin{aligned}\mu &= 2(x_{i2}^b v_1 - x_{i1}^b v_2 - v_1 x_{i2}^a + x_{i1}^b v_3 x_{i2}^a + v_2 x_{i1}^a - x_{i2}^b v_3 x_{i1}^a), \\ A_{11} &= (-x_{i2}^a v_3 + v_2)^2 + (-v_1 + v_3 x_{i1}^a)^2, \\ A_{12} &= (-x_{i2}^a v_3 + v_2)(v_2 - x_{i2}^b v_3) + (-v_1 + v_3 x_{i1}^a)(x_{i1}^b v_3 - v_1), \\ A_{22} &= (v_2 - x_{i2}^b v_3)^2 + (x_{i1}^b v_3 - v_1)^2.\end{aligned}$$

Finally, we plug $\mathbf{k}$ back into the right-hand side of Equation (4.2) to obtain the sought optimal line $\hat{\mathbf{l}}_i$ through $\mathbf{v}$ corresponding to the segment s_i. The corresponding error is $\mathcal{F}^{(i)}(\hat{\mathbf{l}}_i) = \mathcal{F}_{\min}^{(i)} = 1/\lambda_{\max}$. Note that we can also obtain this error $\mathcal{F}_{\min}^{(i)}$ by taking the larger roots of the characteristic polynomial of the matrix $\mathtt{A}$,

$$\mathcal{F}_{\min}^{(i)} = \frac{\mu}{A_{11} + A_{22} + \sqrt{(A_{11} - A_{22})^2 + 4A_{12}^2}}, \tag{4.5}$$

---

[1] We denote the Liebowitz error $\mathcal{F}$ in calligraphic script in order to be consistent with the notation of Liebowitz. Elsewhere, however, we use letters in calligraphic script strictly in order to denote sets.

[2] See Liebowitz [22] for the complete derivation of this step, which includes with it the derivation of the matrix $\mathtt{A}$ as well.

which is computationally less expensive to work out than an eigenvalue decomposition. It is the error $\mathcal{F}^{(i)}_{\min}$ that we use in our grouping of segments with respect to a candidate vanishing point rather than Rother's distance function $d(\mathbf{v}, s)$; accordingly, given a candidate vanishing point $\mathbf{v}$, we consider each segment s_i for which $\mathcal{F}^{(i)}_{\min} < T_{\text{Lieb}}$ to be an inlier of $\mathbf{v}$. See Liebowitz [22] for a more detailed treatment of how to compute the line $\hat{\mathbf{l}}_i$ through a vanishing point $\mathbf{v}$ corresponding to a segment s_i.

Optimal Intersection Estimation. Once Sinha et al. have grouped inlier segments $s \in \mathcal{S}_{\mathbf{v}}$ with a given candidate vanishing point $\mathbf{v} \in \mathbb{P}^2$, they carry out no supplementary re-estimation of that candidate vanishing point with respect to its inliers in $\mathcal{S}_{\mathbf{v}}$. In contrast, we wish to compute an optimal point of intersection corresponding to the segments determined to be inliers of a candidate vanishing point. With respect to point-line incidence (cf. Section 3.2.1 in Chapter 3), the *ideal* point of intersection for a set of lines $\mathbf{l} \in \mathbb{P}^2$ would be given by the vector $\mathbf{v}^* \in \mathbb{P}^2$ that is orthogonal to each vector $\mathbf{l}$. Since we compute our lines $\mathbf{l}$ from segments $s \in \mathcal{S}$ extracted from a quantized and inherently noisy image, an ideal vector $\mathbf{v}^*$ will in practice—except by fluke—never exist. Given a set of n lines $\mathbf{l}_i \in \mathbb{P}^2$, the least-squares point of intersection with respect to point-line incidence is given by the vector $\hat{\mathbf{v}}_{\text{SVD}} \in \mathbb{P}^2$ that minimizes the quantity

$$\left\| \begin{bmatrix} \mathbf{l}_1 & \cdots & \mathbf{l}_n \end{bmatrix}^\top \hat{\mathbf{v}}_{\text{SVD}} \right\|^2, \tag{4.6}$$

where each vector $\mathbf{l}_i$ is scaled to unit length (cf. Cipolla and Boyer [7]). This minimizing vector $\hat{\mathbf{v}}_{\text{SVD}}$ is precisely the vector corresponding to the smallest singular value of the singular value decomposition (SVD) of the $n \times 3$ matrix $\begin{bmatrix} \mathbf{l}_1 & \cdots & \mathbf{l}_n \end{bmatrix}^\top$ (cf. Appendix D). Note that computing the vector $\hat{\mathbf{v}}_{\text{SVD}}$ amounts to fitting a great circle through the set of points $\mathbf{l}_i$ lying on the unit sphere.

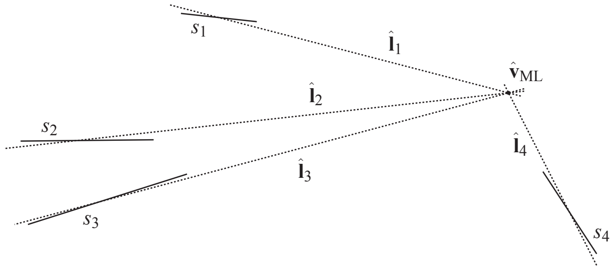


Figure 4.3: Maximum likelihood intersection estimation. The point $\hat{\mathbf{v}}_{\text{ML}}$ is the point that minimizes the Liebowitz error $\sum_{s_i \in \mathcal{S}_{\mathbf{v}}} d^2_\perp(\hat{\mathbf{l}}_i, \mathbf{x}^a_i) + d^2_\perp(\hat{\mathbf{l}}_i, \mathbf{x}^b_i)$.

We may proceed to even further refine the result $\hat{\mathbf{v}}_{\text{SVD}}$ we have thus obtained. A maximum likelihood (ML) estimate of the corresponding vanishing point over all segments s_i is given by the vector $\hat{\mathbf{v}}_{\text{ML}} = \arg\min_{\mathbf{v}} \text{cost}(\mathbf{v})$, where

$$\text{cost}(\mathbf{v}) = \sum_{s_i \in \mathcal{S}_{\mathbf{v}}} d^2_\perp(\hat{\mathbf{l}}_i, \mathbf{x}^a_i) + d^2_\perp(\hat{\mathbf{l}}_i, \mathbf{x}^b_i) = \sum_{s_i \in \mathcal{S}_{\mathbf{v}}} \mathcal{F}^{(i)}_{\min}. \tag{4.7}$$

Since we know how to compute the Liebowitz error $\mathcal{F}^{(i)}_{\min}$ with respect to each segment s_i given any candidate vanishing point $\mathbf{v}$, we have what we need to minimize $\text{cost}(\cdot)$

over different values of $\mathbf{v}$ using the Levenberg-Marquardt non-linear least squares optimization technique (cf. Lourakis [25]). We initialize the solver[3] with the estimate $\hat{\mathbf{v}}_{\mathrm{SVD}}$ obtained by means of the aforementioned SVD approach. For a more detailed treatment of this ML intersection estimation technique, we refer the reader once again to Liebowitz [22].

Orthogonality Criterion. For a pair of candidate vanishing points $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{P}^2$, our criterion requires that the unit direction vectors $\mathbf{d}_1, \mathbf{d}_2 \in \mathbb{R}^3$ corresponding to their back-projections be within a tight threshold of orthogonality; i.e., $|\mathbf{d}_1^\top \mathbf{d}_2| < T_{\mathrm{ortho}}$. For a triplet $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3 \in \mathbb{P}^2$, we check each pair $\mathbf{d}_i, \mathbf{d}_j, i \neq j$, of corresponding back-projections for orthogonality in the same manner.

Sinha et al. do not enforce orthogonality in the orientations corresponding to the candidate vanishing points extracted in any single view, assuming instead that enough of the orientations they extract across the k available views will correspond to the scene's dominant three pairwise-orthogonal orientations. Since we seek a solution that corresponds as closely as possible to the dominant three pairwise-orthogonal orientations of the scene, however, we choose to enforce orthogonality already in the orientations extracted from each view.

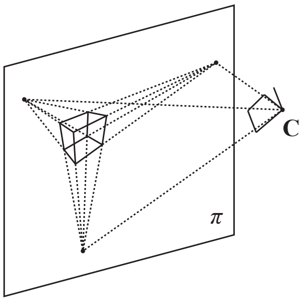


Figure 4.4: Our orthogonality criterion. We constrain the constellation of two or three vanishing points extracted from any single view to back-project to rays that are pairwise within a threshold T_{ortho} of orthogonality.

A Constellation's Vote. We assign a vote to each constellation, intended to reflect its relative 'goodness' vis-à-vis the segments in $\mathcal{S}$. Given a constellation $\mathcal{C}$ of two or three candidate vanishing points, its vote is given by

$$\mathrm{vote}(\mathcal{C}) = \sum_{\mathbf{v} \in \mathcal{C}} \sum_{s_i \in \mathcal{S}_{\mathbf{v}}} 1 - \frac{\mathcal{F}_{\mathrm{min}}^{(i)}}{T_{\mathrm{Lieb}}}, \tag{4.8}$$

where $\mathcal{F}_{\mathrm{min}}^{(i)}$ is, once again, the error of the optimal line $\hat{\mathbf{l}}_i$ through the candidate vanishing point $\mathbf{v}$ with respect to the segment s_i; the set $\mathcal{S}_{\mathbf{v}}$ contains all inlier segments s_i of $\mathbf{v}$, such that as before, each $\mathcal{F}_{\mathrm{min}}^{(i)}$ constrained to be smaller than the threshold T_{Lieb}. Note that $1 - \mathcal{F}_{\mathrm{min}}^{(i)}/T_{\mathrm{Lieb}} = 1$—and is thus maximized—for a segment $s_i \in \mathcal{S}_{\mathbf{v}}$ if and only if its Liebowitz error $\mathcal{F}_{\mathrm{min}}^{(i)}$ with respect to $\mathbf{v}$ is naught.

---

[3] See `http://www.ics.forth.gr/~lourakis/levmar/` to obtain `levmar`, the implementation of the Levenberg-Marquardt non-linear least squares solver that we used to minimize $\mathrm{cost}(\cdot)$.

Pseudocode. For each of the k views of the scene, we extract a constellation $\mathcal{C}$ of two or three vanishing points corresponding ideally to the dominant three pairwise-orthogonal orientations of the scene. Our approach is inspired by RANSAC (cf. Appendix C) and is an adaptation of one presented in Rother [35]. We provide the pseudocode of our approach in Algorithm 1.

Algorithm 1 Extracting a Constellation of Vanishing Points in a Single View

```
for $N$ iterations do
    take 6 distinct image line segments at random from $\mathcal{S}$ and compute the candidate vanishing points $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$
    for all 4 constellations $\mathcal{C} \in \{\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}, \{\mathbf{v}_1, \mathbf{v}_2\}, \{\mathbf{v}_1, \mathbf{v}_3\}, \{\mathbf{v}_2, \mathbf{v}_3\}\}$ do
        $\mathrm{vote}_{\mathcal{C}} \leftarrow \mathrm{vote}(\mathcal{C})$ {the support of the constellation $\mathcal{C}$}
        if $|\mathcal{C}| = 3$ yet the constellation with the greatest vote thus far encountered contains only a pair of candidate vanishing points, and the constellation $\mathcal{C}$ satisfies the orthogonality criterion then
            store $\mathcal{C}$ as the constellation with best support
        else if $\mathrm{vote}_{\mathcal{C}}$ is the greatest constellation vote thus far encountered and the constellation $\mathcal{C}$ satisfies the orthogonality criterion then
            store $\mathcal{C}$ as the constellation with best support
        end if
    end for
end for
return the re-estimation of each candidate vanishing point in the constellation with best support
```

4.3 Optimizing across k Views of a Scene

A constellation extracted using Algorithm 1 from any one view does not for all input necessarily correspond to the scene's *dominant* three pairwise-orthogonal orientations, owing in part to the fact that a competing constellation might happen to genuinely have better support in a particular view, and in part to the fact that Algorithm 1 involves choosing from constellations selected at random. Moreover, since we compute those orientations from a re-estimation of each candidate vanishing point in a best-support constellation $\mathcal{C}$, and since prior to re-estimation the corresponding orientations are themselves constrained to be pairwise-orthogonal to only within a threshold T_{ortho}, the orientations extracted using Algorithm 1 will in general fall short of being exactly pairwise-orthogonal. We accordingly seek to obtain a result that takes into account the information extracted from across the k available views and that yields a triplet of genuinely pairwise-orthogonal orientations that are as close as possible to the dominant three pairwise-orthogonal orientations of the scene.

A vanishing point back-projects to a ray through the view's camera center $\mathbf{C}$ whose *direction*, if given by a unit vector, can be either of an antipodal pair of vectors; to which of the pair of antipodal unit vectors that direction corresponds depends on the camera's pose with respect to the back-projection's orientation. Let the set $\mathcal{T}$—which we call a *tripod*—contain three orthonormal vectors $\mathbf{t} \in \mathbb{R}^3$. Let the set $\mathcal{K}$ contain the k contellations $\mathcal{C}$ of two or three candidate vanishing points extracted across k available views. Let $\mathcal{X}$ be the set of antipodal pairs of unit vectors corresponding to the back-projection of each vanishing point from the union of the k contellations $\mathcal{C} \in \mathcal{K}$. We proceed by fitting a tripod $\mathcal{T}$ to the antipodal unit direction vectors in $\mathcal{X}$ by iteratively rotating the tripod $\mathcal{T}$ with respect to the vectors in $\mathcal{X}$ close to the tripod's axes. We carry out this fitting, initialized with a tripod $\mathcal{T}$ corresponding to the back-projection of the candidate vanishing points in each of the k constellations $\mathcal{K}$; we then choose the resulting fitted tripod with the highest support as the basis for the winning set of three dominant pairwise-orthogonal scene orientations.

An Iteration of Tripod Fitting. Given, without loss of generality, a vector $\mathbf{t}_1 \in \mathcal{T} = \{\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3\}$ and the set $\mathcal{X}_1 \subset \mathcal{X}$ of the unit vectors in $\mathcal{X}$ within an angle T_{axis} of $\mathbf{t}_1$, the mean unit vector $\boldsymbol{\mu}_1$ of the vectors in $\mathcal{X}_1$ is given simply by the normalized sum of all $\mathbf{x} \in \mathcal{X}_1$,

$$\boldsymbol{\mu}_1 = \sum_{\mathbf{x}\in\mathcal{X}_1} \mathbf{x} \Bigg/ \Bigg\| \sum_{\mathbf{x}\in\mathcal{X}_1} \mathbf{x} \Bigg\| . \tag{4.9}$$

Let the matrix $\mathtt{R}_1$—which we can obtain ultimately by means of a corresponding unit quaternion—be the matrix that rotates the vector $\mathbf{t}_1$ into the vector $\boldsymbol{\mu}_1$. We treat the denominator of the right-hand side of (4.9) as a measure of confidence ω_1 in the rotation given by $\mathtt{R}_1$, the magnitude of which depends on the cardinality of $\mathcal{X}_1$ and on the extent to which the vectors $\mathbf{x} \in \mathcal{X}_1$ are clustered together. Having also computed the rotation matrices $\mathtt{R}_2, \mathtt{R}_3$ and weights ω_2, ω_3 corresponding, respectively, to the axes $\mathbf{t}_2, \mathbf{t}_3 \in \mathcal{T}$, an axis $\mathbf{t} \in \mathcal{T}$ rotates to $\mathbf{t}'$ by our tripod fitting technique according to

$$\begin{aligned}
\mathbf{t}' &= \frac{\omega_1\mathtt{R}_1\mathbf{t} + \omega_2\mathtt{R}_2\mathbf{t} + \omega_3\mathtt{R}_3\mathbf{t}}{\|\omega_1\mathtt{R}_1\mathbf{t} + \omega_2\mathtt{R}_2\mathbf{t} + \omega_3\mathtt{R}_3\mathbf{t}\|} \\
&= \frac{(\omega_1\mathtt{R}_1 + \omega_2\mathtt{R}_2 + \omega_3\mathtt{R}_3)\mathbf{t}}{\|(\omega_1\mathtt{R}_1 + \omega_2\mathtt{R}_2 + \omega_3\mathtt{R}_3)\mathbf{t}\|} \\
&= \frac{\mathtt{A}\mathbf{t}}{\|\mathtt{A}\mathbf{t}\|} = \mathtt{R}\mathbf{t}.
\end{aligned} \tag{4.10}$$

In order to express the transformation in (4.10) as a single matrix irrespective of $\mathbf{t}$, we seek the orthogonal[4] matrix $\mathtt{R}$ for which $\mathtt{R}\mathbf{t}$ gives $\mathbf{t}'$. By the SVD, we can decompose the matrix $\mathtt{A}$ such that $\mathtt{A} = \mathtt{U}\Sigma\mathtt{V}^\top$, where $\mathtt{U}, \mathtt{V}^\top$ are orthogonal matrices and Σ is a diagonal matrix; the orthogonal matrix closest in a least-squares sense to the matrix $\mathtt{A}$ is $\hat{\mathtt{R}} = \mathtt{U}\mathtt{V}^\top$ (cf. Appendix D). For a single iteration of our tripod fitting algorithm, the tripod $\mathcal{T}$ thus rotates to $\mathcal{T}'$ according to

$$\mathcal{T}' = \bigcup_{\mathbf{t}\in\mathcal{T}} \{\hat{\mathtt{R}}\mathbf{t}\} \tag{4.11}$$

Initialization. We run our fitting algorithm k times, once for a tripod corresponding to the back-projections of the candidate vanishing points in each of the k available constellations $\mathcal{C} \in \mathcal{K}$. If a constellation $\mathcal{C}$ contains only a pair of candidate vanishing points, we compute the third axis of the corresponding tripod $\mathcal{T}$ from the vector product of its first two. Since we demand that our final tripod have pairwise-orthogonal axes, we orthogonalize every tripod $\mathcal{T}$ that we use to initialize our tripod fitting algorithm. This reduces to orthogonalizing the matrix $\mathtt{T} = \begin{bmatrix}\mathbf{t}_1 & \mathbf{t}_2 & \mathbf{t}_3\end{bmatrix}$ in the same manner as presented above; i.e., $\mathtt{T} = \mathtt{U}\Sigma\mathtt{V}^\top$, and so $\hat{\mathtt{T}} = \mathtt{U}\mathtt{V}^\top = \begin{bmatrix}\hat{\mathbf{t}}_1 & \hat{\mathbf{t}}_2 & \hat{\mathbf{t}}_3\end{bmatrix}$.

---

[4] We recall that if a matrix $\mathtt{R}$ is orthogonal, then $\mathtt{R}\mathbf{t} = \mathtt{R}\mathbf{t}/\|\mathtt{R}\mathbf{t}\|$; i.e., it is a rotation matrix.

Support. From among k runs of our tripod fitting algorithm—each run initialized with a parwise-orthogonal tripod corresponding to one of the k views—we choose our best-fit tripod from among the k outcomes based on cosine similarity (cf. Banerjee et al. [2]). For each of the k outcome tripods $\mathcal{T}$, we compute

$$\gamma_{\mathcal{T}} = \sum_{\mathbf{t} \in \mathcal{T}} \sum_{\mathbf{x} \in \mathcal{X}_{\mathbf{t}}} \cos(\mathbf{x}^\top \mathbf{t}), \tag{4.12}$$

which expresses the aggregate cosine similarity between each tripod axis $\mathbf{t} \in \mathcal{T}$ and every vector $\mathbf{x} \in \mathcal{X}_{\mathbf{t}}$, and is thus[5] a measure of the tripod's support. We accordingly choose the tripod with best support as our best-fit tripod.

Pseudocode. We obtain a best-fit tripod with respect to $\mathcal{X}$ as the final result with best overall support from among k runs of an iterative fitting procedure, with each run distinctly initialized with a tripod corresponding to one of the k available constellations $\mathcal{C} \in \mathcal{K}$. The result with best support is the tripod $\mathcal{T}$ which, within N iterations of initialization, yields the highest weight $\gamma_{\mathcal{T}}$. We present the pseudocode of our approach in Algorithm 2.

Algorithm 2 Fitting a Tripod with Pairwise-Orthogonal Axes across k Views

```
$\mathcal{K} \leftarrow$ the set of $k$ constellations $\mathcal{C}$ obtained across $k$ views using Algorithm 1
$\mathcal{X} \leftarrow$ the set of antipodal unit vectors corresponding to the back-projection of each candidate vanishing point contained across all $k$ constellations in $\mathcal{K}$
for all $k$ constellations $\mathcal{C} \in \mathcal{K}$ do
    $\mathcal{T} \leftarrow$ the set of vectors corresponding to the back-projections of the pair or triplet of candidate vanishing points in the constellation $\mathcal{C}$
    if the set $\mathcal{T}$ contains only a pair of vectors then
        $\mathcal{T} \leftarrow \mathcal{T} \cup \{\mathbf{t}_1 \times \mathbf{t}_2\}$, where $\mathbf{t}_1, \mathbf{t}_2 \in \mathcal{T}$
    end if
    $\mathcal{T} \leftarrow \mathrm{orthogonalize}(\mathcal{T})$ {the tripod initialization}
    for $N$ iterations or until change is below a threshold $T_\epsilon$ do
        for all 3 pairwise-orthogonal tripod axes $\mathbf{t} \in \mathcal{T}$ do
            $\mathcal{X}_{\mathbf{t}} \leftarrow$ all $\mathbf{x} \in \mathcal{X}$ such that $\cos^{-1}(\mathbf{x}^\top \mathbf{t}) < T_{\mathrm{axis}}$
            $\omega_{\mathbf{t}} \leftarrow \| \sum_{\mathbf{x} \in \mathcal{X}_{\mathbf{t}}} \mathbf{x} \|$
            $\boldsymbol{\mu}_{\mathbf{t}} \leftarrow \sum_{\mathbf{x} \in \mathcal{X}_{\mathbf{t}}} \mathbf{x} / \omega_{\mathbf{t}}$
            $\mathtt{R}_{\mathbf{t}} \leftarrow$ the matrix that rotates $\mathbf{t}$ into $\boldsymbol{\mu}_{\mathbf{t}}$
        end for
        $\mathtt{A} \leftarrow \sum_{\mathbf{t} \in \mathcal{T}} \omega_{\mathbf{t}} \mathtt{R}_{\mathbf{t}}$
        $\hat{\mathtt{R}} \leftarrow \mathrm{orthogonalize}(\mathtt{A})$
        $\mathcal{T} \leftarrow \bigcup_{\mathbf{t} \in \mathcal{T}} \{\hat{\mathtt{R}}\mathbf{t}\}$
    end for
    $\gamma_{\mathcal{T}} \leftarrow \sum_{\mathbf{t} \in \mathcal{T}} \sum_{\mathbf{x} \in \mathcal{X}_{\mathbf{t}}} \cos(\mathbf{x}^\top \mathbf{t})$ {the support of the tripod $\mathcal{T}$}
end for
return the tripod with best support
```

---

[5] We recall that $0 = \arg\max_\theta \cos(\theta), \cos(0) = 1$, with $\cos(0) > \cos(\theta)$ for all $0 > \theta \geq \pi$.

Chapter 5

Evaluation

> *Student: Herr Professor, aber die Daten stimmen mit Ihrer Theorie nicht überein…*
> *Professor: Das ist aber schlecht für die Daten!*
>
> —overheard[1] from a colleague at VRVis

Following a brief note on computational complexity, we proceed to examine our algorithm's performance. We examine that performance by considering three data sets—`acv`, `ares` and `techgate`—corresponding to real-world urban scenes at Vienna's Donau City development, incidentally home to VRVis. We first demonstrate the outcome of a run of our algorithm on each of our three data sets by identifying the respective inlier segments of the vanishing points corresponding to the projection per view of the extracted pairwise-orthogonal scene orientations (cf. Figures 5.1, 5.2 and 5.3). We then provide a depiction of the antipodal directions extracted across all views of each data set, and with them the corresponding best-fit tripods (cf. Figure 5.4). In order to satisfy ourselves that the tripod fitting algorithm yields plausible results, we view a result thus obtained from a handful of different poses (cf. Figure 5.5). We compare these with the antipodal directions extracted via the approach of Sinha et al. [38], numbering—as in their paper—eight per view; to these, we likewise fit a tripod in our manner, since Sinha et al. omit a description of how exactly they choose their three pairwise-orthogonal scene orientations (cf. Appendix B). Note that in each case, we rendered the best-fit tripod (in red) more easily visible by superimposing vector graphics—drawn by hand—over the best-fit tripod (also in red) present in the respective screenshot. Finally, we show graphs of relative inlier counts (cf. Figures 5.6, 5.7 and 5.8) and error measures (cf. Figures 5.9, 5.10 and 5.11)—once for each of our three data sets—for three runs each of our algorithm and our adaptation of the approach of Sinha et al. Note also that all parameters were kept the same across all runs and for each data set, and that the vanishing point re-estimation approach we used is the SVD-based technique from Section 4.2 of Chapter 4.

5.1 Remarks on Complexity

The running time bottleneck of our algorithm—certainly if camera geometry is recovered in a pre-processing step—lies at the extraction of candidate vanishing points cor-

---

[1] (and uttered in jest, of course)

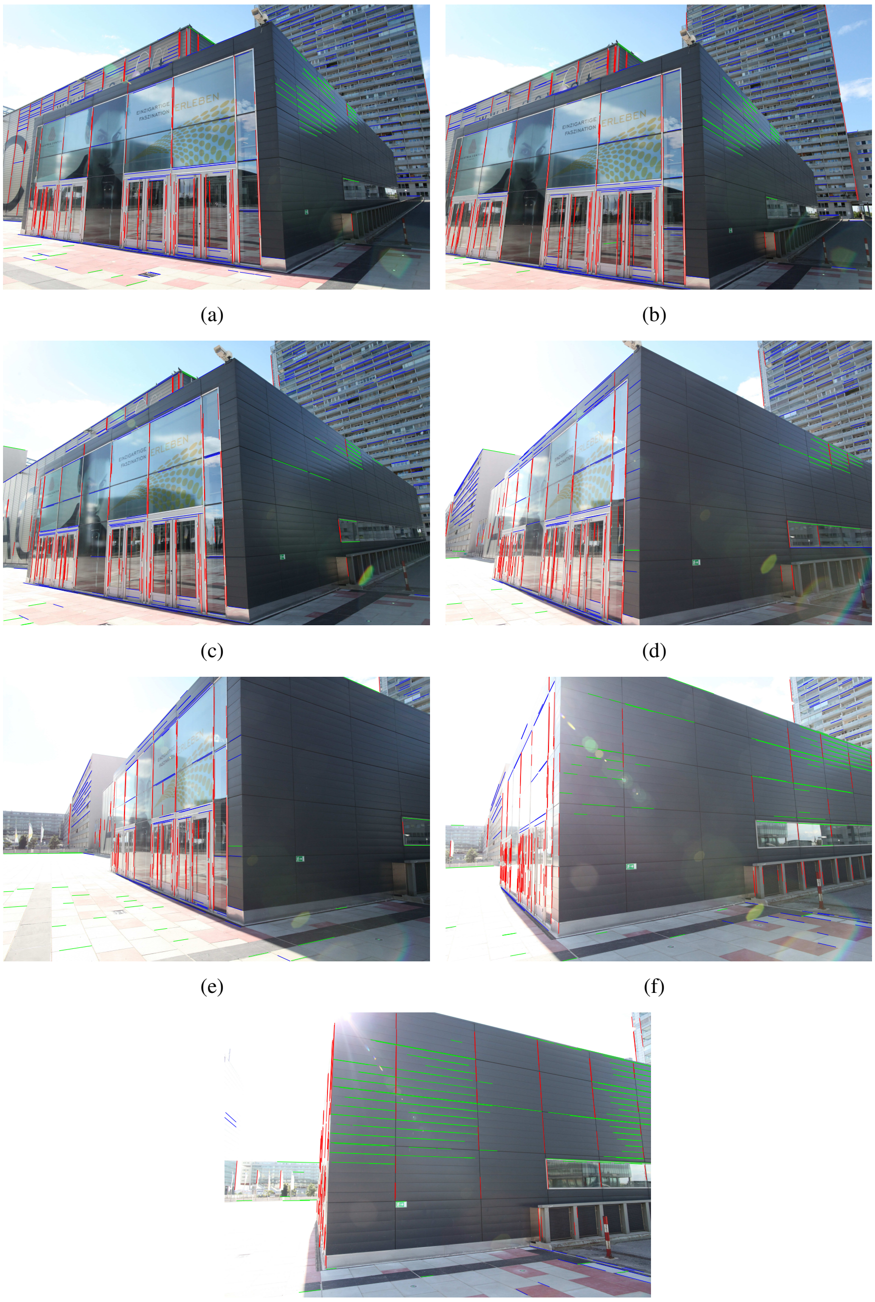
responding to pairwise-orthogonal scene orientations (cf. Algorithm 1). Given n line segments extracted in one view, there exist a total of $\binom{n}{2} = n(n-1)/2 \in O(n^2)$ candidate vanishing points from among which to choose. The number of ways to choose three distinct candidate vanishing points from among the total is precisely

$$\binom{\binom{n}{2}}{3} = \frac{1}{3!} \cdot \binom{n}{2} \cdot \left(\binom{n}{2} - 1\right) \cdot \left(\binom{n}{2} - 2\right), \tag{5.1}$$

since we have $\binom{n}{2}$ candidates available for our first point, $\binom{n}{2} - 1$ for our second, and $\binom{n}{2} - 2$ for our third, and there are 3! ways of ordering those three points. Accordingly, the complexity of an enumeration—carried out in order to determine which constellation has best support—of just each unique triplet (recall that in Algorithm 1, we also consider pairs!) of distinct candidate vanishing points *in a single view* is $O(n^6)$ in the number of line segments extracted in that view, repeated for each view. It is in order to vie with this crippling complexity that we opt instead to obtain our best-support result from among a (potentially) much smaller number N of constellations, obtained from pairs of segments chosen at random from among the available n. In all of our experiments, we set that number to $N = 1000$.

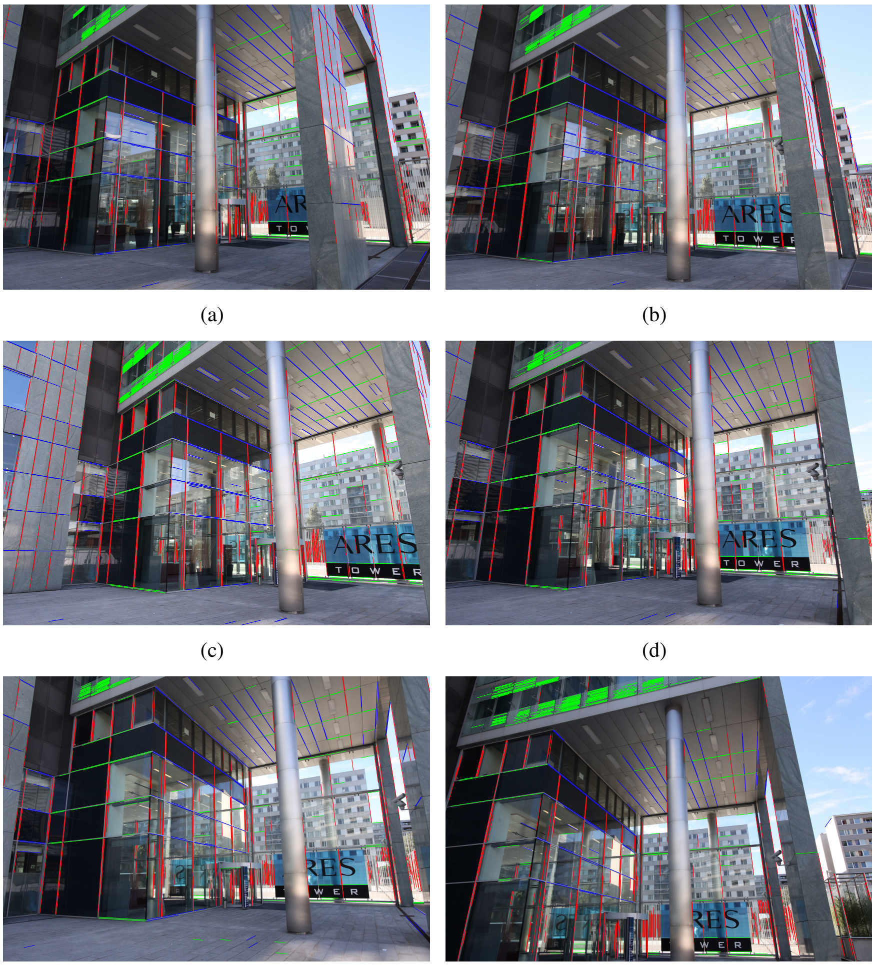
5.2 Results

Tallying counts of inlier segments per vanishing point or the corresponding cumulative error values relative to inlier tallies in an image does not in general and by itself yield a meaningful measure of the performance of an algorithm for the extraction of the vanishing points corresponding to the underlying scene's dominant pairwise-orthogonal orientations. Our data sets, however, are of the sort that most line segments do in fact correspond to one such vanishing point; accordingly, we expect that only a minority of segments should wind up unclassified with respect to the dominant three pairwise-orthogonal scene orientations our algorithm extracts. Unrelatedly, we expect our algorithm to be stable; in this regard, we expect that the aforementioned inlier proportions and normalized cumulative error for the inliers of each vanishing point remain consistent across runs. In our experiments, the results for our approach support the contention that our algorithm satisfies both of these criteria; in contrast, the approach of Sinha et al. yielded results that are of a comparatively poorer quality, and that were less consistent across runs.



(a) (b) (c) (d) (e) (f) (g)

Figure 5.1: The `acv` data set with an approximation of its dominant three pairwise-orthogonal scene orientations extracted using our approach, with the inlier segments of their corresponding vanishing points shown per view in red, green and blue, respectively.

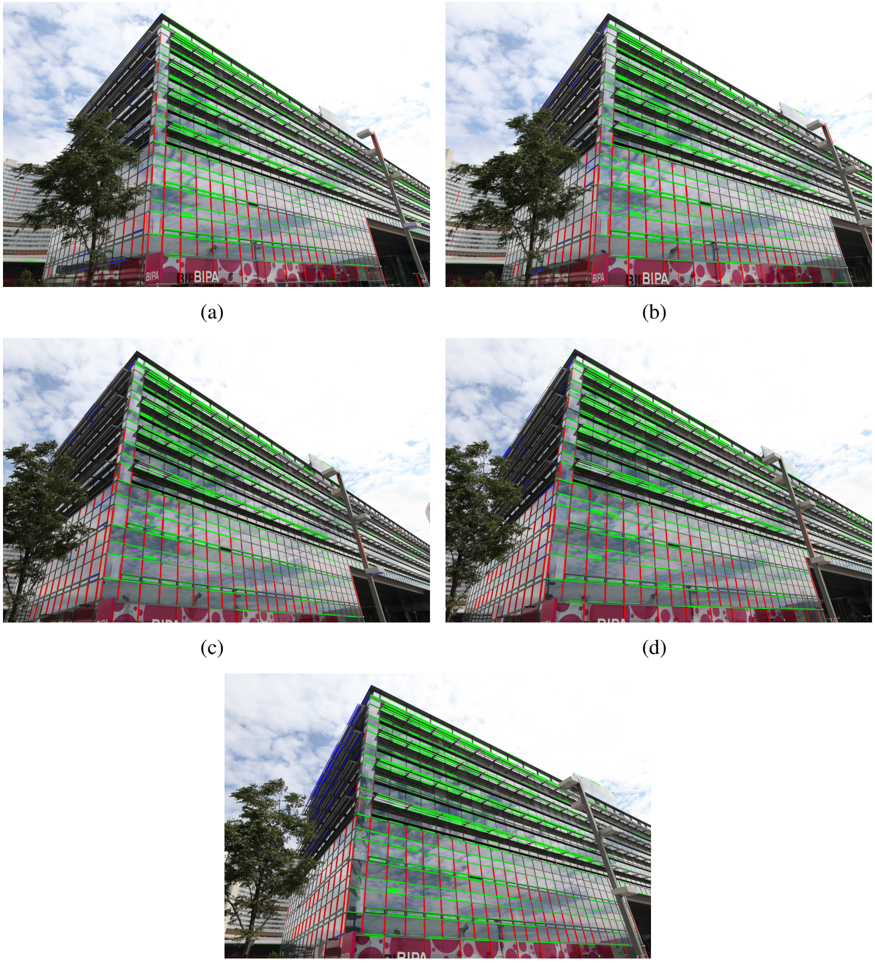


(a) (b)

(c) (d)

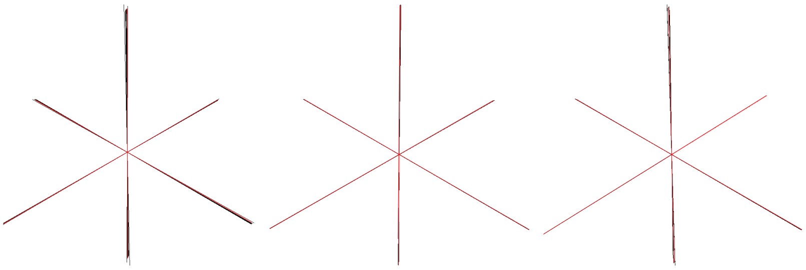
(e) (f)

Figure 5.2: The `ares` data set with an approximation of its dominant three pairwise-orthogonal scene orientations extracted using our approach, with the inlier segments of their corresponding vanishing points shown per view in red, green and blue, respectively.

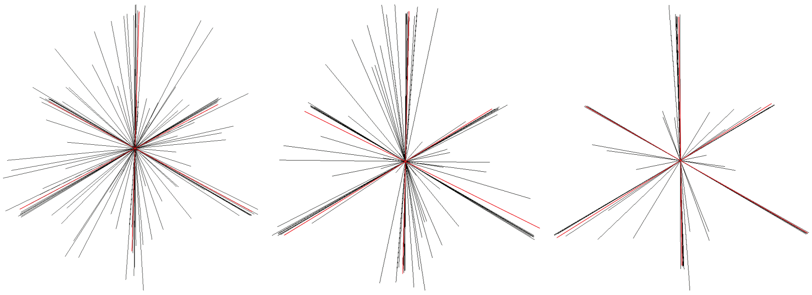


(a) (b) (c) (d) (e)

Figure 5.3: The `techgate` data set (note the displacement of the lamp post) with an approximation of its dominant three pairwise-orthogonal scene orientations extracted using our approach, with the inlier segments of their corresponding vanishing points shown per view in red, green and blue, respectively.



(a) The `acv` data set. (b) The `ares` data set. (c) The `techgate` data set.



(d) The `acv` data set. (e) The `ares` data set. (f) The `techgate` data set.

Figure 5.4: Antipodal unit direction vectors extracted across all views of the given data set, with the corresponding best-fit tripod indicated in red. The top row corresponds to the results obtained using our approach and given in Figures 5.1, 5.2 and 5.3, respectively; the bottom, to our tripod fitting with respect to the antipodal directions obtained via the approach of Sinha et al.

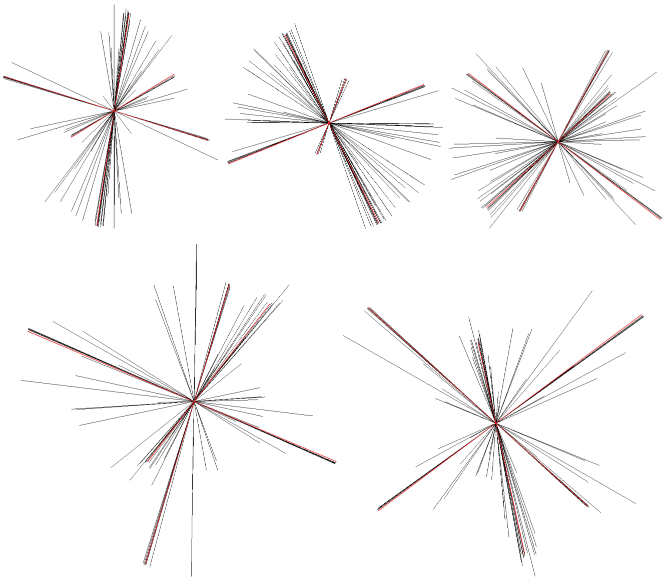


Figure 5.5: A best-fit tripod and the antipodal directions (obtained via the method of Sinha et al.) to which it was fitted using our tripod fitting technique, viewed from a handful of different poses.

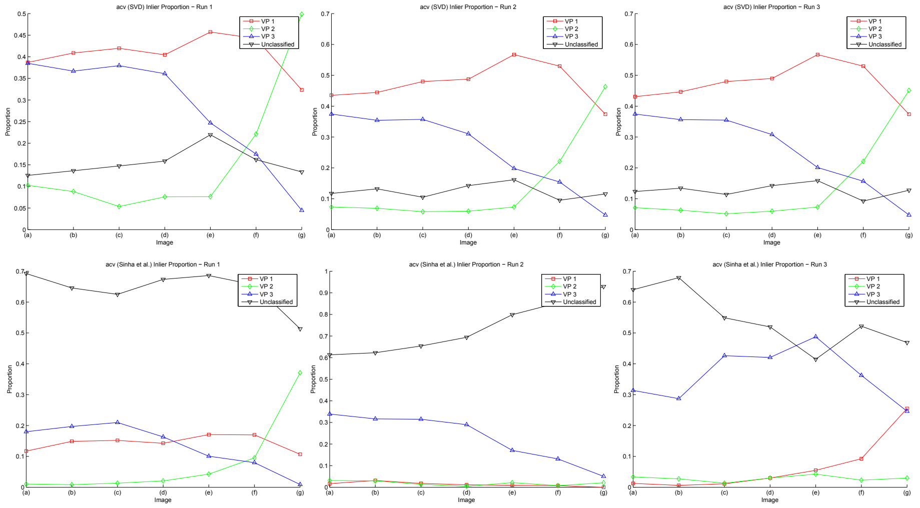


Figure 5.6: **Inlier proportions** for the `acv` data set across three runs. The top row corresponds to the results obtained using our approach; the bottom, to our tripod fitting with respect to the antipodal directions obtained via the approach of Sinha et al. In both cases, run 1 refers to selfsame respective run that gave rise to the corresponding tripod in Figure 5.4.

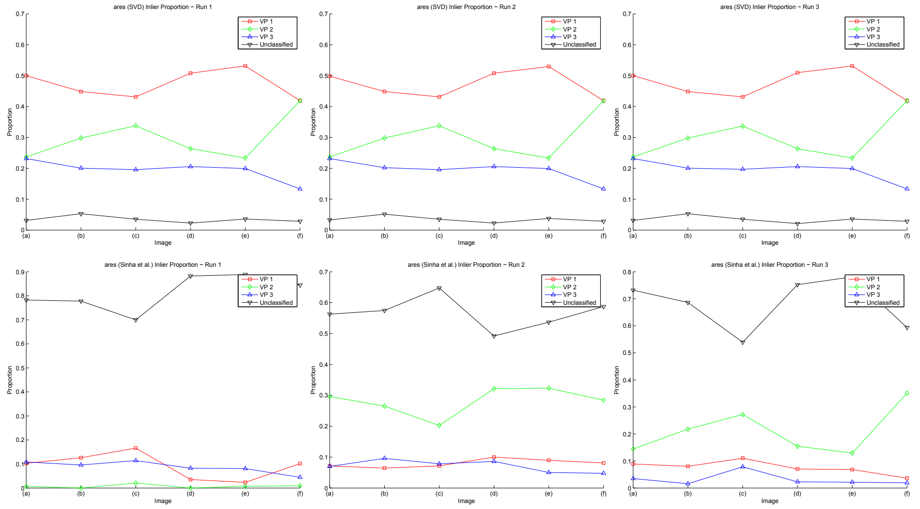


Figure 5.7: **Inlier proportions** for the `ares` data set across three runs. The top row corresponds to the results obtained using our approach; the bottom, to our tripod fitting with respect to the antipodal directions obtained via the approach of Sinha et al. In both cases, run 1 refers to selfsame respective run that gave rise to the corresponding tripod in Figure 5.4. Note that the graphs corresponding to runs 2 and 3 of our approach are indeed distinct, and that VP 2 had no inliers for images (b) and (d) in run 1 of the approach of Sinha et al.

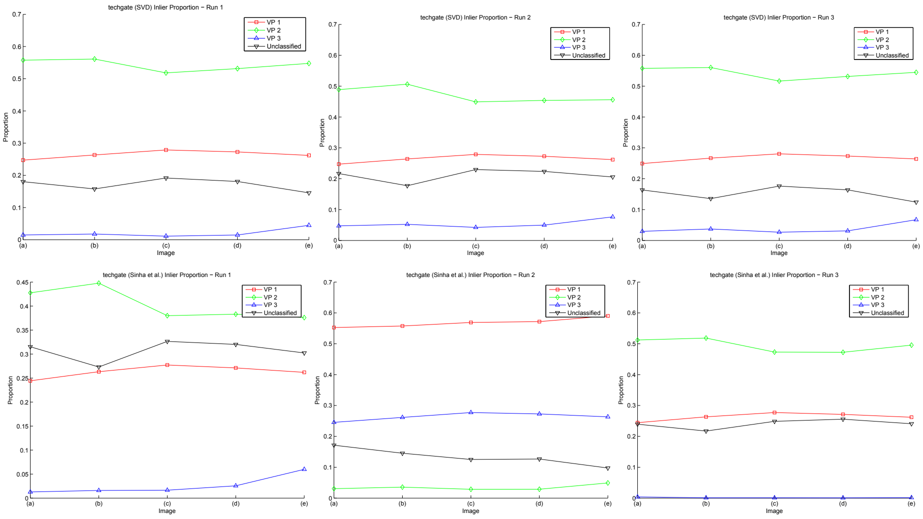


Figure 5.8: **Inlier proportions** for the `techgate` data set across three runs. The top row corresponds to the results obtained using our approach; the bottom, to our tripod fitting with respect to the antipodal directions obtained via the approach of Sinha et al. In both cases, run 1 refers to selfsame respective run that gave rise to the corresponding tripod in Figure 5.4.

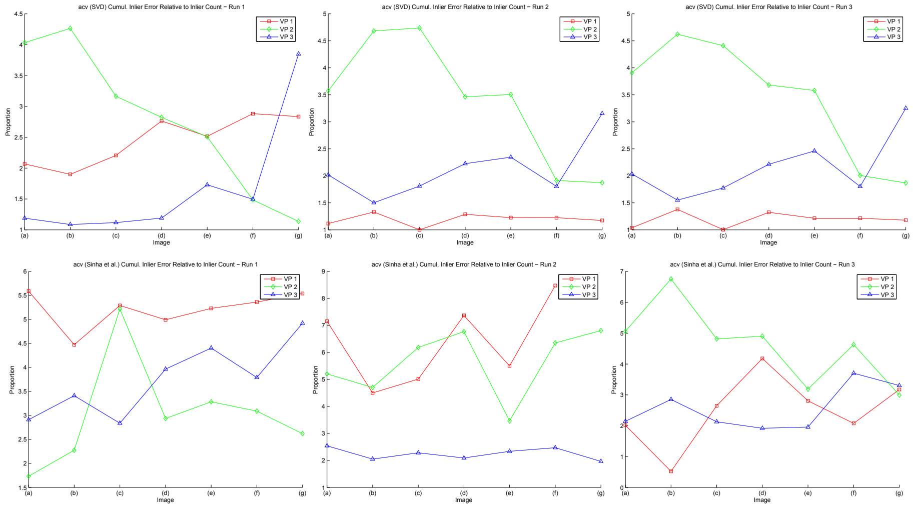


Figure 5.9: **Cumulative inlier error relative to inlier count** for the `acv` data set across three runs. The top row corresponds to the results obtained using our approach; the bottom, to our tripod fitting with respect to the antipodal directions obtained via the approach of Sinha et al. In both cases, run 1 refers to selfsame respective run that gave rise to the corresponding tripod in Figure 5.4.

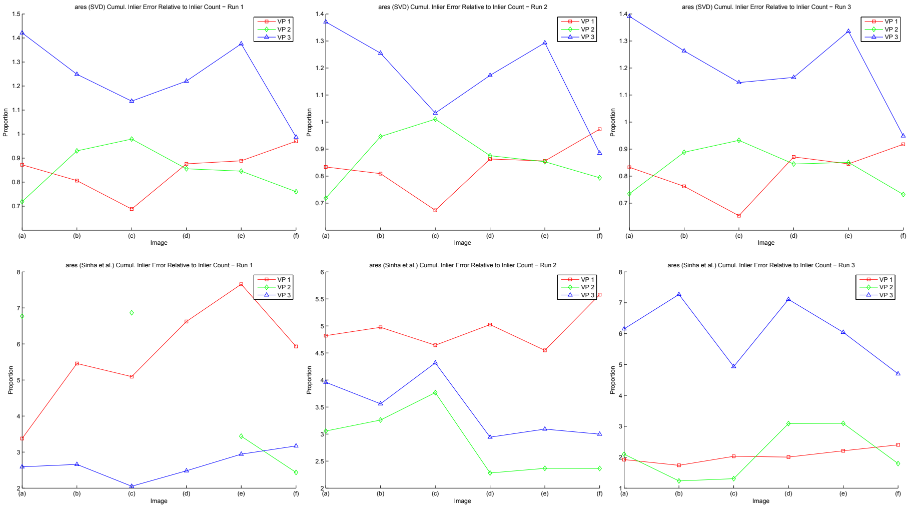


Figure 5.10: **Cumulative inlier error relative to inlier count** for the `ares` data set across three runs. The top row corresponds to the results obtained using our approach; the bottom, to our tripod fitting with respect to the antipodal directions obtained via the approach of Sinha et al. In both cases, run 1 refers to selfsame respective run that gave rise to the corresponding tripod in Figure 5.4. Note that the missing values in run 1 of the approach of Sinha et al. are due to the fact that the corresponding inlier counts are naught (cf. Figure 5.7).

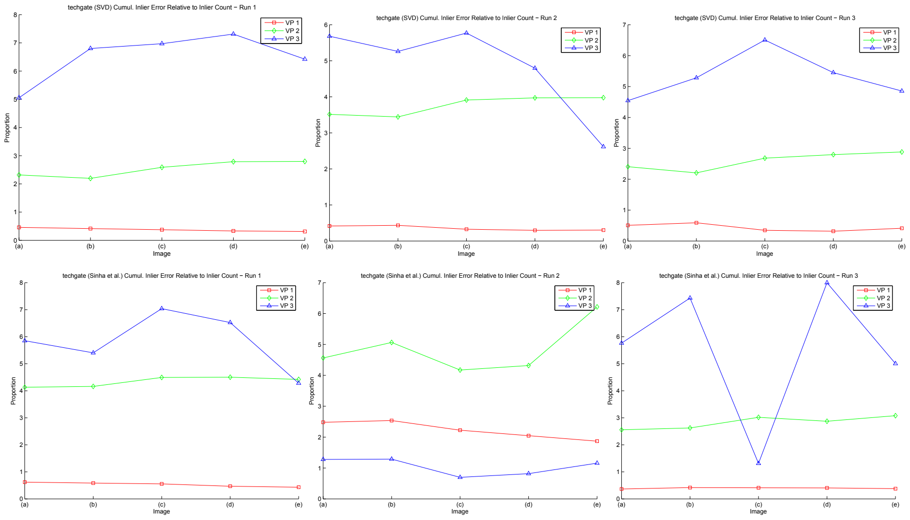


Figure 5.11: **Cumulative inlier error relative to inlier count** for the `techgate` data set across three runs. The top row corresponds to the results obtained using our approach; the bottom, to our tripod fitting with respect to the antipodal directions obtained via the approach of Sinha et al. In both cases, run 1 refers to selfsame respective run that gave rise to the corresponding tripod in Figure 5.4.

Chapter 6

Conclusion

Our approach presents a material refinement of the multiple-view vanishing point extraction technique proposed in Sinha et al. [38]. Our method achieves this refinement by making use of a strong orthogonality criterion per view, optimal segment intersection estimation and a novel tripod fitting technique. Unlike Sinha et al., our tripod fitting paradigm does not require that we assume that one of the extracted scene orientations corresponds to a cluster of points on the unit sphere "that is most well aligned with the up vector for most of the cameras" (cf. Sinha et al.), and guarantees a genuinely pairwise-orthogonal result. By considering antipodal directions, our approach yields results that make better use of the information extracted per view. Moreover, by re-estimating candidate vanishing points according to their inlier segments, we obtain information per view that is more representative of the underlying scene geometry. Finally, by enforcing orthogonality with respect to the constellations extracted per view, we restrict our fitting to relevant potential scene orientations. We found in our experiments that our method consistently outperformed the fundamental approach of Sinha et al., yielding results that were comparatively more stable across runs and that in each case corresponded closely to the respective dominant three pairwise-orthogonal orientations of each of the three scenes considered.

6.1 Recommendations

Our approach is intended as only a single step in the processing pipeline of a larger framework for the reconstruction of (typical) urban scenes. In this regard, we should like to offer the following recommendations—which we consider consequences of our evaluation in Chapter 5 coupled with good sense—in the hope that applying them might lead to better scene reconstructions.

Line Segments. As our algorithm operates on line segments extracted across views, the quality of those segments necessarily influences the quality of the results. Accordingly, the user ought to have control over the parameters that control the output of the chosen line segment extraction algorithm. Since long segments are more likely to be accurate than short ones, another parameter over which the user ought to have control is minimal segment length.

Bad Views. It is not necessarily expedient to optimize scene orientations across views that—upon the user's judgment—contain a predominance of 'bad' segments, even if our algorithm should be robust to some amount of bad data. Accordingly, the user ought to be in a position to remove such bad images from consideration in our multiple-view optimization step.

Fitting a General Tripod. There is to our knowledge in principle no reason why our tripod fitting approach cannot be adjusted to search for triplets of orientations that are something other than pairwise-orthogonal. The only part of the fitting approach that explicitly assumes that we seek pairwise-orthogonal orientations is the initialization step, which orthogonalizes the back-projection of the constellation extracted in a single view. Accordingly, fitting a general tripod reduces to formulating an appropriate initialization strategy.

Additional Scene Orientations. Real-world urban scenes often feature more than only three dominant pairwise-orthogonal scene orientations. Given k views of a scene and a set $\mathcal{S}$ of segments per view, one way to extract additional scene orientations—and, indeed, the manner according to which Sinha et al. proceed—is to allow the user to manually draw (or select) two segments in any one view known by the user to correspond to the selfsame scene orientation; the back-projection of their intersection $\mathbf{v}$ gives the intersection's corresponding scene orientation. One way to refine this result follows neatly from our approach for coming close to finding the scene's dominant three pairwise-orthogonal orientations:

Algorithm 3 Computing an Additional Scene Orientation

```
1: compute an optimal re-estimation $\hat{\mathbf{v}}$ of $\mathbf{v}$ with respect to the inliers $\mathcal{S}_{\mathbf{v}} \subseteq \mathcal{S}$ of $\mathbf{v}$,
   disregarding all segments in $\mathcal{S}$ corresponding to the inliers of the pre-computed
   dominant three pairwise-orthogonal scene orientations
2: back-project $\hat{\mathbf{v}}$ to an antipodal pair of unit vectors
3: project the corresponding orientation to and subsequently carry out steps (1) and
   (2) for each of the $k - 1$ remaining views
4: return a single orientation fitted to the $k$ antipodal directions thus obtained in a
   manner akin to our tripod fitting approach
```

A more automatic—albeit less robust—avenue would involve removing the inliers across all k views corresponding to the extracted dominant three pairwise-orthogonal orientations and clustering over candidate orientations obtained from the remaining segments in a manner akin to the approach of Sinha et al.

Segment Intersection Estimation. We carried out the evaluation of our approach on results we obtained from per-view constellations refined using the SVD-based intersection estimation technique we present in Section 4.2 of Chapter 4. In our experiments, the orientations extracted per view already corresponded closely to their best-fit tripod (cf. Figure 5.4); however, our experiments also showed that small deviations can effect material differences in cumulative inlier error relative to inlier count. As we also noted in Section 4.2 of Chapter 4, we can obtain a potentially better intersection estimation—albeit at greater cost—using the ML estimation approach of Liebowitz. Accordingly, it ought to be up to the user to decide whether the improvement over the SVD approach obtained using the ML approach merits the additional running time.

Chapter 7

Summary

In this master's thesis, we present a material refinement of the method proposed in Sinha et al. [38] for obtaining a close approximation of the dominant three pairwise-orthogonal orientations of a typical urban scene by means of extracting vanishing points across multiple views. Our method achieves this refinement by making use of a strong orthogonality criterion per view, optimal segment intersection estimation and a novel tripod fitting technique. We found in our experiments that our method consistently outperformed the fundamental approach of Sinha et al. Our method yielded results that were comparatively more stable across runs and that in each case corresponded closely to the respective dominant three pairwise-orthogonal orientations of each of the three scenes considered. Our thesis places our method into the context of earlier work on the extraction of vanishing points in the aim of facilitating the reconstruction of urban scenes. Moreover, our thesis includes what is intended to be a self-contained primer to the geometry that underlies the formation of vanishing points.

Chapter 8

Zusammenfassung

In dieser Diplomarbeit wird eine wesentliche Verfeinerung der Methode von Sinha et al. präsentiert, die mittels Extraktion von Fluchtpunkten über mehrere Ansichten hinweg eine nahe Approximation der drei dominanten paarweise orthogonalen Orientierungen einer typischen urbanen Szene berechnet. Unsere Methode erreicht diese Verfeinerung durch Verwendung eines starken Orthogonalitätskriteriums in jeder Ansicht, einer optimalen Berechnung von Segmentschnittpunkten und einem neuartigen Dreibein-Ausrichtungsverfahren. In unseren Experimenten hat unsere Methode konsequent den fundamentalen Ansatz von Sinha et al. übertroffen. Die Ergebnisse waren vergleichsweise stabiler und stellten eine nahe Approximation der jeweiligen dominanten drei paarweise orthogonalen Orientierungen in jeder der drei getesteten Szenen dar. Diese Arbeit stellt unsere Methode in den Kontext früherer Arbeiten zum Thema Fluchtpunktextrahierung, mit Schwerpunkt Vereinfachung der Rekonstruktion urbaner Szenen. Desweiteren beinhaltet diese Arbeit eine in sich geschlossene Einführung in die Geometrie, die der Entstehung von Fluchtpunkten zugrundeliegt.

Appendix A

The Single-View Approach of Rother

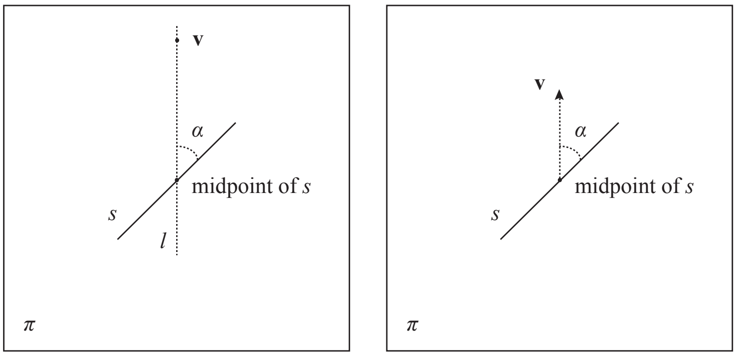
Rother's [34] single-view algorithm for extracting a constellation of three vanishing points corresponding to pairwise-orthogonal scene orientations is divided into two steps: the first is called the *accumulation step*, the second, the *search step*. In the accumulation step, votes are tallied for each of a set of candidate vanishing points computed from extracted image line segment intersections, according to each candidate's support with respect to the segments. In the search step, those votes are used—alongside constraints of camera geometry and orthogonality of the orientations corresponding to candidate vanishing points—to extract the winning constellation.

The algorithm's worst-case complexity is $O(n^5)$ in the number of line segments extracted from the image in a pre-processing step. However, constraints built into the algorithm materially reduce the likelihood of running at worst-case complexity.

Distance Functions. Rother makes use of two distance functions within the framework of his algorithm. One, as illustrated in Figure A.1(a), gives an angular measure $d(\mathbf{v}, s) = \alpha \in [0, \pi/2]$ of the extent to which a candidate vanishing point—perhaps at infinity—represented by $\mathbf{v} \in \mathbb{P}^2$ is expected to correspond to an image line segment s, where an angle $\alpha = 0$ indicates perfect correspondence. Rother uses this first distance function in his accumulation step. It is this distance function that Sinha et al. [38] borrow for their multiple-view vanishing point extraction approach, and which we borrow for our approach implemented for the purposes of this master's thesis. The other, as shown in Figure A.1(b), provides a measure of the distance between a line l and a segment s, both in the image plane, expressed as a tuple $d(l, s) = (|d|, \alpha)$. We explain the meaning of $|d|$ and α in the figure. Rother uses this second distance function in his search step.

A.1 Accumulation Step

Candidate Vanishing Points. Given a pre-computed set $\mathcal{S}$ of n image line segments, Rother computes a candidate vanishing point $\mathbf{v}$ from each non-collinear pair of the $\binom{n}{2}$ possible pairs $\{s_1, s_2\} \subset \mathcal{S}$ of segments. He does this per pair by calculating the point of intersection of the unique pair of lines $l_1, l_2 \subset \mathbb{R}^2$ that pass, respectively, through the line segments $s_1, s_2 \subset \mathbb{R}^2$ in the image plane.



(a) The distance $d(\mathbf{v}, s)$ between a line segment s and a finite vanishing point $\mathbf{v}$ is defined as the lesser angle $\alpha \in [0, \pi/2]$ between s and the the line l joining the midpoint of s with $\mathbf{v}$.

(b) As an infinite vanishing point is represented by an *orientation*, the distance $d(\mathbf{v}, s)$ given an infinite vanishing point is thus defined as the lesser angle α between s and a vector extending from the midpoint of s with the orientation of the infinite vanishing point.

Figure A.1: The distance function $d(\mathbf{v}, s)$ of Rother's algorithm.

By using the homogeneous coordinates of $\mathbb{P}^2$ to represent lines in the plane, we may compute the intersection at infinity of lines parallel in the image plane in the same way as we would the intersection of lines that meet in a finite point. As shown in Section 3.1.1 of Chapter 3, we obtain the homogeneous representation $\mathbf{l} \in \mathbb{P}^2$ of a line $l \subset \mathbb{R}^2$ in the image plane by working out the vector product of two distinct points $\mathbf{p}_1, \mathbf{p}_2 \in \mathbb{P}^2$ on l,

$$\mathbf{l} \sim \mathbf{p}_1 \times \mathbf{p}_2.$$

On account of image noise, the best two points on l to choose are the homogenized endpoints of the corresponding segment s. The point of intersection $\mathbf{v} \in \mathbb{P}^2$ of two lines $\mathbf{l}, \mathbf{l}' \in \mathbb{P}^2$ is given likewise by

$$\mathbf{v} \sim \mathbf{l} \times \mathbf{l}'.$$

We then normalize $\mathbf{v} = (v_1, v_2, v_3)^\top$ such that $\mathbf{v} = (v_1, v_2, 0)^\top$ if the magnitude of v_3 is not much greater than the machine epsilon, and $\mathbf{v} = (v_1/v_3, v_2/v_3, 1)^\top$ otherwise.

Endpoint Criterion. Lines parallel in space are projected either to lines parallel in the image plane that meet at infinity, or to lines in the image plane that meet in a finite point. Consequently, a vanishing point will not lie anywhere on an image line segment but at one of the segment's endpoints (an exception would be a horizon line, or indeed any vanishing line). For this reason, we reject all candidate vanishing points $\mathbf{v} \in V$ that lie between the endpoints of an image segment.

Voting Scheme. Rother assigns a vote to each valid candidate vanishing point $\mathbf{v}$. A higher vote for $\mathbf{v}$ is assumed to indicate a higher likelihood that the candidate vanishing point is a veridical one. We formulate the voting function that Rother provides as

$$\text{vote}(\mathbf{v}) = \sum_{s \in \mathcal{S}_\mathbf{v}} \left[\left(1 - \frac{d(\mathbf{v}, s)}{t}\right) + \lambda \left(\frac{\text{length}(s)}{\max\{\text{length}(s') \mid s' \in \mathcal{S}_\mathbf{v}\}} \right) \right], \qquad \text{(A.1)}$$

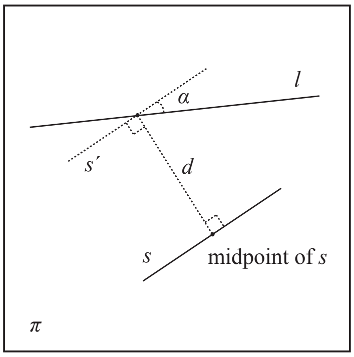


Figure A.2: The distance function $d(l, s) = (|d|, \alpha)$ of Rother's algorithm. The distance $d(l, s)$ between a line l and a line segment s is defined as the tuple $(|d|, \alpha)$, where $|d|$ is the length of the segment d perpendicular to s joining the midpoint of s with l and $\alpha \in [0, \pi/2]$ is the lesser angle between the midpoint of the segment s' and the line l. The segment s' is obtained by translating s by its midpoint along the segment d.

where $t \in]0, \pi/2]$ is a user-specified threshold that sets the maximal allowable magnitude of $d(\mathbf{v}, s)$ since, for $0 \leq d(\mathbf{v}, s) \leq t$, the first term of the voting function is between 1 and 0, inclusive; $\mathcal{S}_\mathbf{v} \subseteq \mathcal{S}$ is the set of all image line segments s for which, accordingly, $0 \leq d(\mathbf{v}, s) \leq t$; and λ is a user-specified weight parameter that establishes the relative influence of the two terms of the voting function. Note that for $\lambda = 1$, the maximal value of both terms, respectively, is 1. The motivation for including the second term of the voting function follows from the assumption that longer line segments are more reliable than shorter ones.

Pseudocode. The first of the two steps in Rother's algorithm is the accumulation step, which takes as input a set $\mathcal{S}$ of line segments extracted from the image in a preprocessing step. We give the pseudocode in Algorithm 4.

Algorithm 4 Rother's Accumulation Step

```
$\mathcal{V} \leftarrow \varnothing$ {the set of candidate vanishing points}
for all pairs $\{s_1, s_2\} \subset \mathcal{S}$ of non-collinear line segments do
    compute candidate vanishing point $\mathbf{v}$ from the intersection of $s_1, s_2$
    $\mathcal{V} \leftarrow \mathcal{V} \cup \{\mathbf{v}\}$ {add $\mathbf{v}$ to the set $\mathcal{V}$ of candidate vanishing points}
    for all line segments $s \in \mathcal{S}$ do
        if $\mathbf{v}$ does not satisfy the endpoint criterion for $s$ then
            $\mathcal{V} \leftarrow \mathcal{V} \setminus \{\mathbf{v}\}$ {remove $\mathbf{v}$ from the set $\mathcal{V}$ of candidate vanishing points}
            continue
        end if
    end for
    $\mathcal{S}_\mathbf{v} \leftarrow$ the set of all segements $s \in \mathcal{S}$ such that $d(\mathbf{v}, s) \leq t$
    $\mathrm{vote}_\mathbf{v} \leftarrow \mathrm{vote}(\mathbf{v})$, computed over the set $\mathcal{S}_\mathbf{v} \subseteq \mathcal{S}$
end for
```

A.2 Search Step

Camera and Orthogonality Criteria. The camera and orthogonality criteria are motivated by constraints imposed by camera geometry on constellations of vanishing point triplets corresponding to pairwise-orthogonal scene orientations. These constraints are discussed more thoroughly in Liebowitz and Zisserman [24]. Taken collectively, the camera and orthogonality criteria for triplets of vanishing points that correspond to pairwise-orthogonal scene orientations are:

i. Three finite $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$: no interior angle of the triangle formed by the three finite vanishing points is greater than or equal to $\pi/2$. We can compute the principle point, which is the orthocenter of the triangle formed by the three candidate vanishing points. Moreover, we can also calculate the focal length, which is the distance from the principal point to the apex of the pyramid whose base is the triangle and whose apex is formed by the right-angle intersections of segments extending from the three candidate vanishing points in the base;

ii. Two finite $\mathbf{v}_1, \mathbf{v}_2$, one infinite $\mathbf{v}_3$: the direction of $\mathbf{v}_3$ is orthogonal to the line through $\mathbf{v}_1, \mathbf{v}_2$. The principal point lies on the segment whose endpoints are $\mathbf{v}_1, \mathbf{v}_2$; since the principal point for a typical camera is near the image center, we choose the point on the segment closest to the image center. This information allows us to compute the focal length;

iii. One finite $\mathbf{v}_1$, two infinite $\mathbf{v}_2, \mathbf{v}_3$: the direction of $\mathbf{v}_2$ is orthogonal to the direction of $\mathbf{v}_3$. The principle point lies at $\mathbf{v}_1$, we cannot compute the focal length.

Vanishing Line Criterion. Two vanishing points $\mathbf{v}, \mathbf{v}'$ corresponding to coplanar scene orientations share a vanishing line if at least one of the two is finite. If both are finite, the vanishing line is the line through the two; if only one is finite, it is the line through the finite vanishing point in the direction of the infinite vanishing point, as discussed in Section 3.5 of Chapter 3. We formulate the vanishing line criterion accordingly:

i. Two finite $\mathbf{v}, \mathbf{v}'$: each segment $s \in \mathcal{S}_{\mathbf{v}} \cap \mathcal{S}_{\mathbf{v}'}$ lies on the vanishing line through the two vanishing points;

ii. One finite $\mathbf{v}$, one infinite $\mathbf{v}'$: the sets $\mathcal{S}_{\mathbf{v}}, \mathcal{S}_{\mathbf{v}'}$ are disjoint; i.e., $\mathcal{S}_{\mathbf{v}} \cap \mathcal{S}_{\mathbf{v}'} = \varnothing$.

According to Rother's approach, a segment s lies on a vanishing line l if, having computed $d(l, s) = (|d|, \alpha)$, the distance $|d|$ and the angle α are each below a threshold.

Pseudocode. The second of the two steps of the Rother algorithm is the search step, which extracts the winning constellation in a combinatorial fashion. We give the pseudocode in Algorithm 5.

Algorithm 5 Rother's Search Step

```
$\mathbf{v}_1 \leftarrow \arg\max_{\mathbf{v}} \mathrm{vote}(\mathbf{v})$
for all pairs $\{\mathbf{v}', \mathbf{v}''\} \subset \mathcal{V} \setminus \{\mathbf{v}_1\}$ of candidate vanishing points do
  if $\{\mathbf{v}_1, \mathbf{v}'\}, \{\mathbf{v}_1, \mathbf{v}''\}, \{\mathbf{v}', \mathbf{v}''\}$ satisfy the vanishing line criterion then
    if the constellation $\{\mathbf{v}_1, \mathbf{v}', \mathbf{v}''\}$ satisfies the camera and orthogonality criteria then
      $\mathrm{vote}_{\{\mathbf{v}', \mathbf{v}''\}} \leftarrow \mathrm{vote}(\mathbf{v}') + \mathrm{vote}(\mathbf{v}'')$ {the constellation's vote}
    end if
  end if
end for
return the constellation $\{\mathbf{v}_1, \mathbf{v}', \mathbf{v}''\}$ with the largest vote
```

Appendix B

The Multiple-View Approach of Sinha et al.

Sinha et al. [38] present a multiple-view approach for extracting the dominant three pairwise-orthogonal orientations—and with them potentially additional orientations—of an urban scene by means of vanishing points. They describe their method in a short appendix, remarking elsewhere in the same publication that the extraction of vanishing points is "not the main focus of [their] paper." Even so, theirs is one of the few papers that describe the application of knowledge of vanishing points extracted across multiple views to facilitating the reconstruction of urban scenes (cf. Chapter 2).

B.1 Algorithm

Three Pairwise-Orthogonal Scene Orientations. Sinha et al. begin by extracting up to n candidate vanishing points per view[1] using a RANSAC-based approach (cf. Appendix C) with support defined in terms of inlier count with respect to the distance measure $d(\mathbf{v}, s)$ of Rother (cf. Appendix A); a segment s is an inlier of a candidate vanishing point $\mathbf{v}$ if $d(\mathbf{v}, s) < T_{\mathrm{Roth}}$ for some chosen threshold T_{Roth}. Once up to n candidate vanishing points have been extracted in each of the k views, Sinha et al. back-project each candidate vanishing point to its corresponding normalized direction vector, which they place on a unit sphere. Next, they cluster—albeit without disclosing how—the points on that unit sphere, extracting the cluster center best alligned with the up vector for most of the cameras. From among the remaining clusters, they obtain another two, collectively constrained to correspond to pairwise-orthogonal orientations. Additionally, Sinha et al. use the three pairwise-orthogonal orientations to refine their camera pose estimation. As with their clustering, however, so too with their pose re-estimation do they choose to pass over the greater details in silence.

Additional Scene Orientations. Sinha et al. allow for the interactive selection of additional scene orientations, presumably from among the remaining available cluster centers. Alternatively, they also allow for the user to draw a pair of lines in a chosen view, known by the user to correspond to lines parallel in the scene; the back-projection of their point of intersection gives the corresponding scene orientation.

---

[1] Sinha et al. report having used $n = 8$ in their experiments.

Pseudocode. We give the pseudocode of the multiple-view approach of Sinha et al. for extracting the dominant three pairwise-orthogonal orientations in Algorithm 6.

Algorithm 6 The Multiple-View Approach of Sinha et al.

```
recover camera geometry for the $k$ available views of the scene
for all $k$ available views of the scene do
    $\mathcal{S}_k \leftarrow$ the set of segments extracted from the $k^{\mathrm{th}}$ view
    $\mathcal{C}_k \leftarrow \varnothing$ {the set of candidate vanishing points corresponding to the $k^{\mathrm{th}}$ view}
    while $|\mathcal{C}_k| \neq n$ do
        if there remain fewer than a pair of segments in $\mathcal{S}_k$ then
            break
        end if
        $\mathbf{v} \leftarrow$ the candidate vanishing point computed from the intersection of a pair of distinct image line segments $s_1, s_2 \in \mathcal{S}_k$ taken at random
        $\mathcal{S}'_k \leftarrow \bigcup_{s \in \mathcal{S}} \{s \mid d(\mathbf{v}, s) < T_{\mathrm{Roth}}\}$ {the set of the inlier segments of $\mathbf{v}$}
        if the candidate vanishing point $\mathbf{v}$ has best RANSAC inlier support then
            $\mathcal{C}_k \leftarrow \mathcal{C}_k \cup \{\mathbf{v}\}$
            $\mathcal{S}_k \leftarrow \mathcal{S}_k \setminus \mathcal{S}'_k$ {the set of remaining outliers}
        end if
    end while
    $\mathcal{X}_k \leftarrow$ the normalized direction vector corresponding to the back-projection of each candidate vanishing point in $\mathcal{C}$
end for
$\mathcal{X} \leftarrow \bigcup_i \mathcal{X}_i$ {the unit direction vectors extracted across $k$ views}
$\hat{\mathcal{X}} \leftarrow \mathrm{cluster}(\mathcal{X})$ {the set of cluster centers corresponding to clusters in $\mathcal{X}$}
$\mathbf{t}_{\mathrm{up}} \leftarrow$ the cluster center best alligned with the up vector for most of the cameras
$\mathcal{T} \leftarrow \mathbf{t}_{\mathrm{up}}$ and two additional cluster centers, constrained to be pairwise-orthogonal
return the three directions in $\mathcal{T}$
```

Appendix C

Random Sample Consensus

Random Sample Consensus, or RANSAC, is an algorithmic framework put forward by Fischler and Bolles [11] for robustly fitting a mathematical model—i.e., for estimating a model's parameters—to a set $\mathcal{S}$ of data points that contain outliers. The presence of outliers is characteristic of data sets that are drawn from empirical measurements.

Model fitting approaches that make equal use of all data points in $\mathcal{S}$—such as ordinary least squares—make no special provision for gross outliers in the data. Prior to the introduction of RANSAC, a popular way to address the problem of fitting mathematical models to data with outliers was to iteratively compute a model's best fit to the points in $\mathcal{S}$ and remove the point most distant from the fit, until a threshold—either in distance from the fit or number of iterations—is reached. In their paper, Fischler and Bolles provide an example of how a single gross outlier mixed in with otherwise good data can cause this particular heuristic to fail.

C.1 Framework

Given a set of data points $\mathcal{S}$ to which some particular mathematical model is to be fit, RANSAC begins with the minimal number of data points $\mathcal{S}' \subseteq \mathcal{S}$—selected at random—needed to instantiate the model's parameters M. Accordingly, in the event that we should wish to fit a line, the minimal number of data points $\mathcal{S}' \subseteq \mathcal{S}$ we would need is two. RANSAC then proceeds to determine which of the data points in $\mathcal{S}$ are within a distance threshold T_{dist} from the instantiated model. If, again, our model is a line, then its inliers are the data points in $\mathcal{S}$ that come close enough to lying on that line. These inlier data points collectively form the instantiated model's *consensus set* $\mathcal{C} \subseteq \mathcal{S}$. If the cardinality of $\mathcal{C}$—called the instantiated model's *support*—is greater than a threshold T_{size}, RANSAC invokes a smoothing technique such as least squares to yield an optimal result vis-à-vis the data points in $\mathcal{C}$, and the algorithm terminates. Otherwise, the model is reinstantiated with a new minimal set of random data points $\mathcal{S}' \subseteq \mathcal{S}$, again chosen at random. If a threshold of N iterations is reached without having encountered a large enough consensus set, it is the consensus set with best support encountered that is judged the winner, and likewise undergoes the aforementioned smoothing technique.

Pseudocode. We provide the pseudocode for the general RANSAC algorithmic framework in Algorithm 7.

Algorithm 7 The RANSAC Framework

```
$s \leftarrow$ the minimal number of data points needed to initialize $M$
for $N$ iterations do
    $\mathcal{S}' \leftarrow$ a random subset of size $s$ of data points in $\mathcal{S}$
    $M \leftarrow$ the model parameters instantiated using the data points in $\mathcal{S}'$
    $\mathcal{C} \leftarrow$ the consensus set of data points in $\mathcal{S}$ within a distance $T_{\mathrm{dist}}$ of $M$
    if $|\mathcal{C}| > T_{\mathrm{size}}$ then
        return the model parameters $M$ re-estimated using the data points in $\mathcal{C}$
    end if
end for
return the model parameters $M$ re-estimated using the data points in the best-support consensus set encountered
```

Appendix D

Singular Value Decomposition

D.1 Formulation

By the singular value decomposition (SVD)[1], we can decompose any $m \times n$ matrix $\mathtt{A}, m \geq n$, into a pair of orthogonal matrices $\mathtt{U}, \mathtt{V}$ and a diagonal matrix Σ such that

$$\mathtt{A} = \mathtt{U}\Sigma\mathtt{V}^\top = \sum_{i=1}^{r} \lambda_i \mathbf{u}_i \mathbf{v}_i^\top, \tag{D.1}$$

where r is the rank of $\mathtt{A}$. The columns of the $m \times n$ matrix $\mathtt{U}$ are the eigenvectors $\mathbf{u}_i \in \mathbb{R}^m$ of $\mathtt{AA}^\top$,

$$\mathtt{U} = \left[\begin{array}{ccc} \mathbf{u}_1 & \cdots & \mathbf{u}_n \end{array} \right]. \tag{D.2}$$

The $n \times n$ matrix Σ is a diagonal matrix with non-negative entries—called the *singular values* of $\mathtt{A}$—that are the square roots $\sigma_i = \sqrt{\lambda_i}$ of the eigenvalues λ_i of $\mathtt{A}^\top\mathtt{A}$,

$$\Sigma = \left[\begin{array}{ccc} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \\ & \mathbf{0} & \end{array} \right], \tag{D.3}$$

where $\sigma_1 \geq \cdots \geq \sigma_r \geq 0, \sigma_{r+1} = \cdots = \sigma_n = 0$. Finally, the columns of the orthogonal $n \times n$ matrix $\mathtt{V}$ are the eigenvectors $\mathbf{v}_i \in \mathbb{R}^n$ of $\mathtt{A}^\top\mathtt{A}$,

$$\mathtt{V} = \left[\begin{array}{ccc} \mathbf{v}_1 & \cdots & \mathbf{v}_n \end{array} \right]. \tag{D.4}$$

The geometric interpretation of the SVD is a rotation[2] $\mathtt{V}^\top$, followed by a (perhaps anisotropic) stretching Σ and finally a second rotation $\mathtt{U}$. A review of applications of the SVD for solving computer vision problems is available in Section A4.4 of Appendix 4 in Hartley and Zisserman [15].

---

[1] For a more in-depth discussion of the SVD, see `http://www.prip.tuwien.ac.at/teaching/ws/StME/apponly.pdf`.

[2] Let us recall that if an $m \times n$ matrix $\mathtt{M}$ is orthogonal, the column vectors $\mathbf{m}_i$ of $\mathtt{M}$ must be orthonormal, i.e., $\mathbf{m}_i^\top \mathbf{m}_j = \delta_{ij}$. Accordingly, each column vector $\mathbf{m}_i$ has unit length and each pair of column vectors $\mathbf{m}_i, \mathbf{m}_j, i \neq j$, are orthogonal. The column space of $\mathtt{M}$ is accordingly an orthonormal basis of an n-dimensional subspace of $\mathbb{R}^m$. Since it follows that $||\mathtt{M}\mathbf{x}|| = ||\mathbf{x}||, \mathbf{x} \in \mathbb{R}^n$, the matrix $\mathtt{M}$ is a rotation matrix.

D.2 Minimizing the Quantity $\|\mathtt{A}\mathbf{x}\|^2$ over $\mathbf{x}$

Given an $m \times n$ matrix $\mathtt{A} = \mathtt{U}\Sigma\mathtt{V}^\top$, $m > n$, the vector $\mathbf{x}$, $\|\mathbf{x}\| = 1$, that minimizes the quantity $\|\mathtt{A}\mathbf{x}\|^2$ is the rightmost column of $\mathtt{V}$ (cf. Hartley and Zisserman [15]).

D.3 Orthogonalizing a Square Matrix

Given an $n \times n$ matrix $\mathtt{A} = \mathtt{U}\Sigma\mathtt{V}^\top$, the least-squares orthogonalization of $\mathtt{A}$ is given by $\mathtt{U}\mathtt{V}^\top$ (cf. Schönemann [36]), which amounts to simply disregarding the influence of the stretching matrix Σ. Note that this is precisely the solution to the so-called orthogonal Procrustes[3] problem.

---

[3] Procrustes (Προκρούστης), son of Poseidon, was an Attic bandit who offered travellers a bed in which to pass the night. He is infamous for having forced his victims fit this bed by either stretching their limbs or cutting them away. A Procrustean constraint is thus one to which exact conformity is enforced.

Bibliography

[1] K. Andersen. *The Geometry of an Art: The History of the Mathematical Theory of Perspective from Alberti to Monge*. Springer Science+Business Media, LLC, New York, 2007.

[2] A. Banerjee, I. S. Dhillon, J. Ghosh, and S. Sra. Clustering on the Unit Hypersphere using von Mises-Fisher Distributions. *Journal of Machine Learning*, 6:1345–1382, 2006.

[3] S. T. Barnard. Interpreting Perspective Images. *Artificial Intelligence*, 21(4):435–462, 1983.

[4] M. Born and E. Wolf. *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light (7th Edition)*. Cambridge University Press, 7th edition, October 1999.

[5] J. Burns, A. Hanson, and E. Riseman. Extracting straight lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(4):425–445, 1986.

[6] B. Caprile and V. Torre. Using Vanishing Points for Camera Calibration, 1990.

[7] R. Cipolla and E. Boyer. 3D Model Acquisition from Uncalibrated Images. In *IAPR Workshop on Machine Vision Applications*, pages 559–568. Citeseer, 1998.

[8] R. T. Collins and R. S. Weiss. Vanishing Point Calculation as a Statistical Inference on the Unit Sphere. *Third International Conference on*, 1990.

[9] A. Criminisi. Single-View Metrology: Algorithms and Applications, 2002.

[10] R. O. Duda and P. E. Hart. Use of the Hough Transformation to Detect Lines and Curves in Pictures. *Communications of the ACM*, 15(1):11–15, 1972.

[11] M. A. Fischler and R. 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.

[12] W. E. L. Grimson and D. P. Huttenlocher. On the Sensitivity of the Hough Transform for Object Recognition. *on Pattern Analysis and Machine*, 1990.

[13] D. S. Guru, B. H. Shekar, and P. Nagabhushan. A Simple and Robust Line Detection Algorithm based on Small Eigenvalue Analysis. *Pattern Recognition Letters*, 2004.

[14] HAL9000 S.r.l. - Haltadefinizione. Leonardo da Vinci, The last supper, Milan, Santa Maria delle Grazie, 1494 - 1498.

[15] R. I. Hartley and A. Zisserman. *Multiple-View Geometry in Computer Vision*. Cambridge University Press New York, NY, USA, second edition, 2003.

[16] P. V. C. Hough. Machine Analysis of Bubble Chamber Pictures. In *International Conference on High Energy Accelerators and Instrumentation*, volume 73, 1959.

[17] A. Irschara, C. Zach, and H. Bischof. Towards Wiki-based Dense City Modeling. pages 1–8, October 2007.

[18] M. Kemp. *The Science of Art: Optical Themes in Western Art from Brunelleschi to Seurat*. Yale University Press, New Haven, 1990.

[19] F. Klein. *Elementary Mathematics from an Advanced Standpoint*. Macmillan, New York, 1939.

[20] J. Košecká and W. Zhang. Efficient Computation of Vanishing Points. *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, pages 223–228, 2002.

[21] K. Levenberg. A Method for the Solution of Certain Nonlinear Problems in Least Squares. *Quarterly of Applied Mathematics*, 1944.

[22] D. Liebowitz. *Camera Calibration and Reconstruction of Geometry from Images*. PhD thesis, 2001.

[23] D. Liebowitz and A. Zisserman. Metric Rectification for Perspective Images of Planes. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 482–488, 1998.

[24] D. Liebowitz and A. Zisserman. Combining Scene and Auto-calibration Constraints. *Proceedings of the Seventh IEEE International Conference on Computer Vision*, pages 293–300 vol.1, 1999.

[25] M. I. A. Lourakis. levmar: Levenberg-Marquardt Non-Linear Least Squares Algorithms in C/C++.

[26] E. Lutton, H. Maître, and J. Lopez-Krahe. Contribution to the Determination of Vanishing Points using Hough Transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 430–438, 1994.

[27] Y. Ma, S. Soatto, J. Košecká, and S. S. Sastry. *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer Verlag, 2003.

[28] M. J. Magee and J. K. Aggarwal. Determining Vanishing Points from Perspective Images. *Computer Vision, Graphics, and Image Processing*, 26(2):256–267, 1984.

[29] D. W. Marquardt. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *Journal of the Society for Industrial and Applied Mathematics*, 1963.

[30] J. L. Mundy. The Relationship between Photogrammetry and Computer Vision. *CAD and CG Sinica*, 2002.

[31] R. Pflugfelder. *Self-Calibrating Cameras in Video Surveillance*. PhD thesis, 2008.

[32] L. Quan and R. Mohr. Determining Perspective Structures using Hierarchical Hough Transform. *Pattern Recognition Letters*, 9(4):279–286, 1989.

[33] P. Rosin and G. West. Segmentation of Edges into Lines and Arcs. *Image and Vision Computing*, 7:109–114, 1989.

[34] C. Rother. A New Approach to Vanishing Point Detection in Architectural Environments. *Image and Vision Computing*, 20:647–655, 2002.

[35] C. Rother. *Multi-View Reconstruction and Camera Recovery using a Real or Virtual Reference Plane*. PhD thesis, 2003.

[36] P. H. Schönemann. A Generalized Solution of the Orthogonal Procrustes Problem. *Psychometrika*, 31(1):1–10, 1966.

[37] J. A. Shufelt. Performance evaluation and analysis of vanishing point detection techniques. *IEEE Transactions on Pattern Analysis and Machine*, 1999.

[38] S. N. Sinha, D. Steedly, R. Szeliski, M. Agrawala, and M. Pollefeys. Interactive 3D Architectural Modeling from Unordered Photo Collections. *ACM Transactions on Graphics*, 27(5):1, December 2008.

[39] C. E. Springer. *Geometry and Analysis of Projective Spaces*. W. H. Freeman and Company, San Francisco, 1964.

[40] F. A. van Den Heuvel. Vanishing Point Detection for Architectural Photogrammetry. *International Archives of Photogrammetry and Remote Sensing*, 32:652–659, 1998.

[41] T. Werner and A. Zisserman. New Techniques for Automated Architectural Reconstruction from Photographs, 2002.