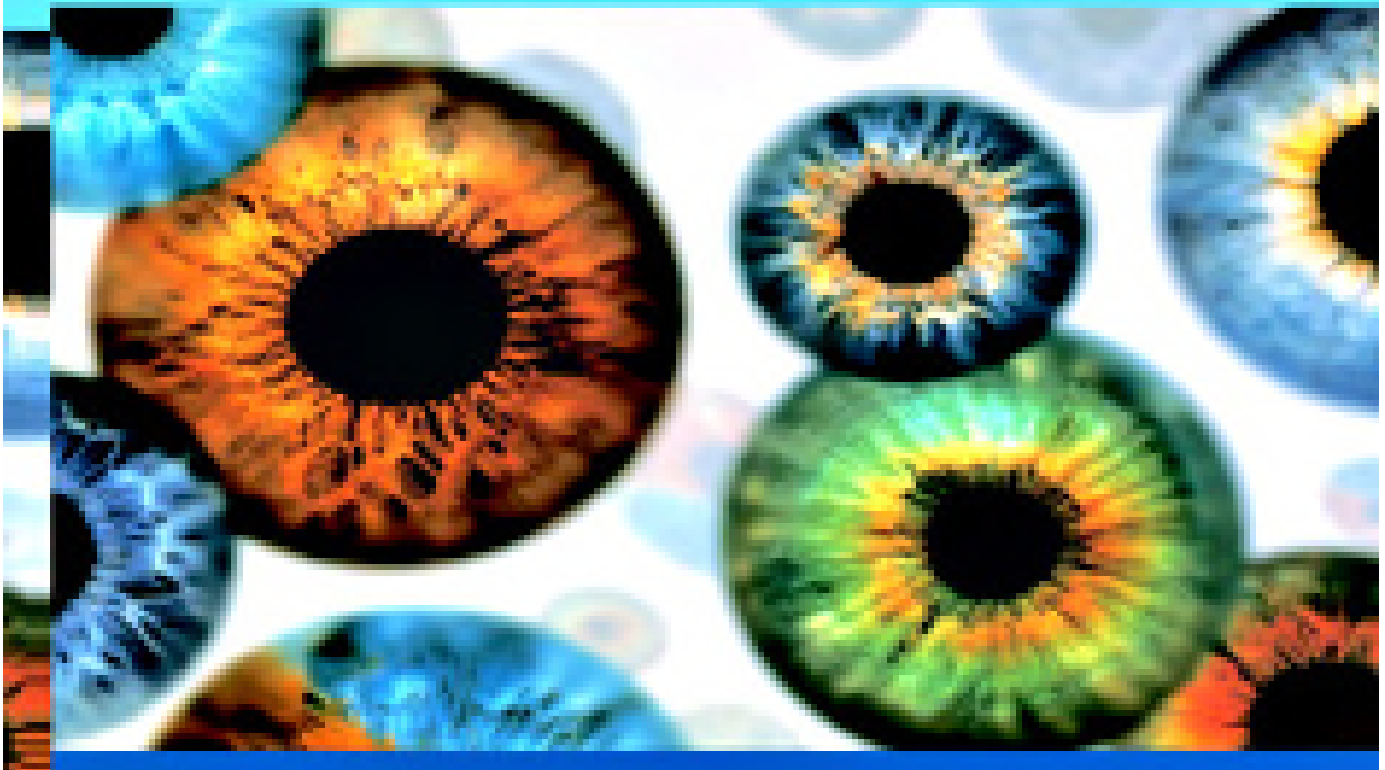


SECOND EDITION

# Multiple View Geometry

in computer vision



**Richard Hartley and Andrew Zisserman**

CAMBRIDGE

Motive Technologies Exhibit 2010,  
IPR2026-00034

CAMBRIDGE

[www.cambridge.org/9780521540513](http://www.cambridge.org/9780521540513)

# **Multiple View Geometry in Computer Vision**

**Second Edition**

Richard Hartley

Australian National University,  
Canberra, Australia

Andrew Zisserman

University of Oxford, UK



**CAMBRIDGE**  
UNIVERSITY PRESS

CAMBRIDGE UNIVERSITY PRESS

Cambridge, New York, Melbourne, Madrid, Cape Town, Singapore, São Paulo

Cambridge University Press

The Edinburgh Building, Cambridge CB2 2RU, UK

Published in the United States of America by Cambridge University Press, New York

[www.cambridge.org](http://www.cambridge.org)

Information on this title: [www.cambridge.org/9780521540513](http://www.cambridge.org/9780521540513)

© Cambridge University Press 2000, 2003

This publication is in copyright. Subject to statutory exception and to the provision of relevant collective licensing agreements, no reproduction of any part may take place without the written permission of Cambridge University Press.

First published in print format 2004

ISBN-13 978-0-521-18618-9 eBook (EBL)

ISBN-10 0-521-18618-5 eBook (EBL)

ISBN-13 978-0-521-54051-3 paperback

ISBN-10 0-521-54051-8 paperback

Cambridge University Press has no responsibility for the persistence or accuracy of URLs for external or third-party internet websites referred to in this publication, and does not guarantee that any content on such websites is, or will remain, accurate or appropriate.

## Dedication

This book is dedicated to Joe Mundy whose vision and constant search for new ideas led us into this field.

---

# Contents

|                                                                                    |             |      |
|------------------------------------------------------------------------------------|-------------|------|
| <i>Foreword</i>                                                                    | <i>page</i> | xi   |
| <i>Preface</i>                                                                     |             | xiii |
| <b>1 Introduction – a Tour of Multiple View Geometry</b>                           |             | 1    |
| 1.1 Introduction – the ubiquitous projective geometry                              |             | 1    |
| 1.2 Camera projections                                                             |             | 6    |
| 1.3 Reconstruction from more than one view                                         |             | 10   |
| 1.4 Three-view geometry                                                            |             | 12   |
| 1.5 Four view geometry and $n$ -view reconstruction                                |             | 13   |
| 1.6 Transfer                                                                       |             | 14   |
| 1.7 Euclidean reconstruction                                                       |             | 16   |
| 1.8 Auto-calibration                                                               |             | 17   |
| 1.9 The reward I : 3D graphical models                                             |             | 18   |
| 1.10 The reward II: video augmentation                                             |             | 19   |
| <b>PART 0: The Background: Projective Geometry, Transformations and Estimation</b> |             | 23   |
| <i>Outline</i>                                                                     |             | 24   |
| <b>2 Projective Geometry and Transformations of 2D</b>                             |             | 25   |
| 2.1 Planar geometry                                                                |             | 25   |
| 2.2 The 2D projective plane                                                        |             | 26   |
| 2.3 Projective transformations                                                     |             | 32   |
| 2.4 A hierarchy of transformations                                                 |             | 37   |
| 2.5 The projective geometry of 1D                                                  |             | 44   |
| 2.6 Topology of the projective plane                                               |             | 46   |
| 2.7 Recovery of affine and metric properties from images                           |             | 47   |
| 2.8 More properties of conics                                                      |             | 58   |
| 2.9 Fixed points and lines                                                         |             | 61   |
| 2.10 Closure                                                                       |             | 62   |
| <b>3 Projective Geometry and Transformations of 3D</b>                             |             | 65   |
| 3.1 Points and projective transformations                                          |             | 65   |
| 3.2 Representing and transforming planes, lines and quadrics                       |             | 66   |

|          |                                                              |            |
|----------|--------------------------------------------------------------|------------|
| 3.3      | Twisted cubics                                               | 75         |
| 3.4      | The hierarchy of transformations                             | 77         |
| 3.5      | The plane at infinity                                        | 79         |
| 3.6      | The absolute conic                                           | 81         |
| 3.7      | The absolute dual quadric                                    | 83         |
| 3.8      | Closure                                                      | 85         |
| <b>4</b> | <b>Estimation – 2D Projective Transformations</b>            | <b>87</b>  |
| 4.1      | The Direct Linear Transformation (DLT) algorithm             | 88         |
| 4.2      | Different cost functions                                     | 93         |
| 4.3      | Statistical cost functions and Maximum Likelihood estimation | 102        |
| 4.4      | Transformation invariance and normalization                  | 104        |
| 4.5      | Iterative minimization methods                               | 110        |
| 4.6      | Experimental comparison of the algorithms                    | 115        |
| 4.7      | Robust estimation                                            | 116        |
| 4.8      | Automatic computation of a homography                        | 123        |
| 4.9      | Closure                                                      | 127        |
| <b>5</b> | <b>Algorithm Evaluation and Error Analysis</b>               | <b>132</b> |
| 5.1      | Bounds on performance                                        | 132        |
| 5.2      | Covariance of the estimated transformation                   | 138        |
| 5.3      | Monte Carlo estimation of covariance                         | 149        |
| 5.4      | Closure                                                      | 150        |
|          | <b>PART I: Camera Geometry and Single View Geometry</b>      | <b>151</b> |
|          | <i>Outline</i>                                               | 152        |
| <b>6</b> | <b>Camera Models</b>                                         | <b>153</b> |
| 6.1      | Finite cameras                                               | 153        |
| 6.2      | The projective camera                                        | 158        |
| 6.3      | Cameras at infinity                                          | 166        |
| 6.4      | Other camera models                                          | 174        |
| 6.5      | Closure                                                      | 176        |
| <b>7</b> | <b>Computation of the Camera Matrix P</b>                    | <b>178</b> |
| 7.1      | Basic equations                                              | 178        |
| 7.2      | Geometric error                                              | 180        |
| 7.3      | Restricted camera estimation                                 | 184        |
| 7.4      | Radial distortion                                            | 189        |
| 7.5      | Closure                                                      | 193        |
| <b>8</b> | <b>More Single View Geometry</b>                             | <b>195</b> |
| 8.1      | Action of a projective camera on planes, lines, and conics   | 195        |
| 8.2      | Images of smooth surfaces                                    | 200        |
| 8.3      | Action of a projective camera on quadrics                    | 201        |
| 8.4      | The importance of the camera centre                          | 202        |
| 8.5      | Camera calibration and the image of the absolute conic       | 208        |

|                                   |                                                             |     |
|-----------------------------------|-------------------------------------------------------------|-----|
| 8.6                               | Vanishing points and vanishing lines                        | 213 |
| 8.7                               | Affine 3D measurements and reconstruction                   | 220 |
| 8.8                               | Determining camera calibration $K$ from a single view       | 223 |
| 8.9                               | Single view reconstruction                                  | 229 |
| 8.10                              | The calibrating conic                                       | 231 |
| 8.11                              | Closure                                                     | 233 |
| <b>PART II: Two-View Geometry</b> |                                                             | 237 |
|                                   | <i>Outline</i>                                              | 238 |
| <b>9</b>                          | <b>Epipolar Geometry and the Fundamental Matrix</b>         | 239 |
| 9.1                               | Epipolar geometry                                           | 239 |
| 9.2                               | The fundamental matrix $F$                                  | 241 |
| 9.3                               | Fundamental matrices arising from special motions           | 247 |
| 9.4                               | Geometric representation of the fundamental matrix          | 250 |
| 9.5                               | Retrieving the camera matrices                              | 253 |
| 9.6                               | The essential matrix                                        | 257 |
| 9.7                               | Closure                                                     | 259 |
| <b>10</b>                         | <b>3D Reconstruction of Cameras and Structure</b>           | 262 |
| 10.1                              | Outline of reconstruction method                            | 262 |
| 10.2                              | Reconstruction ambiguity                                    | 264 |
| 10.3                              | The projective reconstruction theorem                       | 266 |
| 10.4                              | Stratified reconstruction                                   | 267 |
| 10.5                              | Direct reconstruction – using ground truth                  | 275 |
| 10.6                              | Closure                                                     | 276 |
| <b>11</b>                         | <b>Computation of the Fundamental Matrix <math>F</math></b> | 279 |
| 11.1                              | Basic equations                                             | 279 |
| 11.2                              | The normalized 8-point algorithm                            | 281 |
| 11.3                              | The algebraic minimization algorithm                        | 282 |
| 11.4                              | Geometric distance                                          | 284 |
| 11.5                              | Experimental evaluation of the algorithms                   | 288 |
| 11.6                              | Automatic computation of $F$                                | 290 |
| 11.7                              | Special cases of $F$ -computation                           | 293 |
| 11.8                              | Correspondence of other entities                            | 294 |
| 11.9                              | Degeneracies                                                | 295 |
| 11.10                             | A geometric interpretation of $F$ -computation              | 297 |
| 11.11                             | The envelope of epipolar lines                              | 298 |
| 11.12                             | Image rectification                                         | 302 |
| 11.13                             | Closure                                                     | 308 |
| <b>12</b>                         | <b>Structure Computation</b>                                | 310 |
| 12.1                              | Problem statement                                           | 310 |
| 12.2                              | Linear triangulation methods                                | 312 |
| 12.3                              | Geometric error cost function                               | 313 |
| 12.4                              | Sampson approximation (first-order geometric correction)    | 314 |

|                  |                                                                    |            |
|------------------|--------------------------------------------------------------------|------------|
| 12.5             | An optimal solution                                                | 315        |
| 12.6             | Probability distribution of the estimated 3D point                 | 321        |
| 12.7             | Line reconstruction                                                | 321        |
| 12.8             | Closure                                                            | 323        |
| <b>13</b>        | <b>Scene planes and homographies</b>                               | <b>325</b> |
| 13.1             | Homographies given the plane and vice versa                        | 326        |
| 13.2             | Plane induced homographies given $F$ and image correspondences     | 329        |
| 13.3             | Computing $F$ given the homography induced by a plane              | 334        |
| 13.4             | The infinite homography $H_\infty$                                 | 338        |
| 13.5             | Closure                                                            | 340        |
| <b>14</b>        | <b>Affine Epipolar Geometry</b>                                    | <b>344</b> |
| 14.1             | Affine epipolar geometry                                           | 344        |
| 14.2             | The affine fundamental matrix                                      | 345        |
| 14.3             | Estimating $F_A$ from image point correspondences                  | 347        |
| 14.4             | Triangulation                                                      | 353        |
| 14.5             | Affine reconstruction                                              | 353        |
| 14.6             | Necker reversal and the bas-relief ambiguity                       | 355        |
| 14.7             | Computing the motion                                               | 357        |
| 14.8             | Closure                                                            | 360        |
| <b>PART III:</b> | <b>Three-View Geometry</b>                                         | <b>363</b> |
|                  | <i>Outline</i>                                                     | 364        |
| <b>15</b>        | <b>The Trifocal Tensor</b>                                         | <b>365</b> |
| 15.1             | The geometric basis for the trifocal tensor                        | 365        |
| 15.2             | The trifocal tensor and tensor notation                            | 376        |
| 15.3             | Transfer                                                           | 379        |
| 15.4             | The fundamental matrices for three views                           | 383        |
| 15.5             | Closure                                                            | 387        |
| <b>16</b>        | <b>Computation of the Trifocal Tensor <math>\mathcal{T}</math></b> | <b>391</b> |
| 16.1             | Basic equations                                                    | 391        |
| 16.2             | The normalized linear algorithm                                    | 393        |
| 16.3             | The algebraic minimization algorithm                               | 395        |
| 16.4             | Geometric distance                                                 | 396        |
| 16.5             | Experimental evaluation of the algorithms                          | 399        |
| 16.6             | Automatic computation of $\mathcal{T}$                             | 400        |
| 16.7             | Special cases of $\mathcal{T}$ -computation                        | 404        |
| 16.8             | Closure                                                            | 406        |
| <b>PART IV:</b>  | <b>N-View Geometry</b>                                             | <b>409</b> |
|                  | <i>Outline</i>                                                     | 410        |
| <b>17</b>        | <b>N-Linearities and Multiple View Tensors</b>                     | <b>411</b> |
| 17.1             | Bilinear relations                                                 | 411        |
| 17.2             | Trilinear relations                                                | 414        |

|           |                                                     |     |
|-----------|-----------------------------------------------------|-----|
| 17.3      | Quadrilinear relations                              | 418 |
| 17.4      | Intersections of four planes                        | 421 |
| 17.5      | Counting arguments                                  | 422 |
| 17.6      | Number of independent equations                     | 428 |
| 17.7      | Choosing equations                                  | 431 |
| 17.8      | Closure                                             | 432 |
| <b>18</b> | <b><i>N</i>-View Computational Methods</b>          | 434 |
| 18.1      | Projective reconstruction – bundle adjustment       | 434 |
| 18.2      | Affine reconstruction – the factorization algorithm | 436 |
| 18.3      | Non-rigid factorization                             | 440 |
| 18.4      | Projective factorization                            | 444 |
| 18.5      | Projective reconstruction using planes              | 447 |
| 18.6      | Reconstruction from sequences                       | 452 |
| 18.7      | Closure                                             | 456 |
| <b>19</b> | <b>Auto-Calibration</b>                             | 458 |
| 19.1      | Introduction                                        | 458 |
| 19.2      | Algebraic framework and problem statement           | 459 |
| 19.3      | Calibration using the absolute dual quadric         | 462 |
| 19.4      | The Kruppa equations                                | 469 |
| 19.5      | A stratified solution                               | 473 |
| 19.6      | Calibration from rotating cameras                   | 481 |
| 19.7      | Auto-calibration from planes                        | 485 |
| 19.8      | Planar motion                                       | 486 |
| 19.9      | Single axis rotation – turntable motion             | 490 |
| 19.10     | Auto-calibration of a stereo rig                    | 493 |
| 19.11     | Closure                                             | 497 |
| <b>20</b> | <b>Duality</b>                                      | 502 |
| 20.1      | Carlsson–Weinshall duality                          | 502 |
| 20.2      | Reduced reconstruction                              | 508 |
| 20.3      | Closure                                             | 513 |
| <b>21</b> | <b>Cheirality</b>                                   | 515 |
| 21.1      | Quasi-affine transformations                        | 515 |
| 21.2      | Front and back of a camera                          | 518 |
| 21.3      | Three-dimensional point sets                        | 519 |
| 21.4      | Obtaining a quasi-affine reconstruction             | 520 |
| 21.5      | Effect of transformations on cheirality             | 521 |
| 21.6      | Orientation                                         | 523 |
| 21.7      | The cheiral inequalities                            | 525 |
| 21.8      | Which points are visible in a third view            | 528 |
| 21.9      | Which points are in front of which                  | 530 |
| 21.10     | Closure                                             | 531 |

|                                                                 |     |
|-----------------------------------------------------------------|-----|
| <b>22 Degenerate Configurations</b>                             | 533 |
| 22.1 Camera resectioning                                        | 533 |
| 22.2 Degeneracies in two views                                  | 539 |
| 22.3 Carlsson–Weinshall duality                                 | 546 |
| 22.4 Three-view critical configurations                         | 553 |
| 22.5 Closure                                                    | 558 |
| <br>                                                            |     |
| <b>PART V : Appendices</b>                                      | 561 |
| <i>Appendix 1</i> Tensor Notation                               | 562 |
| <i>Appendix 2</i> Gaussian (Normal) and $\chi^2$ Distributions  | 565 |
| <i>Appendix 3</i> Parameter Estimation                          | 568 |
| <i>Appendix 4</i> Matrix Properties and Decompositions          | 578 |
| <i>Appendix 5</i> Least-squares Minimization                    | 588 |
| <i>Appendix 6</i> Iterative Estimation Methods                  | 597 |
| <i>Appendix 7</i> Some Special Plane Projective Transformations | 628 |
| <i>Bibliography</i>                                             | 634 |
| <i>Index</i>                                                    | 646 |

---

## Foreword

*By Olivier Faugeras*

Making a computer see was something that leading experts in the field of Artificial Intelligence thought to be at the level of difficulty of a summer student's project back in the sixties. Forty years later the task is still unsolved and seems formidable. A whole field, called Computer Vision, has emerged as a discipline in itself with strong connections to mathematics and computer science and looser connections to physics, the psychology of perception and the neuro sciences.

One of the likely reasons for this half-failure is the fact that researchers had overlooked the fact, perhaps because of this plague called naive introspection, that perception in general and visual perception in particular are far more complex in animals and humans than was initially thought. There is of course no reason why we should pattern Computer Vision algorithms after biological ones, but the fact of the matter is that

- (i) the way biological vision works is still largely unknown and therefore hard to emulate on computers, and
- (ii) attempts to ignore biological vision and reinvent a sort of silicon-based vision have not been so successful as initially expected.

Despite these negative remarks, Computer Vision researchers have obtained some outstanding successes, both practical and theoretical.

On the side of practice, and to single out one example, the possibility of guiding vehicles such as cars and trucks on regular roads or on rough terrain using computer vision technology was demonstrated many years ago in Europe, the USA and Japan. This requires capabilities for real-time three-dimensional dynamic scene analysis which are quite elaborate. Today, car manufacturers are slowly incorporating some of these functions in their products.

On the theoretical side some remarkable progress has been achieved in the area of what one could call geometric Computer Vision. This includes the description of the way the appearance of objects changes when viewed from different viewpoints as a function of the objects' shape and the cameras parameters. This endeavour would not have been achieved without the use of fairly sophisticated mathematical techniques encompassing many areas of geometry, ancient and novel. This book deals in particular with the intricate and beautiful geometric relations that exist between the images of objects in the world. These relations are important to analyze for their own sake because

this is one of the goals of science to provide explanations for appearances; they are also important to analyze because of the range of applications their understanding opens up.

The book has been written by two pioneers and leading experts in geometric Computer Vision. They have succeeded in what was something of a challenge, namely to convey in a simple and easily accessible way the mathematics that is necessary for understanding the underlying geometric concepts, to be quite exhaustive in the coverage of the results that have been obtained by them and other researchers worldwide, to analyze the interplay between the geometry and the fact that the image measurements are necessarily noisy, to express many of these theoretical results in algorithmic form so that they can readily be transformed into computer code, and to present many real examples that illustrate the concepts and show the range of applicability of the theory.

Returning to the original holy grail of making a computer see we may wonder whether this kind of work is a step in the right direction. I must leave the readers of the book to answer this question, and be content with saying that no designer of systems using cameras hooked to computers that will be built in the foreseeable future can ignore this work. This is perhaps a step in the direction of defining what it means for a computer to see.

---

## Preface

Over the past decade there has been a rapid development in the understanding and modelling of the geometry of multiple views in computer vision. The theory and practice have now reached a level of maturity where excellent results can be achieved for problems that were certainly unsolved a decade ago, and often thought unsolvable. These tasks and algorithms include:

- Given two images, and no other information, compute matches between the images, and the 3D position of the points that generate these matches and the cameras that generate the images.
- Given three images, and no other information, similarly compute the matches between images of points and lines, and the position in 3D of these points and lines and the cameras.
- Compute the epipolar geometry of a stereo rig, and trifocal geometry of a trinocular rig, without requiring a calibration object.
- Compute the internal calibration of a camera from a sequence of images of natural scenes (i.e. calibration “on the fly”).

The distinctive flavour of these algorithms is that they are *uncalibrated* — it is not necessary to know or first need to compute the camera internal parameters (such as the focal length).

Underpinning these algorithms is a new and more complete theoretical understanding of the geometry of multiple uncalibrated views: the number of parameters involved, the constraints between points and lines imaged in the views; and the retrieval of cameras and 3-space points from image correspondences. For example, to determine the epipolar geometry of a stereo rig requires specifying only seven parameters, the camera calibration is not required. These parameters are determined from the correspondence of seven or more image point correspondences. Contrast this uncalibrated route, with the previous calibrated route of a decade ago: each camera would first be calibrated from the image of a carefully engineered calibration object with known geometry. The calibration involves determining 11 parameters for each camera. The epipolar geometry would then have been computed from these two sets of 11 parameters.

This example illustrates the importance of the uncalibrated (projective) approach — using the appropriate representation of the geometry makes explicit the parameters

that are required at each stage of a computation. This avoids computing parameters that have no effect on the final result, and results in simpler algorithms. It is also worth correcting a possible misconception. In the uncalibrated framework, entities (for instance point positions in 3-space) are often recovered to within a precisely defined ambiguity. This ambiguity does not mean that the points are poorly estimated.

More practically, it is often not possible to calibrate cameras once-and-for-all; for instance where cameras are moved (on a mobile vehicle) or internal parameters are changed (a surveillance camera with zoom). Furthermore, calibration information is simply not available in some circumstances. Imagine computing the motion of a camera from a video sequence, or building a virtual reality model from archive film footage where both motion and internal calibration information are unknown.

The achievements in multiple view geometry have been possible because of developments in our theoretical understanding, but also because of improvements in estimating mathematical objects from images. The first improvement has been an attention to the error that should be minimized in over-determined systems – whether it be algebraic, geometric or statistical. The second improvement has been the use of robust estimation algorithms (such as RANSAC), so that the estimate is unaffected by “outliers” in the data. Also these techniques have generated powerful search and matching algorithms.

Many of the problems of reconstruction have now reached a level where we may claim that they are solved. Such problems include:

- (i) Estimation of the multifocal tensors from image point correspondences, particularly the fundamental matrix and trifocal tensors (the quadrifocal tensor having not received so much attention).
- (ii) Extraction of the camera matrices from these tensors, and subsequent projective reconstruction from two, three and four views.

Other significant successes have been achieved, though there may be more to learn about these problems. Examples include:

- (i) Application of bundle adjustment to solve more general reconstruction problems.
- (ii) Metric (Euclidean) reconstruction given minimal assumptions on the camera matrices.
- (iii) Automatic detection of correspondences in image sequences, and elimination of outliers and false matches using the multifocal tensor relationships.

**Roadplan.** The book is divided into six parts and there are seven short appendices. Each part introduces a new geometric relation: the homography for background, the camera matrix for single view, the fundamental matrix for two views, the trifocal tensor for three views, and the quadrifocal tensor for four views. In each case there is a chapter describing the relation, its properties and applications, and a companion chapter describing algorithms for its estimation from image measurements. The estimation algorithms described range from cheap, simple, approaches through to the optimal algorithms which are currently believed to be the best available.

- Part 0: Background.** This part is more tutorial than the others. It introduces the central ideas in the projective geometry of 2-space and 3-space (for example ideal points, and the absolute conic); how this geometry may be represented, manipulated, and estimated; and how the geometry relates to various objectives in computer vision such as rectifying images of planes to remove perspective distortion.
- Part 1: Single view geometry.** Here the various cameras that model the perspective projection from 3-space to an image are defined and their anatomy explored. Their estimation using traditional techniques of calibration objects is described, as well as camera calibration from vanishing points and vanishing lines.
- Part 2: Two view geometry.** This part describes the epipolar geometry of two cameras, projective reconstruction from image point correspondences, methods of resolving the projective ambiguity, optimal triangulation, transfer between views via planes.
- Part 3: Three view geometry.** Here the trifocal geometry of three cameras is described, including transfer of a point correspondence from two views to a third, and similarly transfer for a line correspondence; computation of the geometry from point and line correspondences, retrieval of the camera matrices.
- Part 4: N-views.** This part has two purposes. First, it extends three view geometry to four views (a minor extension) and describes estimation methods applicable to N-views, such as the factorization algorithm of Tomasi and Kanade for computing structure and motion simultaneously from multiple images. Second, it covers themes that have been touched on in earlier chapters, but can be understood more fully and uniformly by emphasising their commonality. Examples include deriving multi-linear view constraints on correspondences, auto-calibration, and ambiguous solutions.
- Appendices.** These describe further background material on tensors, statistics, parameter estimation, linear and matrix algebra, iterative estimation, the solution of sparse matrix systems, and special projective transformations.

**Acknowledgements.** We have benefited enormously from ideas and discussions with our colleagues: Paul Beardsley, Stefan Carlsson, Olivier Faugeras, Andrew Fitzgibbon, Jitendra Malik, Steve Maybank, Amnon Shashua, Phil Torr, Bill Triggs.

If there are only a countable number of errors in this book then it is due to Antonio Criminisi, David Liebowitz and Frederik Schaffalitzky who have with great energy and devotion read most of it, and made numerous suggestions for improvements. Similarly both Peter Sturm and Bill Triggs have suggested many improvements to various chapters. We are grateful to other colleagues who have read individual chapters: David Capel, Lourdes de Agapito Vicente, Bob Kaucic, Steve Maybank, Peter Tu.

We are particularly grateful to those who have provided multiple figures: Paul Beardsley, Antonio Criminisi, Andrew Fitzgibbon, David Liebowitz, and Larry Shapiro; and for individual figures from: Martin Armstrong, David Capel, Lourdes de Agapito Vicente, Eric Hayman, Phil Pritchett, Luc Robert, Cordelia Schmid, and others who are explicitly acknowledged in figure captions.

At Cambridge University Press we thank David Tranah for his constant source of advice and patience, and Michael Behrend for excellent copy editing.

A small number of minor errors have been corrected in the reprinted editions, and we thank the following readers for pointing these out: Luis Baumela, Niclas Borlin, Mike Brooks, Jun ho. Choi, Wojciech Chojnacki, Carlo Colombo, Nicolas Dano, Andrew Fitzgibbon, Bogdan Georgescu, Fredrik Kahl, Bob Kaucic, Jae-Hak Kim, Hansung Lee, Dennis Maier, Karsten Muelhmann, David Nister, Andreas Olsson, Stéphane Paris, Frederik Schaffalitzky, Bill Severson, Pedro Lopez de Teruel Alcolea, Bernard Thiesse, Ken Thornton, Magdalena Urbanek, Gergely Vass, Eugene Vendrovsky, Sui Wei, and Tomáš Werner.

**The second edition.** This new paperback edition has been expanded to include some of the developments since the original version of July 2000. For example, the book now covers the discovery of a closed form factorization solution in the projective case when a plane is visible in the scene, and the extension of affine factorization to non-rigid scenes. We have also extended the discussion of single view geometry (chapter 8) and three view geometry (chapter 15), and added an appendix on parameter estimation.

In preparing this second edition we are very grateful to colleagues who have made suggestion for improvements and additions. These include Marc Pollefeys, Bill Triggs and in particular Tomáš Werner who provided excellent and comprehensive comments. We also thank Antonio Criminisi, Andrew Fitzgibbon, Rob Fergus, David Liebowitz, and particularly Josef Šivic, for proof reading and very helpful comments on parts of the new material. As always we are grateful to David Tranah of CUP.

The figures appearing in this book can be downloaded from

<http://www.robots.ox.ac.uk/~vgg/hzbook.html>

This site also includes Matlab code for several of the algorithms, and lists the errata of earlier printings.

I am never forget the day my first book is published. Every chapter I stole from somewhere else. Index I copy from old Vladivostok telephone directory. This book, this book was sensational!

Excerpts from “Nikolai Ivanovich Lobachevsky” by Tom Lehrer.

---

## More Single View Geometry

Chapter 6 introduced the projection matrix as the model for the action of a camera on points. This chapter describes the link between other 3D entities and their images under perspective projection. These entities include planes, lines, conics and quadrics; and we develop their forward and back-projection properties.

The camera is dissected further, and reduced to its centre point and image plane. Two properties are established: images acquired by cameras with the same centre are related by a plane projective transformation; and images of entities on the plane at infinity,  $\pi_\infty$ , do not depend on camera position, only on camera rotation and internal parameters,  $K$ .

The images of entities (points, lines, conics) on  $\pi_\infty$  are of particular importance. It will be seen that the image of a point on  $\pi_\infty$  is a vanishing point, and the image of a line on  $\pi_\infty$  a vanishing line; their images depend on both  $K$  and camera rotation. However, the image of the absolute conic,  $\omega$ , depends only on  $K$ ; it is unaffected by the camera's rotation. The conic  $\omega$  is intimately connected with camera calibration,  $K$ , and the relation  $\omega = (KK^T)^{-1}$  is established. It follows that  $\omega$  defines the angle between rays back-projected from image points.

These properties enable camera relative rotation to be computed from vanishing points independently of camera position. Further, since  $K$  enables the angle between rays to be computed from image points, in turn  $K$  may be computed from the known angle between rays. In particular  $K$  may be determined from vanishing points corresponding to orthogonal scene directions. This means that a camera can be calibrated from scene features, without requiring known world coordinates.

A final geometric entity introduced in this chapter is the calibrating conic, which enables a geometric visualization of  $K$ .

### 8.1 Action of a projective camera on planes, lines, and conics

In this section (and indeed in most of this book) it is only the  $3 \times 4$  *form* and rank of the camera projection matrix  $P$  that is important in determining its action. The particular properties and relations of its elements are often not relevant.

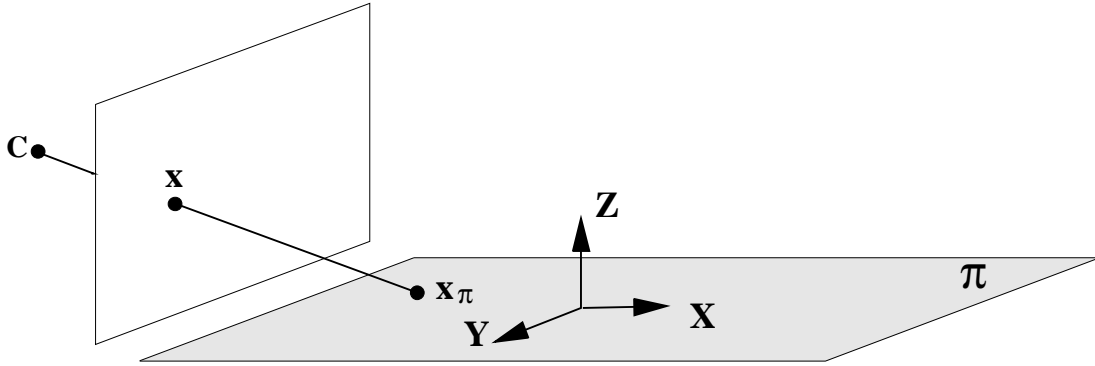


Fig. 8.1. **Perspective image of points on a plane.** The  $XY$ -plane of the world coordinate frame is aligned with the plane  $\pi$ . Points on the image and scene planes are related by a plane projective transformation.

### 8.1.1 On planes

The point imaging equation  $\mathbf{x} = \mathbf{P}\mathbf{X}$  is a map from a point in a world coordinate frame, to a point in image coordinates. We have the freedom to choose the world coordinate frame. Suppose it is chosen such that the  $XY$ -plane corresponds to a plane  $\pi$  in the scene, so that points on the scene plane have zero  $Z$ -coordinate as shown in figure 8.1 (it is assumed that the camera centre does not lie on the scene plane). Then, if the columns of  $\mathbf{P}$  are denoted as  $\mathbf{p}_i$ , the image of a point on  $\pi$  is given by

$$\mathbf{x} = \mathbf{P}\mathbf{X} = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \mathbf{p}_3 & \mathbf{p}_4 \end{bmatrix} \begin{pmatrix} X \\ Y \\ 0 \\ 1 \end{pmatrix} = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \mathbf{p}_4 \end{bmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}.$$

So that the map between points  $\mathbf{x}_\pi = (X, Y, 1)^\top$  on  $\pi$  and their image  $\mathbf{x}$  is a general planar homography (a plane to plane projective transformation):  $\mathbf{x} = \mathbf{H}\mathbf{x}_\pi$ , with  $\mathbf{H}$  a  $3 \times 3$  matrix of rank 3. This shows that:

- *The most general transformation that can occur between a scene plane and an image plane under perspective imaging is a plane projective transformation.*

If the camera is affine, then a similar derivation shows that the scene and image planes are related by an affine transformation.

**Example 8.1.** For a calibrated camera (6.8–p156)  $\mathbf{P} = \mathbf{K}[\mathbf{R} \mid \mathbf{t}]$ , the homography between a world plane at  $Z = 0$  and the image is

$$\mathbf{H} = \mathbf{K} [\mathbf{r}_1, \mathbf{r}_2, \mathbf{t}] \quad (8.1)$$

where  $\mathbf{r}_i$  are the columns of  $\mathbf{R}$ . △

### 8.1.2 On lines

**Forward projection.** A line in 3-space projects to a line in the image. This is easily seen geometrically – the line and camera centre define a plane, and the image is the intersection of this plane with the image plane (figure 8.2) – and is proved algebraically

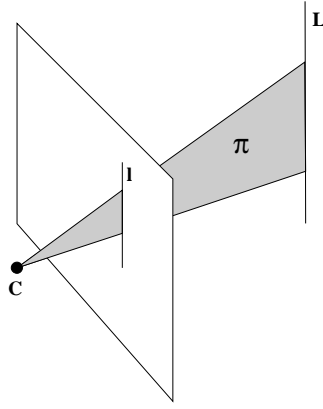


Fig. 8.2. **Line projection.** A line  $L$  in 3-space is imaged as a line  $l$  by a perspective camera. The image line  $l$  is the intersection of the plane  $\pi$ , defined by  $L$  and the camera centre  $C$ , with the image plane. Conversely an image line  $l$  back-projects to a plane  $\pi$  in 3-space. The plane is the “pull-back” of the line.

by noting that if  $A, B$  are points in 3-space, and  $a, b$  their images under  $P$ , then a point  $X(\mu) = A + \mu B$  on a line which is the join of  $A, B$  in 3-space projects to a point

$$\begin{aligned} \mathbf{x}(\mu) &= P(A + \mu B) = PA + \mu PB \\ &= \mathbf{a} + \mu \mathbf{b} \end{aligned}$$

which is on the line joining  $a$  and  $b$ .

**Back-projection of lines.** The set of points in space which map to a line in the image is a plane in space defined by the camera centre and image line, as shown in figure 8.2. Algebraically,

**Result 8.2.** *The set of points in space mapping to a line  $l$  via the camera matrix  $P$  is the plane  $P^T l$ .*

**Proof.** A point  $x$  lies on  $l$  if and only if  $x^T l = 0$ . A space point  $X$  maps to a point  $PX$ , which lies on the line if and only if  $X^T P^T l = 0$ . Thus, if  $P^T l$  is taken to represent a plane, then  $X$  lies on this plane if and only if  $X$  maps to a point on the line  $l$ . In other words,  $P^T l$  is the back-projection of the line  $l$ .  $\square$

Geometrically there is a *star* (two-parameter family) of planes through the camera centre, and the three rows of the projection matrix  $P^{i^T}$  (6.12–p159) are a basis for this star. The plane  $P^T l$  is a linear combination of this basis corresponding to the element of the star containing the camera centre and the line  $l$ . For example, if  $l = (0, 1, 0)^T$  then the plane is  $P^2$ , and is the back projection of the image  $x$ -axis.

**Plücker line representation.** *Understanding this material on Plücker line mapping is not required for following the rest of the book.*

We now turn to forward projection of lines. If a line in 3-space is represented by Plücker coordinates then its image can be expressed as a linear map on these coordinates. We will develop this map for both the  $4 \times 4$  matrix and 6-vector line representations.

**Result 8.3.** Under the camera mapping  $P$ , a line in 3-space represented as a Plücker matrix  $L$ , as defined in (3.8–p70), is imaged as the line  $l$  where

$$[l]_{\times} = PLP^T. \quad (8.2)$$

where the notation  $[l]_{\times}$  is defined in (A4.5–p581).

**Proof.** Suppose as above that  $\mathbf{a} = PA$ ,  $\mathbf{b} = PB$ . The Plücker matrix  $L$  for the line through  $A, B$  in 3-space is  $L = AB^T - BA^T$ . Then the matrix  $M = PLP^T = \mathbf{a}\mathbf{b}^T - \mathbf{b}\mathbf{a}^T$  is  $3 \times 3$  and antisymmetric, with null-space  $\mathbf{a} \times \mathbf{b}$ . Consequently,  $M = [\mathbf{a} \times \mathbf{b}]_{\times}$ , and since the line through the image points is given by  $l = \mathbf{a} \times \mathbf{b}$ , this completes the proof.  $\square$

It is clear from the form of (8.2) that there is a linear relation between the image line coordinates  $l_i$  and the world line coordinates  $L_{jk}$ , but that this relation is quadratic in the elements of the point projection matrix  $P$ . Thus, (8.2) may be rearranged such that the map between the Plücker line coordinates,  $\mathcal{L}$  (a 6-vector), and the image line coordinates  $l$  (a 3-vector) is represented by a single  $3 \times 6$  matrix. It can be shown that

**Definition 8.4.** The *line projection matrix*  $\mathcal{P}$  is the  $3 \times 6$  matrix of rank 3 given by

$$\mathcal{P} = \begin{bmatrix} \mathbf{P}^2 \wedge \mathbf{P}^3 \\ \mathbf{P}^3 \wedge \mathbf{P}^1 \\ \mathbf{P}^1 \wedge \mathbf{P}^2 \end{bmatrix} \quad (8.3)$$

where  $\mathbf{P}^{iT}$  are the rows of the point camera matrix  $P$ , and  $\mathbf{P}^i \wedge \mathbf{P}^j$  are the Plücker line coordinates of the intersection of the planes  $\mathbf{P}^i$  and  $\mathbf{P}^j$ .

Then the forward line projection is given by

**Result 8.5.** Under the line projection matrix  $\mathcal{P}$ , a line in  $\mathbb{P}^3$  represented by Plücker line coordinates  $\mathcal{L}$ , as defined in (3.11–p72), is mapped to the image line

$$l = \mathcal{P}\mathcal{L} = \begin{bmatrix} (\mathbf{P}^2 \wedge \mathbf{P}^3 | \mathcal{L}) \\ (\mathbf{P}^3 \wedge \mathbf{P}^1 | \mathcal{L}) \\ (\mathbf{P}^1 \wedge \mathbf{P}^2 | \mathcal{L}) \end{bmatrix} \quad (8.4)$$

where the product  $(\mathcal{L} | \hat{\mathcal{L}})$  is defined in (3.13–p72).

**Proof.** Suppose the line in 3-space is the join of the points  $A$  and  $B$ , and these project to  $\mathbf{a} = PA$ ,  $\mathbf{b} = PB$  respectively. Then the image line  $l = \mathbf{a} \times \mathbf{b} = (PA) \times (PB)$ . Consider the first element

$$\begin{aligned} l_1 &= (\mathbf{P}^{2T}\mathbf{A})(\mathbf{P}^{3T}\mathbf{B}) - (\mathbf{P}^{2T}\mathbf{B})(\mathbf{P}^{3T}\mathbf{A}) \\ &= (\mathbf{P}^2 \wedge \mathbf{P}^3 | \mathcal{L}) \end{aligned}$$

where the second equality follows from (3.14–p73). The other components follow in a similar manner.  $\square$

The line projection matrix  $\mathcal{P}$  plays the same role for lines as  $P$  does for points. The rows of  $\mathcal{P}$  may be interpreted geometrically as *lines*, in a similar manner to the interpretation of the rows of the point camera matrix  $P$  as *planes* given in section 6.2.1(p158). The rows  $P^{i\top}$  of  $P$  are the principal plane and axis planes of the camera. The rows of  $\mathcal{P}$  are the lines of intersection of pairs of these camera planes. For example, the first row of  $\mathcal{P}$  is  $P^2 \wedge P^3$ , and this is the 6-vector Plücker line representation of the line of intersection of the  $y = 0$  axis plane,  $P^2$ , and the principal plane,  $P^3$ . The three lines corresponding to the three rows of  $\mathcal{P}$  intersect at the camera centre. Consider lines  $\mathcal{L}$  in 3-space for which  $\mathcal{P}\mathcal{L} = \mathbf{0}$ . These lines are in the null-space of  $\mathcal{P}$ . Since each row of  $\mathcal{P}$  is a line, and from result 3.5(p72) the product  $(\mathcal{L}_1|\mathcal{L}_2) = 0$  if two lines intersect, it follows that  $\mathcal{L}$  intersects each of the lines represented by the rows of  $\mathcal{P}$ . These lines are the intersection of the camera planes, and the only point on all 3 camera planes is the camera centre. Thus we have

- The lines  $\mathcal{L}$  in  $\mathbb{P}^3$  for which  $\mathcal{P}\mathcal{L} = \mathbf{0}$  pass through the camera centre.

The  $3 \times 6$  matrix  $\mathcal{P}$  has a 3-dimensional null-space. Allowing for the homogeneous scale factor, this null-space is a two-parameter family of lines containing the camera centre. This is to be expected since there is a star (two parameter family) of lines in  $\mathbb{P}^3$  concurrent with a point.

### 8.1.3 On conics

**Back-projection of conics.** A conic  $C$  back-projects to a cone. A cone is a degenerate quadric, i.e. the  $4 \times 4$  matrix representing the quadric does not have full rank. The cone vertex, in this case the camera centre, is the null-vector of the quadric matrix.

**Result 8.6.** Under the camera  $P$  the conic  $C$  back-projects to the cone

$$Q_{co} = P^\top C P.$$

**Proof.** A point  $\mathbf{x}$  lies on  $C$  if and only if  $\mathbf{x}^\top C \mathbf{x} = 0$ . A space point  $\mathbf{X}$  maps to a point  $P\mathbf{X}$ , which lies on the conic if and only if  $\mathbf{X}^\top P^\top C P \mathbf{X} = 0$ . Thus, if  $Q_{co} = P^\top C P$  is taken to represent a quadric, then  $\mathbf{X}$  lies on this quadric if and only if  $\mathbf{X}$  maps to a point on the conic  $C$ . In other words,  $Q_{co}$  is the back-projection of the conic  $C$ .  $\square$

Note the camera centre  $C$  is the vertex of the degenerate quadric since  $Q_{co} C = P^\top C (PC) = \mathbf{0}$ .

**Example 8.7.** Suppose that  $P = K[I \mid \mathbf{0}]$ ; then the conic  $C$  back-projects to the cone

$$Q_{co} = \begin{bmatrix} K^\top \\ \mathbf{0}^\top \end{bmatrix} C [K \mid \mathbf{0}] = \begin{bmatrix} K^\top C K & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{bmatrix}.$$

The matrix  $Q_{co}$  has rank 3. Its null-vector is the camera centre  $C = (0, 0, 0, 1)^\top$ .  $\triangle$

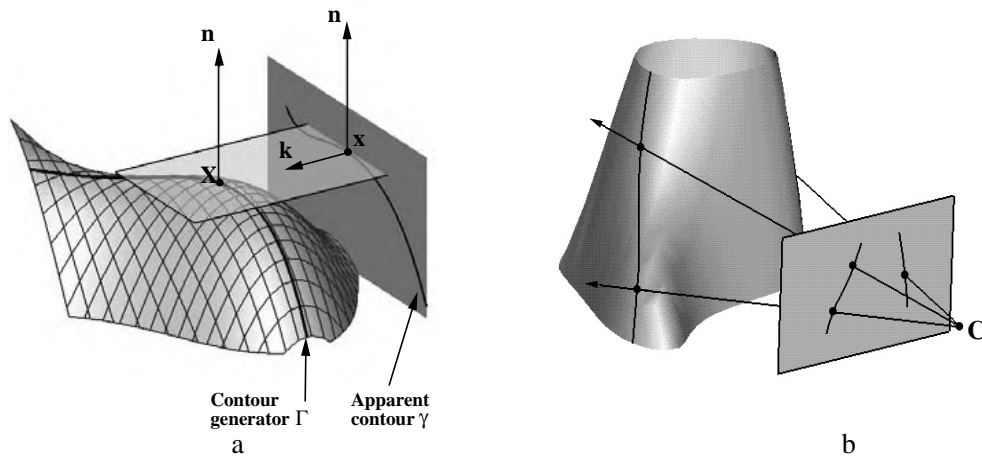


Fig. 8.3. **Contour generator and apparent contour.** (a) for parallel projection; (b) for central projection. The ray from the camera centre through  $\mathbf{x}$  is tangent to the surface at  $\mathbf{X}$ . The set of such tangent points  $\mathbf{X}$  defines the contour generator, and their image defines the apparent contour. In general the contour generator is a space curve. Figure courtesy of Roberto Cipolla and Peter Giblin.

## 8.2 Images of smooth surfaces

The image outline of a smooth surface  $S$  results from surface points at which the imaging rays are *tangent* to the surface, as shown in figure 8.3. Similarly, *lines* tangent to the outline back-project to planes which are *tangent planes* to the surface.

**Definition 8.8.** The *contour generator*  $\Gamma$  is the set of points  $\mathbf{X}$  on  $S$  at which rays are tangent to the surface. The corresponding image *apparent contour*  $\gamma$  is the set of points  $\mathbf{x}$  which are the image of  $\mathbf{X}$ , i.e.  $\gamma$  is the image of  $\Gamma$ .

The apparent contour is also called the “outline” and “profile”. If the surface is viewed in the direction of  $\mathbf{X}$  from the camera centre, then the surface appears to fold, or to have a boundary or occluding contour.

It is evident that the contour generator  $\Gamma$  depends only on the relative position of the camera centre and surface, not on the image plane. However, the apparent contour  $\gamma$  is defined by the intersection of the image plane with the rays to the contour generator, and so does depend on the position of the image plane.

In the case of parallel projection with direction  $\mathbf{k}$ , consider all the rays parallel to  $\mathbf{k}$  which are tangent to  $S$ , see figure 8.3a. These rays form a “cylinder” of tangent rays, and the curve along which this cylinder is tangent to  $S$  is the contour generator  $\Gamma$ . The curve in which the cylinder meets the image plane is the apparent contour  $\gamma$ . Note that both  $\Gamma$  and  $\gamma$  depend in an essential way on  $\mathbf{k}$ . The set  $\Gamma$  slips over the surface as the direction of  $\mathbf{k}$  changes. For example, with  $S$  a sphere,  $\Gamma$  is the great circle orthogonal to  $\mathbf{k}$ . In this case, the contour generator  $\Gamma$  is a plane curve, but in general  $\Gamma$  is a space curve.

We next describe the projection properties of quadrics. For this class of surface algebraic expressions can be developed for the contour generator and apparent contour.

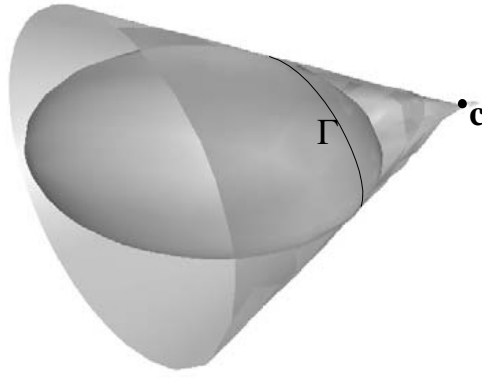


Fig. 8.4. **The cone of rays for a quadric.** The vertex of the cone is the camera centre. (a) The contour generator  $\Gamma$  of a quadric is a plane curve (a conic) which is the intersection of the quadric with the polar plane of the camera centre,  $C$ .

### 8.3 Action of a projective camera on quadrics

A quadric is a smooth surface and so its outline curve is given by points where the back-projected rays are tangent to the quadric surface as shown in figure 8.4.

Suppose the quadric is a sphere, then the cone of rays between the camera centre and quadric is right-circular, i.e. the contour generator is a circle, with the plane of the circle orthogonal to the line joining the camera and sphere centres. This can be seen from the rotational symmetry of the geometry about this line. The image of the sphere is obtained by intersecting the cone with the image plane. It is clear that this is a classical conic section, so that the apparent contour of a sphere is a conic. In particular if the sphere centre lies on the principal (Z) camera axis, then the conic is a circle.

Now consider a 3-space projective transformation of this geometry. Under this map the sphere is transformed to a quadric and the apparent contour to a conic. However, since intersection and tangency are preserved, the contour generator is a (plane) conic. Consequently, the apparent contour of a general quadric is a conic, and the contour generator is also a conic. We will now give algebraic representations for these geometric results.

**Forward projection of quadrics.** Since the outline arises from tangency, it is not surprising that the dual of the quadric,  $Q^*$ , is important here since it defines the tangent planes to the quadric  $Q$ .

**Result 8.9.** Under the camera matrix  $P$  the outline of the quadric  $Q$  is the conic  $C$  given by

$$C^* = PQ^*P^T. \quad (8.5)$$

**Proof.** This expression is simply derived from the observation that lines  $l$  tangent to the conic outline satisfy  $l^T C^* l = 0$ . These lines back-project to planes  $\pi = P^T l$  that are tangent to the quadric and thus satisfy  $\pi^T Q^* \pi = 0$ . Then it follows that for each line

$$\begin{aligned} \pi^T Q^* \pi &= l^T P Q^* P^T l \\ &= l^T C^* l = 0 \end{aligned}$$

and since this is true for all lines tangent to  $C$  the result follows.  $\square$

Note the similarity of (8.5) with the projection of a line represented by a Plücker matrix (8.2). An expression for the projection of the point quadric  $Q$  can be derived from (8.5) but it is quite complicated. However, the plane of the contour generator is easily expressed in terms of  $Q$ :

- *The plane of  $\Gamma$  for a quadric  $Q$  and camera with centre  $C$  is given by  $\pi_\Gamma = QC$ .*

This result follows directly from the pole–polar relation for a point and quadric of section 3.2.3(p73). Its proof is left as an exercise. Note, the intersection of a quadric and plane is a conic. So  $\Gamma$  is a conic and its image  $\gamma$ , which is the apparent contour, is also a conic as has been seen above.

We may also derive an expression for the cone of rays formed by the camera centre and quadric. This cone is a degenerate quadric of rank 3.

**Result 8.10.** *The cone with vertex  $V$  and tangent to the quadric  $Q$  is the degenerate quadric*

$$Q_{co} = (V^T Q V)Q - (QV)(QV)^T.$$

Note that  $Q_{co}V = 0$ , so that  $V$  is the vertex of the cone as required. The proof is omitted.

**Example 8.11.** We write the quadric in block form:

$$Q = \begin{bmatrix} Q_{3 \times 3} & \mathbf{q} \\ \mathbf{q}^T & q_{44} \end{bmatrix}.$$

Then if  $V = (0, 0, 0, 1)^T$ , which corresponds to the cone vertex being at the centre of the world coordinate frame,

$$Q_{co} = \begin{bmatrix} q_{44}Q_{3 \times 3} - \mathbf{q}\mathbf{q}^T & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix}$$

which is clearly a degenerate quadric.  $\triangle$

#### 8.4 The importance of the camera centre

An object in 3-space and camera centre define a set of rays, and an image is obtained by intersecting these rays with a plane. Often this set is referred to as a *cone* of rays, even though it is not a classical cone. Suppose the cone of rays is intersected by two planes, as shown in figure 8.5, then the two images,  $I$  and  $I'$ , are clearly related by a perspective map. This means that images obtained with the same camera centre may be mapped to one another by a plane projective transformation, in other words they are projectively equivalent and so have the same projective properties. A camera can thus be thought of as a projective imaging device – measuring projective properties of the cone of rays with vertex the camera centre.

The result that the two images  $I$  and  $I'$  are related by a homography will now be derived algebraically to obtain a formula for this homography. Consider two cameras

$$P = KR[I \mid -\tilde{C}], \quad P' = K'R'[I \mid -\tilde{C}]$$

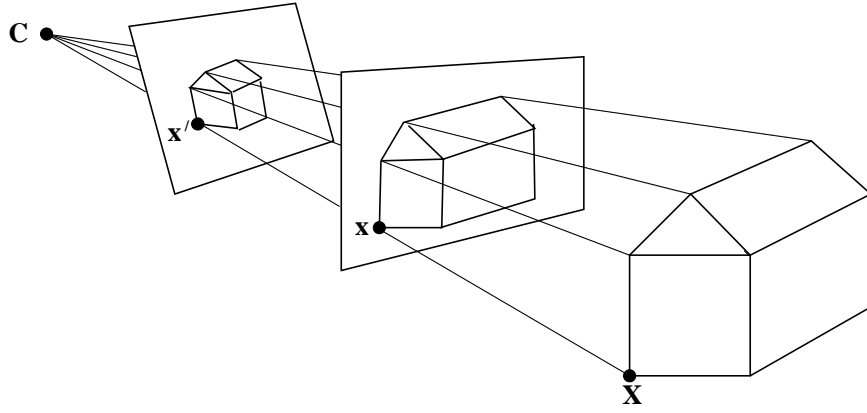


Fig. 8.5. **The cone of rays with vertex the camera centre.** An image is obtained by intersecting this cone with a plane. A ray between a 3-space point  $\mathbf{X}$  and the camera centre  $\mathbf{C}$  pierces the planes in the image points  $\mathbf{x}$  and  $\mathbf{x}'$ . All such image points are related by a planar homography,  $\mathbf{x}' = \mathbf{H}\mathbf{x}$ .

with the same centre. Note that since the cameras have a common centre there is a simple relation between them, namely  $\mathbf{P}' = (\mathbf{K}'\mathbf{R}')(\mathbf{K}\mathbf{R})^{-1}\mathbf{P}$ . It then follows that the images of a 3-space point  $\mathbf{X}$  by the two cameras are related as

$$\mathbf{x}' = \mathbf{P}'\mathbf{X} = (\mathbf{K}'\mathbf{R}')(\mathbf{K}\mathbf{R})^{-1}\mathbf{P}\mathbf{X} = (\mathbf{K}'\mathbf{R}')(\mathbf{K}\mathbf{R})^{-1}\mathbf{x}.$$

That is, the corresponding image points are related by a planar homography (a  $3 \times 3$  matrix) as  $\mathbf{x}' = \mathbf{H}\mathbf{x}$ , where  $\mathbf{H} = (\mathbf{K}'\mathbf{R}')(\mathbf{K}\mathbf{R})^{-1}$ .

We will now investigate several cases of moving the image plane whilst fixing the camera centre. For simplicity the world coordinate frame will be chosen to coincide with the camera's, so that  $\mathbf{P} = \mathbf{K}[\mathbf{I} \mid \mathbf{0}]$  (and it will be assumed that the image plane never contains the centre, as the image would then be degenerate).

#### 8.4.1 Moving the image plane

Consider first an increase in focal length. To a first approximation this corresponds to a displacement of the image plane along the principal axis. The image effect is a simple magnification. This is only a first approximation because with a compound lens zooming will perturb both the principal point and the effective camera centre. Algebraically, if  $\mathbf{x}$ ,  $\mathbf{x}'$  are the images of a point  $\mathbf{X}$  before and after zooming, then

$$\begin{aligned} \mathbf{x} &= \mathbf{K}[\mathbf{I} \mid \mathbf{0}]\mathbf{X} \\ \mathbf{x}' &= \mathbf{K}'[\mathbf{I} \mid \mathbf{0}]\mathbf{X} = \mathbf{K}'\mathbf{K}^{-1}(\mathbf{K}[\mathbf{I} \mid \mathbf{0}]\mathbf{X}) = \mathbf{K}'\mathbf{K}^{-1}\mathbf{x} \end{aligned}$$

so that  $\mathbf{x}' = \mathbf{H}\mathbf{x}$  with  $\mathbf{H} = \mathbf{K}'\mathbf{K}^{-1}$ . If only the focal lengths differ between  $\mathbf{K}$  and  $\mathbf{K}'$  then a short calculation shows that

$$\mathbf{K}'\mathbf{K}^{-1} = \begin{bmatrix} k\mathbf{I} & (1-k)\tilde{\mathbf{x}}_0 \\ \mathbf{0}^\top & 1 \end{bmatrix}.$$

where  $\tilde{\mathbf{x}}_0$  is the inhomogeneous principal point, and  $k = f'/f$  is the magnification factor. This result follows directly from similar triangles: the effect of zooming by a

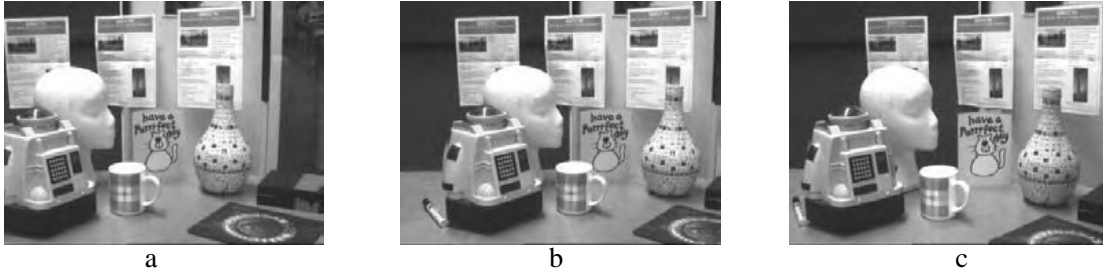


Fig. 8.6. Between images (a) and (b) the camera rotates about the camera centre. Corresponding points (that is images of the same 3D point) are related by a plane projective transformation. Note that 3D points at different depths which are coincident in image (a), such as the mug lip and cat body, are also coincident in (b), so there is no motion parallax in this case. However, between images (a) and (c) the camera rotates about the camera centre and translates. Under this general motion coincident points of differing depth in (a) are imaged at different points in (c), so there is motion parallax in this case due to the camera translation.

factor  $k$  is to move the image point  $\tilde{\mathbf{x}}$  on a line radiating from the principal point  $\tilde{\mathbf{x}}_0$  to the point  $\tilde{\mathbf{x}}' = k\tilde{\mathbf{x}} + (1 - k)\tilde{\mathbf{x}}_0$ . Algebraically, using the most general form (6.10–p157) of the calibration matrix  $\mathbf{K}$ , we may write

$$\begin{aligned} \mathbf{K}' &= \begin{bmatrix} k\mathbf{I} & (1 - k)\tilde{\mathbf{x}}_0 \\ \mathbf{0}^\top & 1 \end{bmatrix} \mathbf{K} = \begin{bmatrix} k\mathbf{I} & (1 - k)\tilde{\mathbf{x}}_0 \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} \mathbf{A} & \tilde{\mathbf{x}}_0 \\ \mathbf{0}^\top & 1 \end{bmatrix} \\ &= \begin{bmatrix} k\mathbf{A} & \tilde{\mathbf{x}}_0 \\ \mathbf{0}^\top & 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} k\mathbf{I} & \\ & 1 \end{bmatrix}. \end{aligned}$$

This shows that

- The effect of zooming by a factor  $k$  is to multiply the calibration matrix  $\mathbf{K}$  on the right by  $\text{diag}(k, k, 1)$ .

### 8.4.2 Camera rotation

A second common example is where the camera is rotated about its centre with no change in the internal parameters. Examples of this “pure” rotation are given in figure 8.6 and figure 8.9. Algebraically, if  $\mathbf{x}$ ,  $\mathbf{x}'$  are the images of a point  $\mathbf{X}$  before and after the pure rotation

$$\begin{aligned} \mathbf{x} &= \mathbf{K}[\mathbf{I} \mid \mathbf{0}]\mathbf{X} \\ \mathbf{x}' &= \mathbf{K}[\mathbf{R} \mid \mathbf{0}]\mathbf{X} = \mathbf{K}\mathbf{R}\mathbf{K}^{-1}\mathbf{K}[\mathbf{I} \mid \mathbf{0}]\mathbf{X} = \mathbf{K}\mathbf{R}\mathbf{K}^{-1}\mathbf{x} \end{aligned}$$

so that  $\mathbf{x}' = \mathbf{H}\mathbf{x}$  with  $\mathbf{H} = \mathbf{K}\mathbf{R}\mathbf{K}^{-1}$ . This homography is a *conjugate rotation* and is discussed further in section A7.1(p628). For now, we mention a few of its properties by way of an example.

#### Example 8.12. Properties of a conjugate rotation

The homography  $\mathbf{H} = \mathbf{K}\mathbf{R}\mathbf{K}^{-1}$  has the same eigenvalues (up to scale) as the rotation matrix, namely  $\{\mu, \mu e^{i\theta}, \mu e^{-i\theta}\}$ , where  $\mu$  is an unknown scale factor (if  $\mathbf{H}$  is scaled such that  $\det \mathbf{H} = 1$ , then  $\mu = 1$ ). Consequently the angle of rotation between views may be computed directly from the phase of the complex eigenvalues of  $\mathbf{H}$ . Similarly, it can be

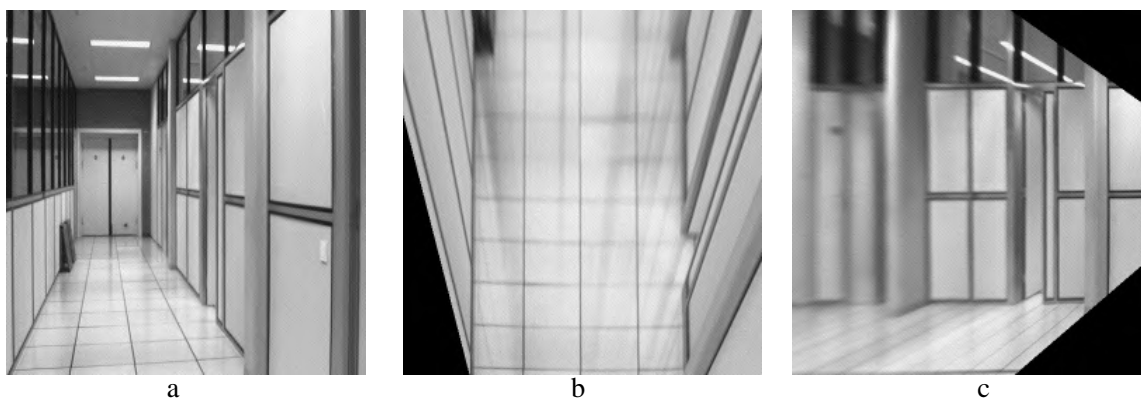


Fig. 8.7. **Synthetic views.** (a) Source image. (b) Fronto-parallel view of the corridor floor generated from (a) using the four corners of a floor tile to compute the homography. (c) Fronto-parallel view of the corridor wall generated from (a) using the four corners of the door frame to compute the homography.

shown (see exercises) that the eigenvector of  $H$  corresponding to the real eigenvalue is the vanishing point of the rotation axis.

For example, between images (a) and (b) of figure 8.6 there is a pure rotation of the camera. The homography  $H$  is computed by algorithm 4.6(p123), and from this the angle of rotation is estimated as  $4.66^\circ$ , and the axis vanishing point as  $(-0.0088, 1, 0.0001)^\top$ , i.e. virtually at infinity in the  $y$  direction, so the rotation axis is almost parallel to the  $y$ -axis.  $\triangle$

The transformation  $H = KRK^{-1}$  is an example of the *infinite homography* mapping  $H_\infty$ , that will appear many times through this book. It is defined in section 13.4(p338). The conjugation property is used for auto-calibration in chapter 19.

### 8.4.3 Applications and examples

The homographic relation between images with the same camera centre can be exploited in several ways. One is the creation of synthetic images by projective warping. Another is mosaicing, where panoramic images can be created by using planar homographies to “sew” together views obtained by a rotating camera.

#### Example 8.13. Synthetic views

New images corresponding to different camera orientations (with the same camera centre) can be generated from an existing image by warping with planar homographies.

In a fronto-parallel view a rectangle is imaged as a rectangle, and the world and image rectangle have the same aspect ratio. Conversely, a fronto-parallel view can be synthesized by warping an image with the homography that maps a rectangle imaged as a quadrilateral to a rectangle with the correct aspect ratio. The algorithm is:

- (i) Compute the homography  $H$  which maps the image quadrilateral to a rectangle with the correct aspect ratio.
- (ii) Projectively warp the source image with this homography.

Examples are shown in figure 8.7.  $\triangle$

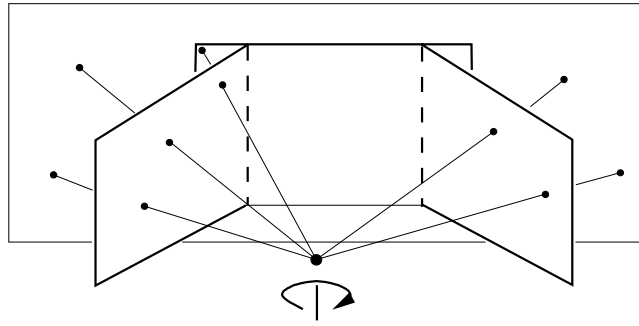


Fig. 8.8. Three images acquired by a rotating camera may be registered to the frame of the middle one, as shown, by projectively warping the outer images to align with the middle one.

### Example 8.14. Planar panoramic mosaicing

Images acquired by a camera rotating about its centre are related to each other by a planar homography. A set of such images may be registered with the plane of one of the images by projectively warping the other images, as illustrated in figure 8.8.



Fig. 8.9. **Planar panoramic mosaicing.** Eight images (out of thirty) acquired by rotating a camcorder about its centre. The thirty images are registered (automatically) using planar homographies and composed into the single panoramic mosaic shown. Note the characteristic “bow tie” shape resulting from registering to an image at the middle of the sequence.

In outline the algorithm is:

- (i) Choose one image of the set as a reference.
- (ii) Compute the homography  $H$  which maps one of the other images of the set to this reference image.

- (iii) Projectively warp the image with this homography, and augment the reference image with the non-overlapping part of the warped image.
- (iv) Repeat the last two steps for the remaining images of the set.

The homographies may be computed by identifying (at least) four corresponding points, or by using the automatic method of algorithm 4.6(p123). An example mosaic is shown in figure 8.9. △

#### 8.4.4 Projective (reduced) notation

It will be seen in chapter 20 that if canonical projective coordinates are chosen for world and image points, i.e.

$$\mathbf{x}_1 = (1, 0, 0, 0)^T, \mathbf{x}_2 = (0, 1, 0, 0)^T, \mathbf{x}_3 = (0, 0, 1, 0)^T, \mathbf{x}_4 = (0, 0, 0, 1)^T,$$

and

$$\mathbf{x}_1 = (1, 0, 0)^T, \mathbf{x}_2 = (0, 1, 0)^T, \mathbf{x}_3 = (0, 0, 1)^T, \mathbf{x}_4 = (1, 1, 1)^T,$$

then the camera matrix

$$P = \begin{bmatrix} a & 0 & 0 & -d \\ 0 & b & 0 & -d \\ 0 & 0 & c & -d \end{bmatrix} \quad (8.6)$$

satisfies  $\mathbf{x}_i = P\mathbf{X}_i$ ,  $i = 1, \dots, 4$ , and also that  $P(a^{-1}, b^{-1}, c^{-1}, d^{-1})^T = \mathbf{0}$ , which means that the camera centre is  $\mathbf{C} = (a^{-1}, b^{-1}, c^{-1}, d^{-1})^T$ . This is known as the *reduced* camera matrix, and it is clearly completely specified by the 3 degrees of freedom of the camera centre  $\mathbf{C}$ . This is a further illustration of the fact that all images acquired by cameras with the same camera centre are projectively equivalent – the camera has been reduced to its essence: a projective device whose action is to map  $\mathbb{P}^3$  to  $\mathbb{P}^2$  with only the position of the camera centre affecting the result. This camera representation is used in establishing duality relations in chapter 20.

#### 8.4.5 Moving the camera centre

The cases of zooming and camera rotation illustrate that moving the image plane, whilst fixing the camera centre, induces a transformation between images that depends only on the image plane motion, but *not* on the 3-space structure. Conversely, no information on 3-space structure can be obtained by this action. However, if the camera centre is moved then the map between corresponding image points *does* depend on the 3-space structure, and indeed may often be used to (partially) determine the structure. This is the subject of much of the remainder of this book.

How can one determine from the images alone whether the camera centre has moved? Consider two 3-space points which have coincident images in the first view, i.e. the points are on the same ray. If the camera centre is moved (not along that ray) the image coincidence is lost. This relative displacement of previously coincident image points is termed *parallax*, and is illustrated in figure 8.6 and shown schematically in figure 8.10. If the scene is static and motion parallax is evident between two views then the camera centre has moved. Indeed, a convenient method for obtaining a camera

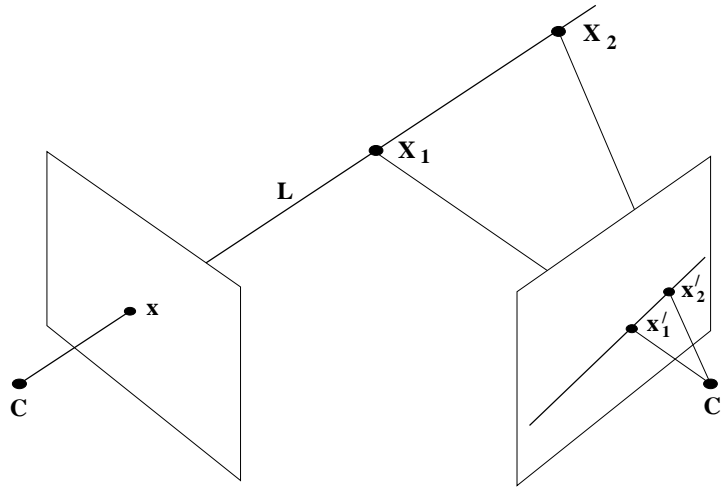


Fig. 8.10. **Motion parallax.** The images of the space points  $X_1$  and  $X_2$  are coincident when viewed by the camera with centre  $C$ . However, when viewed by a camera with centre  $C'$ , which does not lie on the line  $L$  through  $X_1$  and  $X_2$ , the images of the space points are not coincident. In fact the line through the image points  $x'_1$  and  $x'_2$  is the image of the ray  $L$ , and will be seen in chapter 9 to be an **epipolar line**. The vector between the points  $x'_1$  and  $x'_2$  is the parallax.

motion that is only a rotation about its centre (for example for a camera mounted on a robot head) is to adjust the motion until there is no parallax.

An important special case of 3-space structure is when all scene points are coplanar. In this case the images of corresponding points are related by a planar homography even if the camera centre is moved. The map between images in this case is discussed in detail in chapter 13 on planes. In particular vanishing points, which are images of points on the plane  $\pi_\infty$ , are related by a planar homography for any camera motion. We return to this in section 8.6.

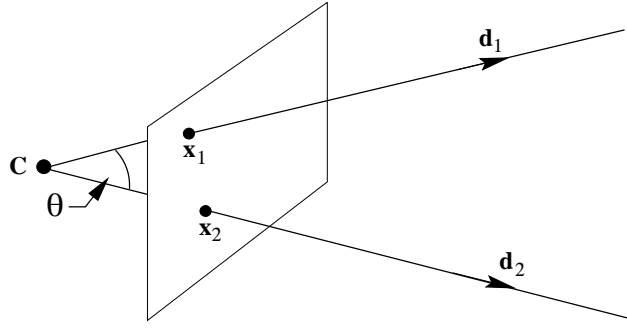
### 8.5 Camera calibration and the image of the absolute conic

Up to this point we have discussed projective properties of the forward and back-projection of various entities (point, lines, conics ...). These properties depend only on the  $3 \times 4$  form of the projective camera matrix  $P$ . Now we describe what is gained if the camera internal calibration,  $K$ , is known. It will be seen that Euclidean properties, such as the angle between two rays, can then be measured.

**What does calibration give?** An image point  $x$  back-projects to a ray defined by  $x$  and the camera centre. Calibration relates the image point to the ray's *direction*. Suppose points on the ray are written as  $\tilde{X} = \lambda d$  in the camera Euclidean coordinate frame, then these points map to the point  $x = K[I \mid 0](\lambda d^T, 1)^T = Kd$  up to scale. Conversely the direction  $d$  is obtained from the image point  $x$  as  $d = K^{-1}x$ . Thus we have established:

**Result 8.15.** The camera calibration matrix  $K$  is the (affine) transformation between  $x$  and the ray's direction  $d = K^{-1}x$  measured in the camera's Euclidean coordinate frame.

Note,  $d = K^{-1}x$  is in general *not* a unit vector.

Fig. 8.11. The angle  $\theta$  between two rays.

The angle between two rays, with directions  $\mathbf{d}_1$ ,  $\mathbf{d}_2$  corresponding to image points  $\mathbf{x}_1$ ,  $\mathbf{x}_2$  respectively, may be obtained from the familiar cosine formula for the angle between two vectors:

$$\begin{aligned} \cos \theta &= \frac{\mathbf{d}_1^T \mathbf{d}_2}{\sqrt{\mathbf{d}_1^T \mathbf{d}_1} \sqrt{\mathbf{d}_2^T \mathbf{d}_2}} = \frac{(\mathbf{K}^{-1} \mathbf{x}_1)^T (\mathbf{K}^{-1} \mathbf{x}_2)}{\sqrt{(\mathbf{K}^{-1} \mathbf{x}_1)^T (\mathbf{K}^{-1} \mathbf{x}_1)} \sqrt{(\mathbf{K}^{-1} \mathbf{x}_2)^T (\mathbf{K}^{-1} \mathbf{x}_2)}} \\ &= \frac{\mathbf{x}_1^T (\mathbf{K}^{-T} \mathbf{K}^{-1}) \mathbf{x}_2}{\sqrt{\mathbf{x}_1^T (\mathbf{K}^{-T} \mathbf{K}^{-1}) \mathbf{x}_1} \sqrt{\mathbf{x}_2^T (\mathbf{K}^{-T} \mathbf{K}^{-1}) \mathbf{x}_2}}. \end{aligned} \quad (8.7)$$

The formula (8.7) shows that if  $\mathbf{K}$ , and consequently the matrix  $\mathbf{K}^{-T} \mathbf{K}^{-1}$ , is known, then the angle between rays can be measured from their corresponding image points. A camera for which  $\mathbf{K}$  is known is termed *calibrated*. A calibrated camera is a *direction sensor*, able to measure the direction of rays – like a 2D protractor.

The calibration matrix  $\mathbf{K}$  also provides a relation between an image line and a scene plane:

**Result 8.16.** *An image line  $l$  defines a plane through the camera centre with normal direction  $\mathbf{n} = \mathbf{K}^T l$  measured in the camera's Euclidean coordinate frame.*

Note, the normal  $\mathbf{n}$  will not in general be a unit vector.

**Proof.** Points  $\mathbf{x}$  on the line  $l$  back-project to directions  $\mathbf{d} = \mathbf{K}^{-1} \mathbf{x}$  which are orthogonal to the plane normal  $\mathbf{n}$ , and thus satisfy  $\mathbf{d}^T \mathbf{n} = \mathbf{x}^T \mathbf{K}^{-T} \mathbf{n} = 0$ . Since points on  $l$  satisfy  $\mathbf{x}^T l = 0$ , it follows that  $l = \mathbf{K}^{-T} \mathbf{n}$ , and hence  $\mathbf{n} = \mathbf{K}^T l$ .  $\square$

### 8.5.1 The image of the absolute conic

We now derive a very important result which relates the calibration matrix  $\mathbf{K}$  to the image of the absolute conic,  $\omega$ . First we must determine the map between the plane at infinity,  $\pi_\infty$ , and the camera image plane. Points on  $\pi_\infty$  may be written as  $\mathbf{X}_\infty = (\mathbf{d}^T, 0)^T$ , and are imaged by a general camera  $\mathbf{P} = \mathbf{K}\mathbf{R}[\mathbf{I} \mid -\tilde{\mathbf{C}}]$  as

$$\mathbf{x} = \mathbf{P}\mathbf{X}_\infty = \mathbf{K}\mathbf{R}[\mathbf{I} \mid -\tilde{\mathbf{C}}] \begin{pmatrix} \mathbf{d} \\ 0 \end{pmatrix} = \mathbf{K}\mathbf{R}\mathbf{d}.$$

This shows that

- the mapping between  $\pi_\infty$  and an image is given by the planar homography  $\mathbf{x} = \mathbf{H}\mathbf{d}$  with

$$\mathbf{H} = \mathbf{K}\mathbf{R}. \quad (8.8)$$

Note that this map is independent of the position of the camera,  $\mathbf{C}$ , and depends only on the camera internal calibration and orientation with respect to the world coordinate frame.

Now, since the absolute conic  $\Omega_\infty$  (section 3.6(p81)) is on  $\pi_\infty$  we can compute its image under  $\mathbf{H}$ , and find

**Result 8.17.** *The image of the absolute conic (the IAC) is the conic  $\omega = (\mathbf{K}\mathbf{K}^\top)^{-1} = \mathbf{K}^{-\top}\mathbf{K}^{-1}$ .*

**Proof.** From result 2.13(p37) under a point homography  $\mathbf{x} \mapsto \mathbf{H}\mathbf{x}$  a conic  $\mathbf{C}$  maps as  $\mathbf{C} \mapsto \mathbf{H}^{-\top}\mathbf{C}\mathbf{H}^{-1}$ . It follows that  $\Omega_\infty$ , which is the conic  $\mathbf{C} = \Omega_\infty = \mathbf{I}$  on  $\pi_\infty$ , maps to  $\omega = (\mathbf{K}\mathbf{R})^{-\top}\mathbf{I}(\mathbf{K}\mathbf{R})^{-1} = \mathbf{K}^{-\top}\mathbf{R}\mathbf{R}^{-1}\mathbf{K}^{-1} = (\mathbf{K}\mathbf{K}^\top)^{-1}$ . So the IAC  $\omega = (\mathbf{K}\mathbf{K}^\top)^{-1}$ .  $\square$

Like  $\Omega_\infty$  the conic  $\omega$  is an imaginary point conic with no real points. For the moment it may be thought of as a convenient algebraic device, but it will be used in computations later in this chapter, and also in chapter 19 on camera auto-calibration.

A few remarks here:

- The image of the absolute conic,  $\omega$ , depends only on the internal parameters  $\mathbf{K}$  of the matrix  $\mathbf{P}$ ; it does not depend on the camera orientation or position.
- It follows from (8.7) that the angle between two rays is given by the simple expression

$$\cos \theta = \frac{\mathbf{x}_1^\top \omega \mathbf{x}_2}{\sqrt{\mathbf{x}_1^\top \omega \mathbf{x}_1} \sqrt{\mathbf{x}_2^\top \omega \mathbf{x}_2}}. \quad (8.9)$$

This expression is independent of the projective coordinate frame in the image, that is, it is unchanged under projective transformation of the image. To see this consider any 2D projective transformation,  $\mathbf{H}$ . The points  $\mathbf{x}_i$  are transformed to  $\mathbf{H}\mathbf{x}_i$ , and  $\omega$  transforms (as any image conic) to  $\mathbf{H}^{-\top}\omega\mathbf{H}^{-1}$ . Thus, (8.9) is unchanged, and hence holds in any projective coordinate frame in the image.

- A particularly important specialization of (8.9) is that if two image points  $\mathbf{x}_1$  and  $\mathbf{x}_2$  correspond to orthogonal directions then

$$\mathbf{x}_1^\top \omega \mathbf{x}_2 = 0. \quad (8.10)$$

This equation will be used at several points later in the book as it provides a linear constraint on  $\omega$ .

- We may also define the dual image of the absolute conic (the DIAC) as

$$\omega^* = \omega^{-1} = \mathbf{K}\mathbf{K}^\top. \quad (8.11)$$

This is a dual (line) conic, whereas  $\omega$  is a point conic (though it contains no real points). The conic  $\omega^*$  is the image of  $\mathbf{Q}_\infty^*$  and is given by (8.5)  $\omega^* = \mathbf{P}\mathbf{Q}_\infty^*\mathbf{P}^\top$ .

- (v) Result 8.17 shows that once  $\omega$  (or equivalently  $\omega^*$ ) is identified in an image then  $K$  is also determined. This follows because a symmetric matrix  $\omega$  may be uniquely decomposed into a product  $\omega^* = KK^T$  of an upper-triangular matrix with positive diagonal entries and its transpose by the Cholesky factorization (see result A4.5(p582)).
- (vi) It was seen in chapter 3 that a plane  $\pi$  intersects  $\pi_\infty$  in a line, and this line intersects  $\Omega_\infty$  in two points which are the circular points of  $\pi$ . The imaged circular points lie on  $\omega$  at the points at which the vanishing line of the plane  $\pi$  intersects  $\omega$ .

These final two properties of  $\omega$  are the basis for a calibration algorithm, as shown in the following example.

**Example 8.18. A simple calibration device**

The image of three squares (on planes which are not parallel, but which need not be orthogonal) provides sufficiently many constraints to compute  $K$ . Consider one of the squares. The correspondences between its four corner points and their images define the homography  $H$  between the plane  $\pi$  of the square and the image. Applying this homography to circular points on  $\pi$  determines their images as  $H(1, \pm i, 0)^T$ . Thus we have two points on the (as yet unknown)  $\omega$ . A similar procedure applied to the other squares generates a total of six points on  $\omega$ , from which it may be computed (since five points are required to determine a conic). In outline the algorithm has the following steps:

- (i) For each square compute the homography  $H$  that maps its corner points,  $(0, 0)^T, (1, 0)^T, (0, 1)^T, (1, 1)^T$ , to their imaged points. (The alignment of the plane coordinate system with the square is a similarity transformation and does not affect the position of the circular points on the plane).
- (ii) Compute the imaged circular points for the plane of that square as  $H(1, \pm i, 0)^T$ . Writing  $H = [\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3]$ , the imaged circular points are  $\mathbf{h}_1 \pm i\mathbf{h}_2$ .
- (iii) Fit a conic  $\omega$  to the six imaged circular points. The constraint that the imaged circular points lie on  $\omega$  may be rewritten as two real constraints. If  $\mathbf{h}_1 \pm i\mathbf{h}_2$  lies on  $\omega$  then  $(\mathbf{h}_1 \pm i\mathbf{h}_2)^T \omega (\mathbf{h}_1 \pm i\mathbf{h}_2) = 0$ , and the imaginary and real parts give respectively:

$$\mathbf{h}_1^T \omega \mathbf{h}_2 = 0 \quad \text{and} \quad \mathbf{h}_1^T \omega \mathbf{h}_1 = \mathbf{h}_2^T \omega \mathbf{h}_2 \quad (8.12)$$

which are equations linear in  $\omega$ . The conic  $\omega$  is determined up to scale from five or more such equations.

- (iv) Compute the calibration  $K$  from  $\omega = (KK^T)^{-1}$  using the Cholesky factorization.

Figure 8.12 shows a calibration object consisting of three planes imprinted with squares, and the computed matrix  $K$ . For the purpose of internal calibration, the squares have the advantage over a standard calibration object (e.g. figure 7.1(p182)) that no measured 3D co-ordinates are required.  $\triangle$

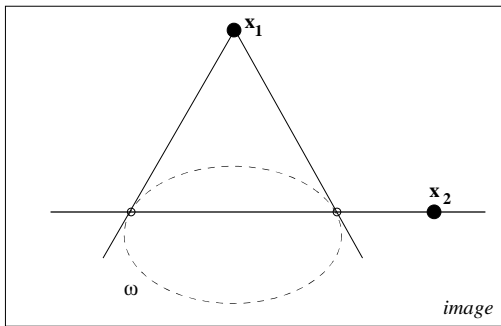


a

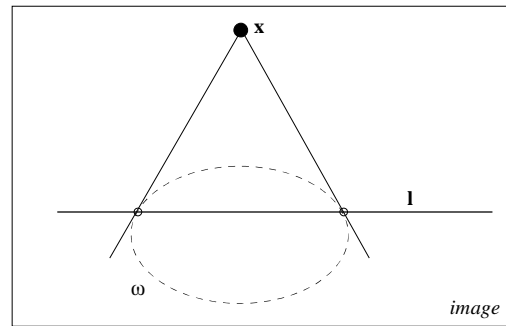
$$K = \begin{bmatrix} 1108.3 & -9.8 & 525.8 \\ 0 & 1097.8 & 395.9 \\ 0 & 0 & 1 \end{bmatrix}$$

b

Fig. 8.12. **Calibration from metric planes.** (a) Three squares provide a simple calibration object. The planes need not be orthogonal. (b) The computed calibration matrix using the algorithm of example 8.18. The image size is  $1024 \times 768$  pixels.



a



b

Fig. 8.13. **Orthogonality represented by conjugacy and pole–polar relationships.** (a) Image points  $x_1, x_2$  back-project to orthogonal rays if the points are conjugate with respect to  $\omega$ , i.e.  $x_1^T \omega x_2 = 0$ . (b) The point  $x$  and line  $l$  back-project to a ray and plane that are orthogonal if  $x$  and  $l$  are pole–polar with respect to  $\omega$ , i.e.  $l = \omega x$ . For example (see section 8.6.3), the vanishing point of the normal direction to a plane and the vanishing line of the plane are pole–polar with respect to  $\omega$ .

We will return to camera calibration in section 8.8, where vanishing points and lines provide constraints on  $K$ . The geometric constraints that are used in example 8.18 are discussed further in section 8.8.1.

### 8.5.2 Orthogonality and $\omega$

The conic  $\omega$  is a device for representing orthogonality in an image. It has already been seen (8.10) that if two image points  $x_1$  and  $x_2$  back-project to orthogonal rays, then the points satisfy  $x_1^T \omega x_2 = 0$ . Similarly, it may be shown that

**Result 8.19.** A point  $x$  and line  $l$  back-projecting to a ray and plane respectively that are orthogonal are related by  $l = \omega x$ .

Geometrically these relations express that image points back-projecting to orthogonal rays are conjugate with respect to  $\omega$  ( $x_1^T \omega x_2 = 0$ ), and that a point and line back-projecting to an orthogonal ray and plane are in a pole–polar relationship ( $l = \omega x$ ). See section 2.8.1(p58). A schematic representation of these two relations is given in figure 8.13.

These geometric representations of orthogonality, and indeed the projective representation (8.9) of the angle between two rays measured from image points, are simply specializations and a recapitulation of relations derived earlier in the book. For example, we have already developed a projective representation (3.23–p82) of the angle between two lines in 3-space, namely

$$\cos \theta = \frac{\mathbf{d}_1^T \Omega_\infty \mathbf{d}_2}{\sqrt{\mathbf{d}_1^T \Omega_\infty \mathbf{d}_1} \sqrt{\mathbf{d}_2^T \Omega_\infty \mathbf{d}_2}}$$

where  $\mathbf{d}_1$  and  $\mathbf{d}_2$  are the directions of the lines (which are the points at which the lines intersect  $\pi_\infty$ ). Rays are lines in 3-space which are coincident at the camera centre, and so (3.23–p82) may be applied directly to rays. This is precisely what (8.9) does – it is simply (3.23–p82) computed in the image.

Under the map (8.8)  $H = KR$ , which is the homography between the plane  $\pi_\infty$  in the world coordinate frame and the image plane,  $\Omega_\infty \mapsto H^T \omega H = (KR)^T \omega (KR)$  and  $\mathbf{d}_i = H^{-1} \mathbf{x}_i = (KR)^{-1} \mathbf{x}_i$ . Substituting these relations into (3.23–p82) gives (8.9). Similarly the conjugacy and pole–polar relations for orthogonality in the image are a direct image of those on  $\pi_\infty$ , as can be seen by comparing figure 3.8(p83) with figure 8.13.

In practice these orthogonality results find greatest application in the case of vanishing points and vanishing lines.

## 8.6 Vanishing points and vanishing lines

One of the distinguishing features of perspective projection is that the image of an object that stretches off to infinity can have finite extent. For example, an infinite scene line is imaged as a line terminating in a *vanishing point*. Similarly, parallel world lines, such as railway lines, are imaged as converging lines, and their image intersection is the vanishing point for the direction of the railway.

### 8.6.1 Vanishing points

The perspective geometry that gives rise to vanishing points is illustrated in figure 8.14. It is evident that geometrically the vanishing point of a line is obtained by intersecting the image plane with a ray parallel to the world line and passing through the camera centre. Thus a vanishing point depends only on the *direction* of a line, not on its position. Consequently a set of parallel world lines have a common vanishing point, as illustrated in figure 8.16.

Algebraically the vanishing point may be obtained as a limiting point as follows: Points on a line in 3-space through the point  $\mathbf{A}$  and with direction  $\mathbf{D} = (\mathbf{d}^T, 0)^T$  are written as  $\mathbf{X}(\lambda) = \mathbf{A} + \lambda \mathbf{D}$ , see figure 8.14b. As the parameter  $\lambda$  varies from 0 to  $\infty$  the point  $\mathbf{X}(\lambda)$  varies from the finite point  $\mathbf{A}$  to the point  $\mathbf{D}$  at infinity. Under a projective camera  $P = K[I \mid \mathbf{0}]$ , a point  $\mathbf{X}(\lambda)$  is imaged at

$$\mathbf{x}(\lambda) = P\mathbf{X}(\lambda) = P\mathbf{A} + \lambda P\mathbf{D} = \mathbf{a} + \lambda \mathbf{kd}$$

where  $\mathbf{a}$  is the image of  $\mathbf{A}$ . Then the vanishing point  $\mathbf{v}$  of the line is obtained as the

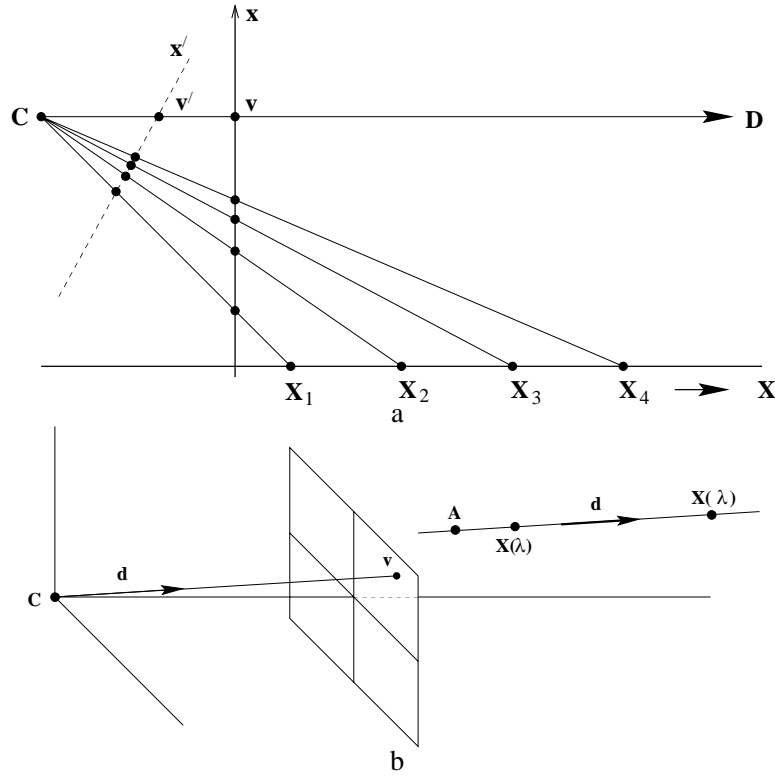


Fig. 8.14. **Vanishing point formation.** (a) Plane to line camera. The points  $X_i, i = 1, \dots, 4$  are equally spaced on the world line, but their spacing on the image line monotonically decreases. In the limit  $X \rightarrow \infty$  the world point is imaged at  $x = v$  on the vertical image line, and at  $x' = v'$  on the inclined image line. Thus the vanishing point of the world line is obtained by intersecting the image plane with a ray parallel to the world line through the camera centre  $C$ . (b) 3-space to plane camera. The vanishing point,  $v$ , of a line with direction  $d$  is the intersection of the image plane with a ray parallel to  $d$  through  $C$ . The world line may be parametrized as  $X(\lambda) = A + \lambda D$ , where  $A$  is a point on the line, and  $D = (d^T, 0)^T$ .

limit

$$v = \lim_{\lambda \rightarrow \infty} x(\lambda) = \lim_{\lambda \rightarrow \infty} (a + \lambda Kd) = Kd.$$

From result 8.15,  $v = Kd$  means that the vanishing point  $v$  back-projects to a ray with direction  $d$ . Note that  $v$  depends only on the direction  $d$  of the line, not on its position specified by  $A$ .

In the language of projective geometry this result is obtained directly: In projective 3-space the plane at infinity  $\pi_\infty$  is the plane of directions, and all lines with the same direction intersect  $\pi_\infty$  in the same point (see chapter 3). The vanishing point is simply the image of this intersection. Thus if a line has direction  $d$ , then it intersects  $\pi_\infty$  in the point  $X_\infty = (d^T, 0)^T$ . Then  $v$  is the image of  $X_\infty$

$$v = PX_\infty = K[I \mid 0] \begin{pmatrix} d \\ 0 \end{pmatrix} = Kd.$$

To summarize:

**Result 8.20.** *The vanishing point of lines with direction  $d$  in 3-space is the intersection*

$\mathbf{v}$  of the image plane with a ray through the camera centre with direction  $\mathbf{d}$ , namely  $\mathbf{v} = \mathbf{Kd}$ .

Note, lines parallel to the image plane are imaged as parallel lines, since  $\mathbf{v}$  is at infinity in the image. However, the converse – that parallel image lines are the image of parallel scene lines – does not hold since lines which intersect on the principal plane are imaged as parallel lines.

### Example 8.21. Camera rotation from vanishing points

Vanishing points are images of points at infinity, and provide orientation (attitude) information in a similar manner to that provided by the fixed stars. Consider two images of a scene obtained by calibrated cameras, where the two cameras differ in orientation and position. The points at infinity are part of the scene and so are independent of the camera. Their images, the vanishing points, are not affected by the change in camera position, but are affected by the camera rotation. Suppose both cameras have the same calibration matrix  $\mathbf{K}$ , and the camera rotates by  $\mathbf{R}$  between views.

Let a scene line have vanishing point  $\mathbf{v}_i$  in the first view, and  $\mathbf{v}'_i$  in the second. The vanishing point  $\mathbf{v}_i$  has direction  $\mathbf{d}_i$  measured in the first camera's Euclidean coordinate frame, and the corresponding vanishing point  $\mathbf{v}'_i$  has direction  $\mathbf{d}'_i$  measured in the second camera's Euclidean coordinate frame. These directions can be computed from the vanishing points, for example  $\mathbf{d}_i = \mathbf{K}^{-1}\mathbf{v}_i / \|\mathbf{K}^{-1}\mathbf{v}_i\|$ , where the normalizing factor  $\|\mathbf{K}^{-1}\mathbf{v}_i\|$  is included to ensure that  $\mathbf{d}_i$  is a unit vector. The directions  $\mathbf{d}_i$  and  $\mathbf{d}'_i$  are related by the camera rotation as  $\mathbf{d}'_i = \mathbf{R}\mathbf{d}_i$ , which represents two independent constraints on  $\mathbf{R}$ . Thus the rotation matrix  $\mathbf{R}$  can be computed from two such corresponding directions.  $\triangle$

**The angle between two scene lines.** We have seen that the vanishing point of a scene line back-projects to a ray parallel to the scene line. Consequently (8.9), which determines the angle between rays back-projected from image points, enables the angle between the directions of two scene lines to be measured from their vanishing points:

**Result 8.22.** Let  $\mathbf{v}_1$  and  $\mathbf{v}_2$  be the vanishing points of two lines in an image, and let  $\boldsymbol{\omega}$  be the image of the absolute conic in the image. If  $\theta$  is the angle between the two line directions, then

$$\cos \theta = \frac{\mathbf{v}_1^T \boldsymbol{\omega} \mathbf{v}_2}{\sqrt{\mathbf{v}_1^T \boldsymbol{\omega} \mathbf{v}_1} \sqrt{\mathbf{v}_2^T \boldsymbol{\omega} \mathbf{v}_2}} . \quad (8.13)$$

### A note on computing vanishing points

Often vanishing points are computed from the image of a set of parallel line segments, though they may be determined in other ways for example by using equal length intervals on a line as described in example 2.18(p50) and example 2.20(p51). In the case of imaged parallel line segments the objective is to estimate their common image intersection – which is the image of the direction of the parallel scene lines. Due to measurement noise the imaged line segments will generally *not* intersect in a unique

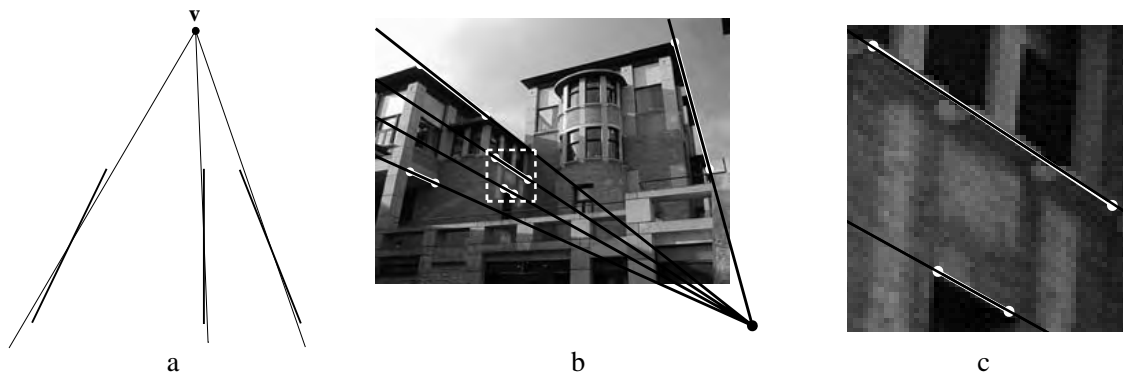


Fig. 8.15. **ML estimate of a vanishing point from imaged parallel scene lines.** (a) Estimating the vanishing point  $\mathbf{v}$  involves fitting a line (shown thin here) through  $\mathbf{v}$  to each measured line (shown thick here). The ML estimate of  $\mathbf{v}$  is the point which minimizes the sum of squared orthogonal distances between the fitted lines and the measured lines' end points. (b) Measured line segments are shown in white, and fitted lines in black. (c) A close-up of the dashed square in (b). Note the very slight angle between the measured and fitted lines.

point. Commonly the vanishing point is then computed by intersecting the lines pairwise and using the centroid of these intersections, or finding the closest point to all the measured lines. However, these are *not* optimal procedures.

Under the assumption of Gaussian measurement noise, the maximum likelihood estimate (MLE) of the vanishing point *and* line segments is computed by determining a set of lines that do intersect in a single point, and which minimize the sum of squared orthogonal distances from the endpoints of the measured line segments as shown in figure 8.15(a). This minimization may be computed numerically using the Levenberg–Marquardt algorithm (section A6.2(p600)). Note that if the lines are defined by fitting to many points, rather than just their end points, one can use the method described in section 16.7.2(p404) to reduce each line to an equivalent pair of weighted end points which can then be used in this algorithm. Figure 8.15(b)(c) shows an example of a vanishing point computed in this manner. It is evident that the residuals between the measured and fitted lines are very small.

### 8.6.2 Vanishing lines

Parallel planes in 3-space intersect  $\pi_\infty$  in a common line, and the image of this line is the vanishing line of the plane. Geometrically the vanishing line is constructed, as shown in figure 8.16, by intersecting the image with a plane parallel to the scene plane through the camera centre. It is clear that a vanishing line depends only on the *orientation* of the scene plane; it does not depend on its position. Since lines parallel to a plane intersect the plane at  $\pi_\infty$ , it is easily seen that the vanishing point of a line parallel to a plane lies on the vanishing line of the plane. An example is shown in figure 8.17.

If the camera calibration  $K$  is known then a scene plane's vanishing line may be used to determine information about the plane, and we mention three examples here:

- (i) The plane's orientation relative to the camera may be determined from its vanishing line. From result 8.16 a plane through the camera centre with normal

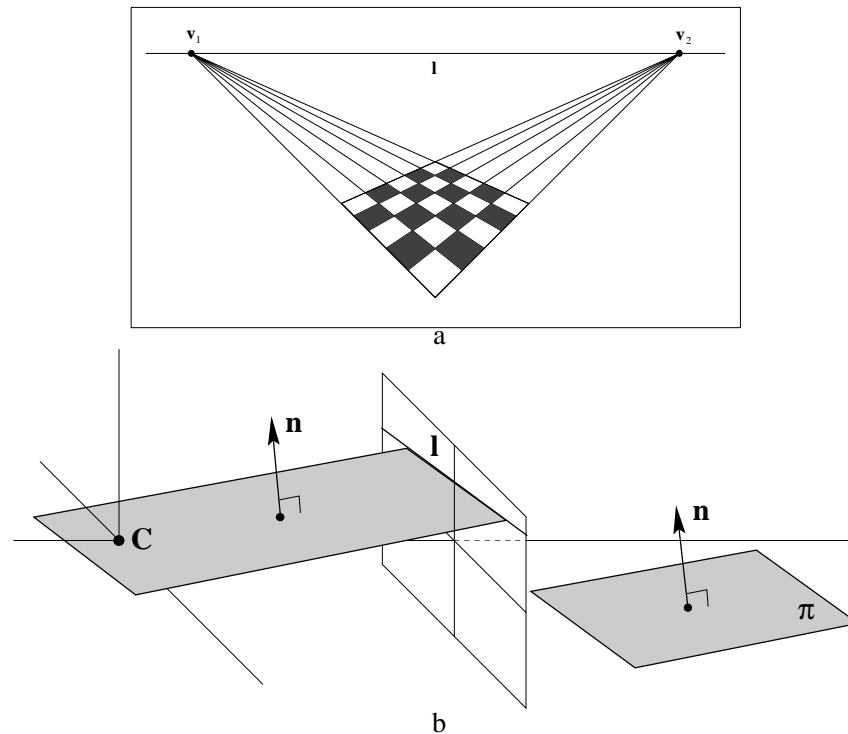


Fig. 8.16. **Vanishing line formation.** (a) The two sets of parallel lines on the scene plane converge to the vanishing points  $v_1$  and  $v_2$  in the image. The line  $l$  through  $v_1$  and  $v_2$  is the vanishing line of the plane. (b) The vanishing line  $l$  of a plane  $\pi$  is obtained by intersecting the image plane with a plane through the camera centre  $C$  and parallel to  $\pi$ .

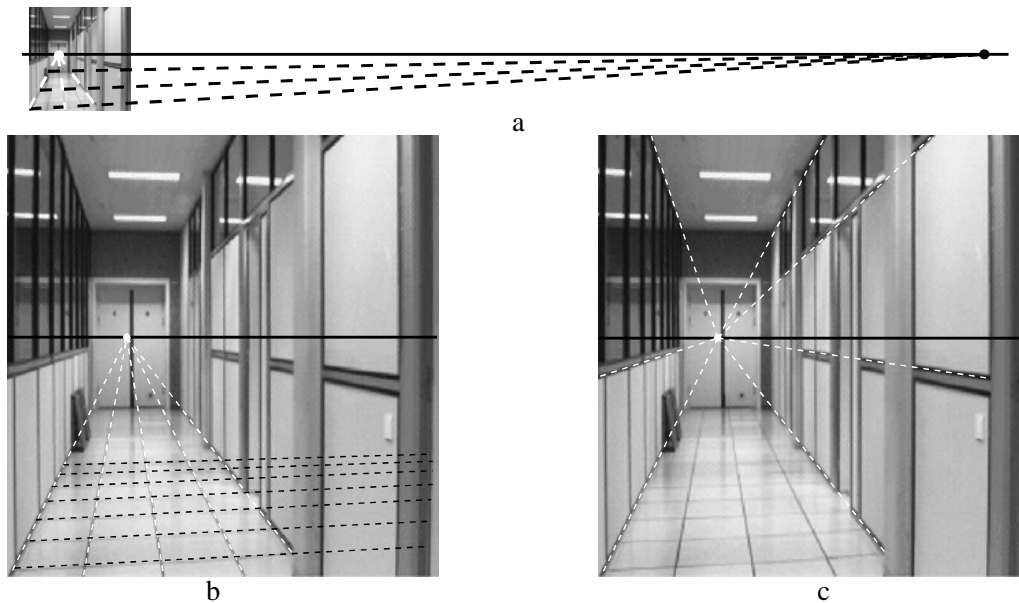


Fig. 8.17. **Vanishing points and lines.** The vanishing line of the ground plane (the horizon) of the corridor may be obtained from two sets of parallel lines on the plane. (a) The vanishing points of lines which are nearly parallel to the image plane are distant from the finite (actual) image. (b) Note the monotonic decrease in the spacing of the imaged equally spaced parallel lines corresponding to the sides of the floor tiles. (c) The vanishing point of lines parallel to a plane (here the ground plane) lies on the vanishing line of the plane.

direction  $\mathbf{n}$  intersects the image plane in the line  $\mathbf{l} = \mathbf{K}^{-\top} \mathbf{n}$ . Consequently,  $\mathbf{l}$  is the vanishing line of planes perpendicular to  $\mathbf{n}$ . Thus a plane with vanishing line  $\mathbf{l}$  has orientation  $\mathbf{n} = \mathbf{K}^{\top} \mathbf{l}$  in the camera's Euclidean coordinate frame.

- (ii) The plane may be metrically rectified given only its vanishing line. This can be seen by considering a synthetic rotation of the camera in the manner of example 8.13(p205). Since the plane normal is known from the vanishing line, the camera can be synthetically rotated by a homography so that the plane is fronto-parallel (i.e. parallel to the image plane). The computation of this homography is discussed in exercise (ix).
- (iii) The angle between two scene planes can be determined from their vanishing lines. Suppose the vanishing lines are  $\mathbf{l}_1$  and  $\mathbf{l}_2$ , then the angle  $\theta$  between the planes is given by

$$\cos \theta = \frac{\mathbf{l}_1^{\top} \boldsymbol{\omega}^* \mathbf{l}_2}{\sqrt{\mathbf{l}_1^{\top} \boldsymbol{\omega}^* \mathbf{l}_1} \sqrt{\mathbf{l}_2^{\top} \boldsymbol{\omega}^* \mathbf{l}_2}}. \quad (8.14)$$

The proof is left as an exercise.

### Computing vanishing lines

A common way to determine a vanishing line of a scene plane is first to determine vanishing points for two sets of lines parallel to the plane, and then to construct the line through the two vanishing points. This construction is illustrated in figure 8.17. Alternative methods of determining vanishing points are shown in example 2.19(p51) and example 2.20(p51).

However, the vanishing line may be determined directly, without using vanishing points as an intermediate step. For example, the vanishing line may be computed given an imaged set of equally spaced coplanar parallel lines. This is a useful method in practice because such sets commonly occur in man-made structures, such as: stairs, windows on the wall of a building, fences, radiators and zebra crossings. The following example illustrates the projective geometry involved.

#### Example 8.23. The vanishing line given the image of three coplanar equally spaced parallel lines

A set of equally spaced lines on the scene plane may be represented as  $ax' + by' + \lambda = 0$ , where  $\lambda$  takes integer values. This set (a pencil) of lines may be written as  $\mathbf{l}'_n = (a, b, n)^{\top} = (a, b, 0)^{\top} + n(0, 0, 1)^{\top}$ , where  $(0, 0, 1)^{\top}$  is the line at infinity on the scene plane. Under perspective imaging the point transformation is  $\mathbf{x} = \mathbf{H}\mathbf{x}'$ , and the corresponding line map is  $\mathbf{l}_n = \mathbf{H}^{-\top} \mathbf{l}'_n = \mathbf{l}_0 + n\mathbf{l}$ , where  $\mathbf{l}$ , the image of  $(0, 0, 1)^{\top}$ , is the vanishing line of the plane. The imaged geometry is shown in figure 8.18(c). Note all lines  $\mathbf{l}_n$  intersect in a common vanishing point (which is given by  $\mathbf{l}_i \times \mathbf{l}_j$ , for  $i \neq j$ ) and the spacing decreases monotonically with  $n$ . The vanishing line  $\mathbf{l}$  may be determined from three lines of the set provided their index ( $n$ ) is identified. For example, from the image of three equally spaced lines,  $\mathbf{l}_0$ ,  $\mathbf{l}_1$  and  $\mathbf{l}_2$ , the closed form solution for the vanishing line is:

$$\mathbf{l} = \left( (\mathbf{l}_0 \times \mathbf{l}_2)^{\top} (\mathbf{l}_1 \times \mathbf{l}_2) \right) \mathbf{l}_1 + 2 \left( (\mathbf{l}_0 \times \mathbf{l}_1)^{\top} (\mathbf{l}_2 \times \mathbf{l}_1) \right) \mathbf{l}_2. \quad (8.15)$$

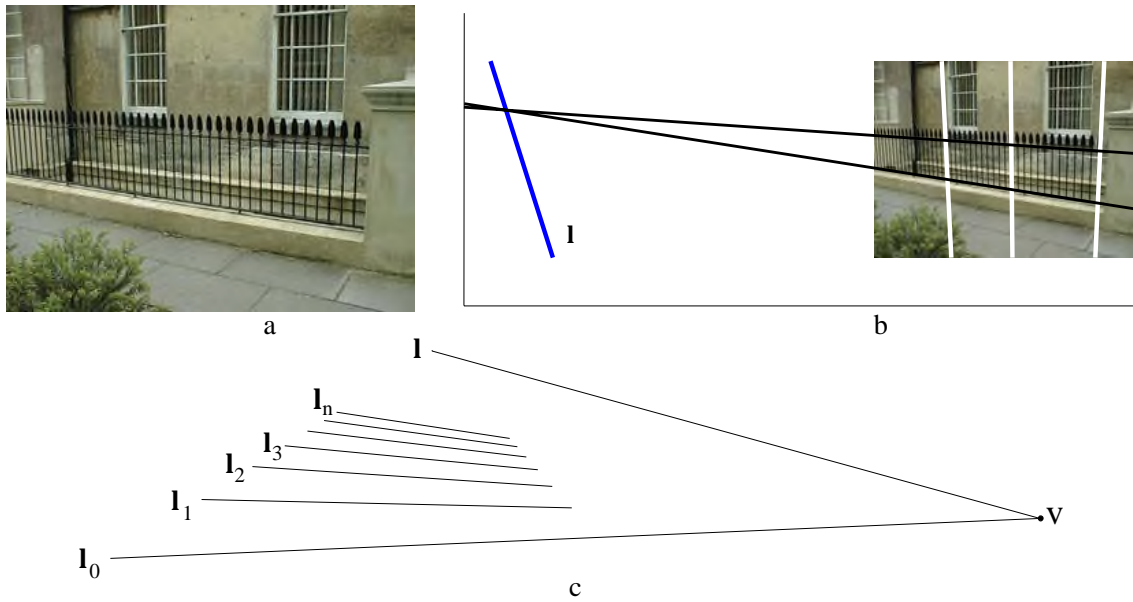


Fig. 8.18. **Determining a planes vanishing line from imaged equally spaced parallel lines.** (a) Image of a vertical fence with equally spaced bars. (b) The computed vanishing line  $l$  from three equally spaced bars (12 apart). Note the vanishing point of the horizontal lines lies on this vanishing line. (c) The spacing between the imaged lines  $l_n$  monotonically decreases with  $n$ .

The proof is left as an exercise. Figure 8.18(b) shows a vanishing line computed in this way.  $\triangle$

### 8.6.3 Orthogonality relationships amongst vanishing points and lines

It is often the case in practice that the lines and planes giving rise to vanishing points are orthogonal. In this case there are particularly simple relationships amongst their vanishing points and lines involving  $\omega$ , and furthermore these relations can be used to (partially) determine  $\omega$ , and consequently the camera calibration  $K$  as will be seen in section 8.8.

It follows from (8.13) that the vanishing points,  $v_1, v_2$ , of two perpendicular world lines satisfy  $v_1^T \omega v_2 = 0$ . This means that the vanishing points are conjugate with respect to  $\omega$ , as illustrated in figure 8.13. Similarly it follows from result 8.19 that the vanishing point  $v$  of a direction perpendicular to a plane with vanishing line  $l$  satisfies  $l = \omega v$ . This means that the vanishing point and line are in a pole–polar relation with respect to  $\omega$ , as is also illustrated in figure 8.13. Summarizing these image relations:

- (i) The vanishing points of lines with perpendicular directions satisfy

$$v_1^T \omega v_2 = 0. \quad (8.16)$$

- (ii) If a line is perpendicular to a plane then their respective vanishing point  $v$  and vanishing line  $l$  are related by

$$l = \omega v \quad (8.17)$$

and inversely  $v = \omega^* l$ .

- (iii) The vanishing lines of two perpendicular planes satisfy  $l_1^T \omega^* l_2 = 0$ .

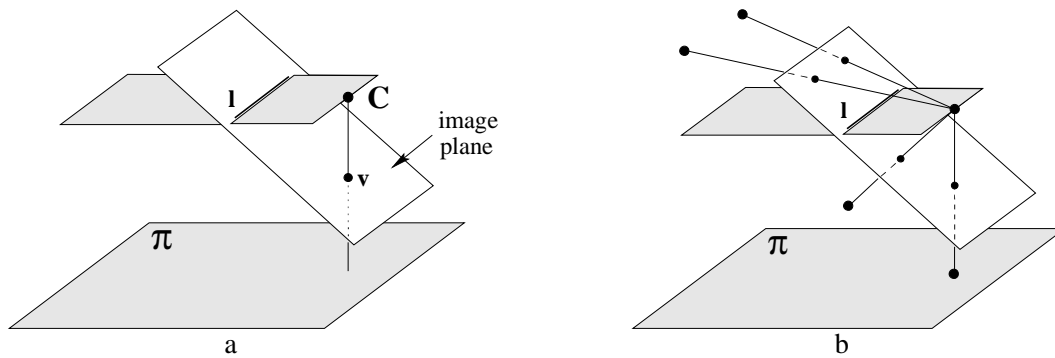


Fig. 8.19. **Geometry of a vertical vanishing point and ground plane vanishing line.** (a) The vertical vanishing point  $v$  is the image of the vertical “footprint” of the camera centre on the ground plane  $\pi$ . (b) The vanishing line  $l$  partitions all points in scene space. Any scene point projecting onto the vanishing line is at the same distance from the plane  $\pi$  as the camera centre; if it lies “above” the line it is farther from the plane, and if “below” then it is closer to the plane than the camera centre.

For example, suppose the vanishing line  $l$  of the ground plane (the horizon) is identified in an image, and the internal calibration matrix  $K$  is known, then the vertical vanishing point  $v$  (which is the vanishing point of the normal direction to the plane) may be obtained from  $v = \omega^*l$ .

### 8.7 Affine 3D measurements and reconstruction

It has been seen in section 2.7.2(p49) that identifying a scene plane’s vanishing line allows affine properties of the scene plane to be measured. If in addition a vanishing point for a direction not parallel to the plane is identified, then affine properties can be computed for the 3-space of the perspectively imaged scene. We will illustrate this idea for the case where the vanishing point corresponds to a direction orthogonal to the plane, although orthogonality is not necessary for the construction. The method described in this section does not require that the internal calibration of the camera  $K$  be known.

It will be convenient to think of the scene plane as the horizontal ground plane, in which case the vanishing line is the *horizon*. Similarly, it will be convenient to think of the direction orthogonal to the scene plane as vertical, so that  $v$  is the vertical vanishing point. This situation is illustrated in figure 8.19.

Suppose we wish to measure the relative lengths of two line segments in the vertical direction as shown in figure 8.20(a). We will show the following result:

**Result 8.24.** *Given the vanishing line of the ground plane  $l$  and the vertical vanishing point  $v$ , then the relative length of vertical line segments can be measured provided their end point lies on the ground plane.*

Clearly the relative lengths cannot be measured directly from their imaged lengths because as a vertical line recedes deeper into the scene (i.e. further from the camera) then its imaged length decreases. The construction to determine the relative lengths proceeds in two steps:

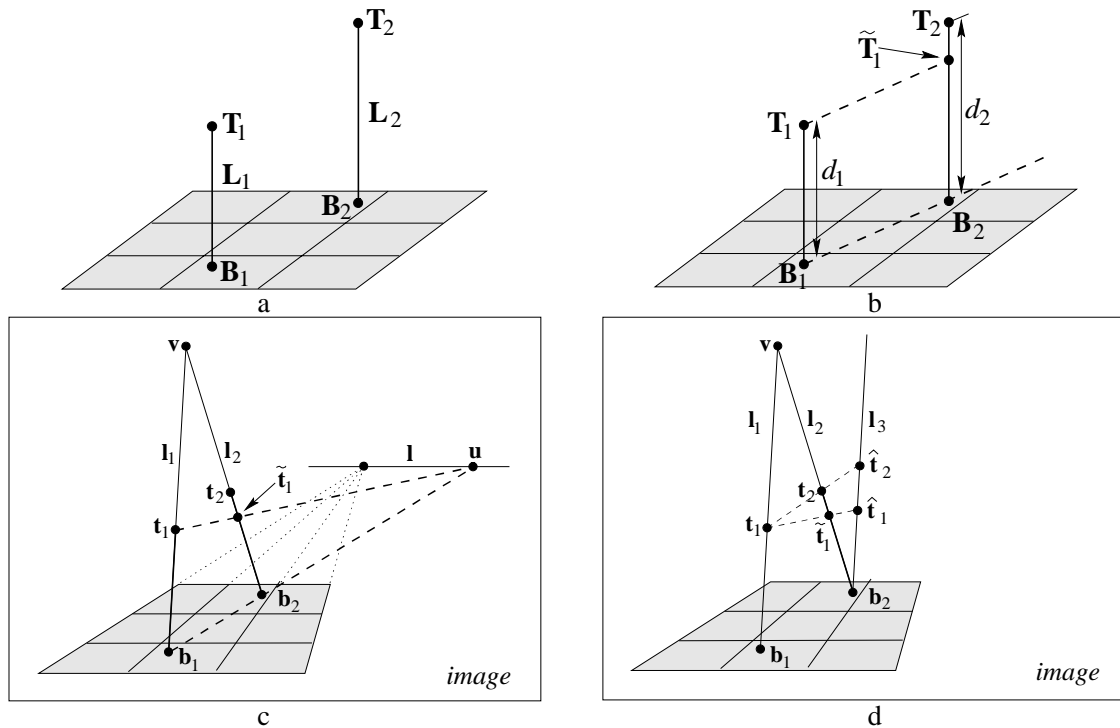


Fig. 8.20. **Computing length ratios of parallel scene lines.** (a) **3D geometry:** The vertical line segments  $L_1 = \langle B_1, T_1 \rangle$  and  $L_2 = \langle B_2, T_2 \rangle$  have length  $d_1$  and  $d_2$  respectively. The base points  $B_1, B_2$  are on the ground plane. We wish to compute the scene length ratio  $d_1 : d_2$  from the imaged configuration. (b) In the scene the length of the line segment  $L_1$  may be transferred to  $L_2$  by constructing a line parallel to the ground plane to generate the point  $\tilde{T}_1$ . (c) **Image geometry:**  $l$  is the ground plane vanishing line, and  $v$  the vertical vanishing point. A corresponding parallel line construction in the image requires first determining the vanishing point  $u$  from the images  $b_i$  of  $B_i$ , and then determining  $\tilde{t}_1$  (the image of  $\tilde{T}_1$ ) by the intersection of  $l_2$  and the line  $\langle t_1, u \rangle$ . (d) The line  $l_3$  is parallel to  $l_1$  in the image. The points  $\hat{t}_1$  and  $\hat{t}_2$  are constructed by intersecting  $l_3$  with the lines  $\langle t_1, \tilde{t}_1 \rangle$  and  $\langle t_2, \tilde{t}_2 \rangle$  respectively. The distance ratio  $d(b_2, \hat{t}_1) : d(b_2, \hat{t}_2)$  is the computed estimate of  $d_1 : d_2$ .

**Step 1: Map the length of one line segment onto the other.** In 3D the length of  $L_1$  may be compared to  $L_2$  by constructing a line parallel to the ground plane in the direction  $\langle B_1, B_2 \rangle$  that transfers  $T_1$  onto  $L_2$ . This transferred point will be denoted  $\tilde{T}_1$  (see figure 8.20(b)). In the image a corresponding construction is carried out by first determining the vanishing point  $u$  which is the intersection of  $\langle b_1, b_2 \rangle$  with  $l$ . Now any scene line parallel to  $\langle B_1, B_2 \rangle$  is imaged as a line through  $u$ , so in particular the image of the line through  $T_1$  parallel to  $\langle B_1, B_2 \rangle$  is the line through  $t_1$  and  $u$ . The intersection of the line  $\langle t_1, u \rangle$  with  $l_2$  defines the image  $\tilde{t}_1$  of the transferred point  $\tilde{T}_1$  (see figure 8.20(c)).

**Step 2: Determine the ratio of lengths on the scene line.** We now have four collinear points on an imaged scene line and wish to determine the actual length ratio in the scene. The four collinear image points are  $b_2, \tilde{t}_1, t_2$  and  $v$ . These may be treated as images of scene points at distances  $0, d_1, d_2$  and  $\infty$ , respectively, along the scene line. The affine ratio  $d_1 : d_2$  may be obtained by applying a projective transfor-

Objective

Given the vanishing line of the ground plane  $\mathbf{l}$  and the vertical vanishing point  $\mathbf{v}$  and the top  $(\mathbf{t}_1, \mathbf{t}_2)$  and base  $(\mathbf{b}_1, \mathbf{b}_2)$  points of two line segments as in figure 8.20, compute the ratio of lengths of the line segments in the scene.

Algorithm

- (i) Compute the vanishing point  $\mathbf{u} = (\mathbf{b}_1 \times \mathbf{b}_2) \times \mathbf{l}$ .
- (ii) Compute the transferred point  $\tilde{\mathbf{t}}_1 = (\mathbf{t}_1 \times \mathbf{u}) \times \mathbf{l}_2$  (where  $\mathbf{l}_2 = \mathbf{v} \times \mathbf{b}_2$ ).
- (iii) Represent the four points  $\mathbf{b}_2, \tilde{\mathbf{t}}_1, \mathbf{t}_2$  and  $\mathbf{v}$  on the image line  $\mathbf{l}_1$  by their distance from  $\mathbf{b}_2$ , as  $0, \tilde{t}_1, t_2$  and  $v$  respectively.
- (iv) Compute a 1D projective transformation  $\mathbb{H}_{2 \times 2}$  mapping homogeneous coordinates  $(0, 1) \mapsto (0, 1)$  and  $(v, 1) \mapsto (1, 0)$  (which maps the vanishing point  $\mathbf{v}$  to infinity). A suitable matrix is given by

$$\mathbb{H}_{2 \times 2} = \begin{bmatrix} 1 & 0 \\ 1 & -v \end{bmatrix}.$$

- (v) The (scaled) distance of the scene points  $\tilde{\mathbf{T}}_1$  and  $\mathbf{T}_2$  from  $\mathbf{B}_2$  on  $\mathbf{L}_2$  may then be obtained from the position of the points  $\mathbb{H}_{2 \times 2}(\tilde{t}_1, 1)^\top$  and  $\mathbb{H}_{2 \times 2}(t_2, 1)^\top$ . Their distance ratio is then given by

$$\frac{d_1}{d_2} = \frac{\tilde{t}_1(v - t_2)}{t_2(v - \tilde{t}_1)}$$

Algorithm 8.1. *Computing scene length ratios from a single image.*

mation to the image line which maps  $\mathbf{v}$  to infinity. A geometric construction of this projectivity is shown in figure 8.20(d) (see example 2.20(p51)).

Details of the algorithm to carry out these two steps are given in algorithm 8.1.

Note, no knowledge of the camera calibration  $\mathbf{K}$  or pose is necessary to apply the algorithm. In fact, the position of the camera centre relative to the ground plane can also be computed. The algorithm is well conditioned even when the vanishing point and/or line are at infinity in the image. For example, under affine image conditionings, or if the image plane is parallel to the vertical scene direction (so that  $\mathbf{v}$  is at infinity). In these cases the distance ratio simplifies to  $\frac{d_1}{d_2} = \frac{\tilde{t}_1}{t_2}$ .

**Example 8.25. Measuring a person's height in a single image**

Suppose we have an image which contains sufficient information to compute the ground plane vanishing line and the vertical vanishing point, and also one object of known height for which the top and base are imaged. Then the height of a person standing on the ground plane can be measured anywhere in the scene provided that their head and feet are both visible. Figure 8.21(a) shows an example. The scene contains plenty of horizontal lines from which to compute a horizontal vanishing point. Two such vanishing points determine the vanishing line of the floor (which is the horizon for this image). The scene also contains plenty of vertical lines from which to compute a vertical vanishing point (figure 8.21(c)). Assuming that the two people are standing vertically, then their relative height may be computed directly from their length ratio using algorithm 8.1. Their absolute height may be determined by comput-

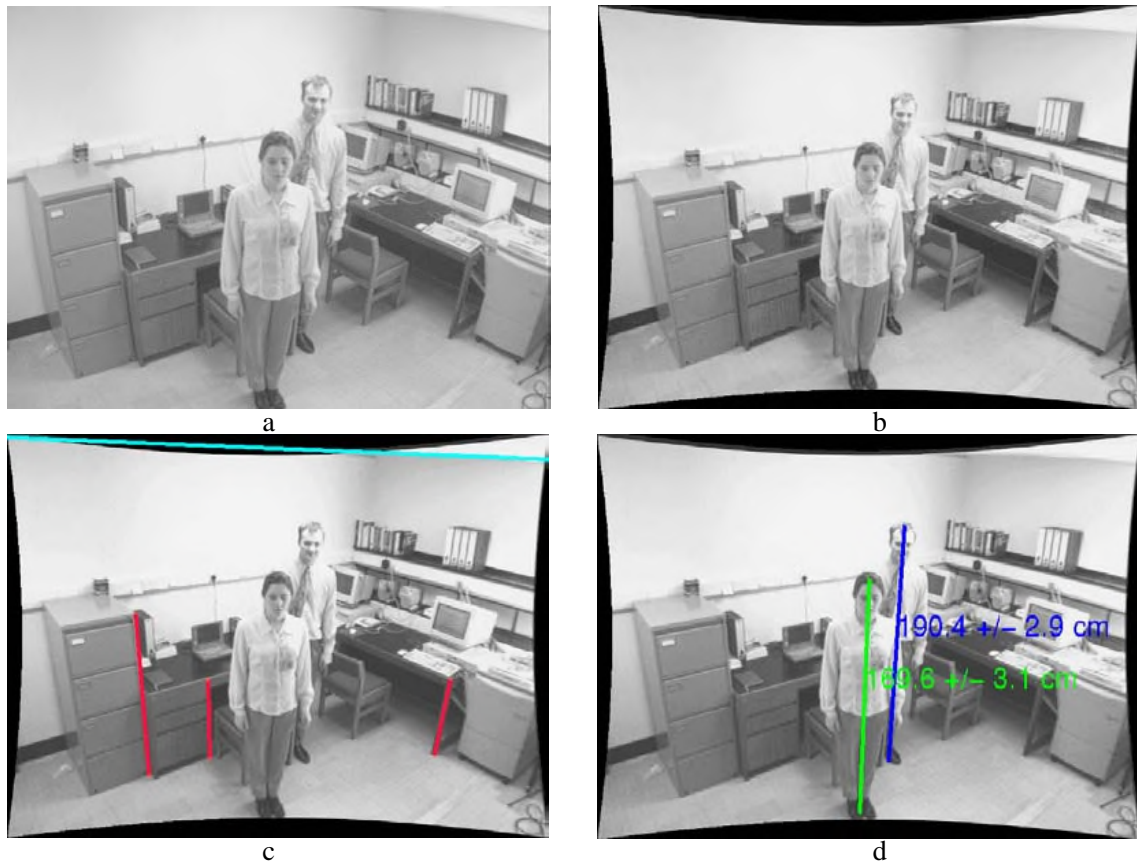


Fig. 8.21. **Height measurements using affine properties.** (a) The original image. We wish to measure the height of the two people. (b) The image after radial distortion correction (see section 7.4(p189)). (c) The vanishing line (shown) is computed from two vanishing points corresponding to horizontal directions. The lines used to compute the vertical vanishing points are also shown. The vertical vanishing point is not shown since it lies well below the image. (d) Using the known height of the filing cabinet on the left of the image, the absolute height of the two people are measured as described in algorithm 8.1. The measured heights are within 2cm of ground truth. The computation of the uncertainty is described in [Criminisi-00].

ing their height relative to an object on the ground plane with known height. Here the known height is provided by the filing cabinet. The result is shown in figure 8.21(d).

△

### 8.8 Determining camera calibration $K$ from a single view

We have seen that once  $\omega$  is known the angle between rays can be measured. Conversely if the angle between rays is known then a constraint is placed on  $\omega$ . Each known angle between two rays gives a constraint of the form (8.13) on  $\omega$ . Unfortunately, for arbitrary angles, and known  $\mathbf{v}_1$  and  $\mathbf{v}_2$ , this gives a quadratic constraint on the entries of  $\omega$ . If the lines are perpendicular, however, (8.13) reduces to (8.16)  $\mathbf{v}_1^\top \omega \mathbf{v}_2 = 0$ , and the constraint on  $\omega$  is linear.

A linear constraint on  $\omega$  also results from a vanishing point and vanishing line arising from a line and its orthogonal plane. A common example is a vertical direction and horizontal plane as in figure 8.19. From (8.17)  $\mathbf{l} = \omega \mathbf{v}$ . Writing this as  $\mathbf{l} \times (\omega \mathbf{v}) = \mathbf{0}$

| Condition                                                                                                     | constraint                                                                                                                                                            | type   | # constraints |
|---------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|---------------|
| vanishing points $\mathbf{v}_1, \mathbf{v}_2$<br>corresponding to orthogonal lines                            | $\mathbf{v}_1^\top \boldsymbol{\omega} \mathbf{v}_2 = 0$                                                                                                              | linear | 1             |
| vanishing point $\mathbf{v}$ and vanishing<br>line $\mathbf{l}$ corresponding to<br>orthogonal line and plane | $[\mathbf{l}]_\times \boldsymbol{\omega} \mathbf{v} = \mathbf{0}$                                                                                                     | linear | 2             |
| metric plane imaged with known<br>homography $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3]$        | $\mathbf{h}_1^\top \boldsymbol{\omega} \mathbf{h}_2 = 0$<br>$\mathbf{h}_1^\top \boldsymbol{\omega} \mathbf{h}_1 = \mathbf{h}_2^\top \boldsymbol{\omega} \mathbf{h}_2$ | linear | 2             |
| zero skew                                                                                                     | $\omega_{12} = \omega_{21} = 0$                                                                                                                                       | linear | 1             |
| square pixels                                                                                                 | $\omega_{12} = \omega_{21} = 0$<br>$\omega_{11} = \omega_{22}$                                                                                                        | linear | 2             |

Table 8.1. Scene and internal constraints on  $\boldsymbol{\omega}$ .

removes the homogeneous scaling factor and results in three homogeneous equations linear in the entries of  $\boldsymbol{\omega}$ . These are equivalent to two independent constraints on  $\boldsymbol{\omega}$ .

All these conditions provide linear constraints on  $\boldsymbol{\omega}$ . Given a sufficient number of such constraints  $\boldsymbol{\omega}$  may be computed and hence the camera calibration  $\mathbf{K}$  also follows since  $\boldsymbol{\omega} = (\mathbf{K}\mathbf{K}^\top)^{-1}$ .

The number of entries of  $\boldsymbol{\omega}$  that need be determined from scene constraints of this sort can be reduced if the calibration matrix  $\mathbf{K}$  has a more specialized form than (6.10–p157). In the case where  $\mathbf{K}$  is known to have zero skew ( $s = 0$ ), or square pixels ( $\alpha_x = \alpha_y$  and  $s = 0$ ), we can take advantage of this condition to help find  $\boldsymbol{\omega}$ . In particular, it is quickly verified by direct computation that:

**Result 8.26.** *If  $s = \mathbf{K}_{12} = 0$  then  $\omega_{12} = \omega_{21} = 0$ . If in addition  $\alpha_x = \mathbf{K}_{11} = \mathbf{K}_{22} = \alpha_y$ , then  $\omega_{11} = \omega_{22}$ .*

Thus, in solving for the image of the absolute conic, one may easily take into account the zero-skew or square-aspect ratio constraint on the camera, if such a constraint is known to exist. One may also verify that no such simple connection as result 8.26 exists between the entries of  $\mathbf{K}$  and those of  $\boldsymbol{\omega}^* = \mathbf{K}\mathbf{K}^\top$ .

We have now seen three sources of constraints on  $\boldsymbol{\omega}$ :

- (i) metric information on a plane imaged with a known homography, see (8.12–p211)
- (ii) vanishing points and lines corresponding to perpendicular directions and planes, (8.16)
- (iii) “internal constraints” such as zero skew or square pixels, as in result 8.26

These constraints are summarized in table 8.1. We now describe how these constraints may be combined to estimate  $\boldsymbol{\omega}$  and thence  $\mathbf{K}$ .

Since all the above constraints (including the internal constraints) are described algebraically as linear equations on  $\boldsymbol{\omega}$ , it is a simple matter to combine them as rows of

Objective

Compute  $K$  via  $\omega$  by combining scene and internal constraints.

Algorithm

- (i) Represent  $\omega$  as a homogeneous 6-vector  $\mathbf{w} = (w_1, w_2, w_3, w_4, w_5, w_6)^T$  where:

$$\omega = \begin{bmatrix} w_1 & w_2 & w_4 \\ w_2 & w_3 & w_5 \\ w_4 & w_5 & w_6 \end{bmatrix}$$

- (ii) Each available constraint from table 8.1 may be written as  $\mathbf{a}^T \mathbf{w} = 0$ . For example, for the orthogonality constraint  $\mathbf{u}^T \omega \mathbf{v} = 0$ , where  $\mathbf{u} = (u_1, u_2, u_3)^T$  and  $\mathbf{v} = (v_1, v_2, v_3)^T$ , the 6-vector  $\mathbf{a}$  is given by

$$\mathbf{a} = (v_1 u_1, v_1 u_2 + v_2 u_1, v_2 u_2, v_1 u_3 + v_3 u_1, v_2 u_3 + v_3 u_2, v_3 u_3)^T.$$

Similar constraints vectors are obtained from the other sources of scene and internal constraints. For example a metric plane generates two such constraints.

- (iii) Stack the equations  $\mathbf{a}^T \mathbf{w} = 0$  from each constraint in the form  $A\mathbf{w} = \mathbf{0}$ , where  $A$  is a  $n \times 6$  matrix for  $n$  constraints.
- (iv) Solve for  $\mathbf{w}$  using the SVD as in algorithm 4.2(p109). This determines  $\omega$ .
- (v) Decompose  $\omega$  into  $K$  using matrix inversion and Cholesky factorization (see section A4.2.1(p582)).

Algorithm 8.2. *Computing  $K$  from scene and internal constraints.*

a constraint matrix. All constraints may be collected together so that for  $n$  constraints the system of equations may be written as  $A\mathbf{w} = \mathbf{0}$ , where  $A$  is a  $n \times 6$  matrix and  $\mathbf{w}$  is a 6-vector containing the six distinct homogeneous entries of  $\omega$ . With a minimum of 5 constraint equations an exact solution is found. With more than five equations, a least-squares solution is found by algorithm A5.4(p593). The method is summarized in algorithm 8.2.

With more than the minimum required five constraints, we have the option to apply some of the constraints as hard constraints – that is, constraints that will be satisfied exactly. This can be done by parametrizing  $\omega$  so that the constraints are satisfied explicitly (for instance setting  $\omega_{21} = \omega_{12} = 0$  for the zero skew constraint, and also  $\omega_{11} = \omega_{22}$  for the square-pixel constraint). The minimization method of algorithm A5.5(p594) may also be used to enforce hard constraints. Otherwise, treating all constraints as soft constraints and using algorithm A5.4(p593) will produce a solution in which the constraints are not satisfied exactly in the presence of noise – for instance, pixels may not be quite square.

Finally, an important issue in practice is that of degeneracy. This occurs when the combined constraints are not independent and results in the matrix  $A$  dropping rank. If the rank is less than the number of unknowns, then a parametrized family of solutions for  $\omega$  (and hence  $K$ ) is obtained. Also, if conditions are near degenerate then the solution is ill-conditioned and the particular member of the family is determined by “noise”. These degeneracies can often be understood geometrically – for example in example 8.18 if the three metric planes are parallel then the three pairs of imaged

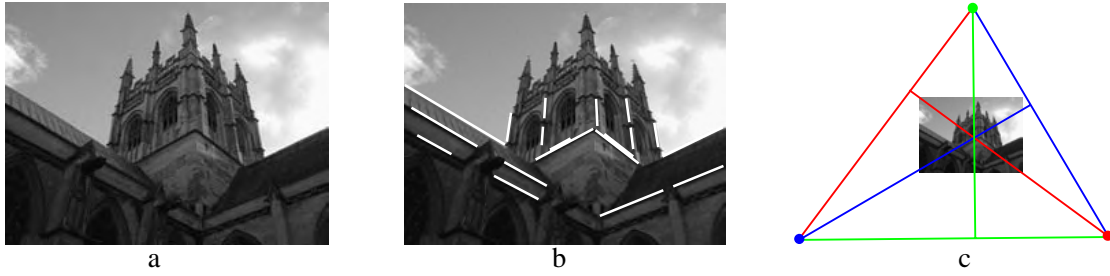


Fig. 8.22. For the case that image skew is zero and the aspect ratio unity **the principal point is the orthocentre of an orthogonal triad of vanishing points**. (a) Original image. (b) Three sets of parallel lines in the scene, with each set having direction orthogonal to the others. (c) The principal point is the orthocentre of the triangle with the vanishing points as vertices.

circular points are coincident and only provide a total of two constraints instead of six. A pragmatic solution to the problem of degeneracy, popularized by Zhang [Zhang-00], is to image a metric plane many times in varying positions. This reduces the chances of degeneracy occurring, and also provides a very over-determined solution.

### Example 8.27. Calibration from three orthogonal vanishing points

Suppose that it is known that the camera has zero skew, and that the pixels are square (or equivalently their aspect ratio is known). A triad of orthogonal vanishing point directions supplies three more constraints. This gives a total of 5 constraints – sufficient to compute  $\omega$ , and hence  $K$ .

In outline the algorithm has the following steps:

- (i) In the case of square pixels  $\omega$  has the form

$$\omega = \begin{bmatrix} w_1 & 0 & w_2 \\ 0 & w_1 & w_3 \\ w_2 & w_3 & w_4 \end{bmatrix}.$$

- (ii) Each pair of vanishing points  $\mathbf{v}_i, \mathbf{v}_j$  generates an equation  $\mathbf{v}_i^T \omega \mathbf{v}_j = 0$ , which is linear in the elements of  $\omega$ . The constraints from the three pairs of vanishing points are stacked together to form an equation  $A\mathbf{w} = \mathbf{0}$ , where  $A$  is a  $3 \times 4$  matrix.
- (iii) The vector  $\mathbf{w}$  is obtained as the null vector of  $A$ , and this determines  $\omega$ . The matrix  $K$  is obtained from  $\omega = (KK^T)^{-1}$  by Cholesky factorization of  $\omega$ , followed by inversion.

An example is shown in figure 8.22(a). Vanishing points are computed corresponding to the three perpendicular directions shown in figure 8.22(b). The image is  $1024 \times 768$  pixels, and the calibration matrix is computed to be

$$K = \begin{bmatrix} 1163 & 0 & 548 \\ 0 & 1163 & 404 \\ 0 & 0 & 1 \end{bmatrix}.$$

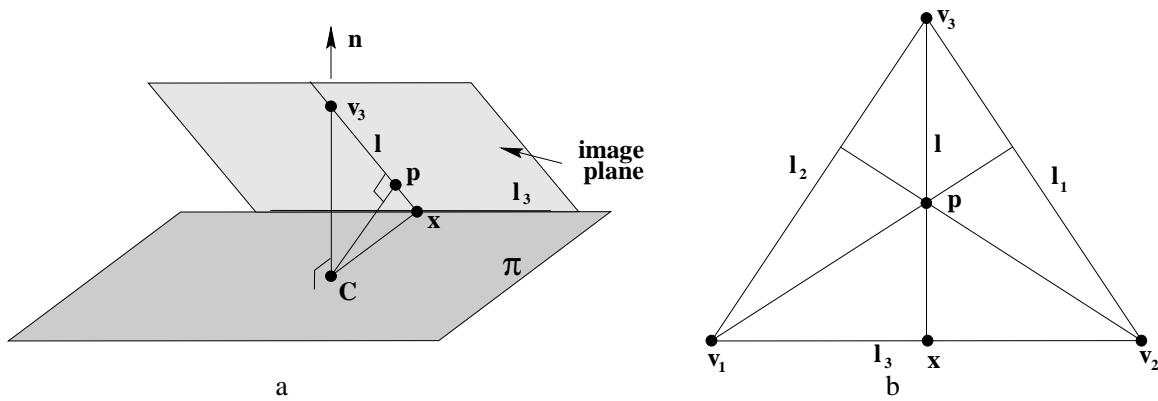


Fig. 8.23. **Geometric construction of the principal point.** The vanishing line  $l_3$  back-projects to a plane  $\pi$  with normal  $\mathbf{n}$ . The vanishing point  $\mathbf{v}_3$  back-projects to a line orthogonal to the plane  $\pi$ . (a) The normal  $\mathbf{n}$  of the plane  $\pi$  through the camera centre  $\mathbf{C}$  and the principal axis define a plane, which intersects the image in the line  $\mathbf{l} = \langle \mathbf{v}_3, \mathbf{x} \rangle$ . The line  $l_3$  is the intersection of  $\pi$  with the image plane, and is also its vanishing line. The point  $\mathbf{v}_3$  is the intersection of the normal with the image plane, and is also its vanishing point. Clearly the principal point lies on  $\mathbf{l}$ , and  $\mathbf{l}$  and  $l_3$  are perpendicular on the image plane. (b) The principal point may be determined from three such constraints as the orthocentre of the triangle.

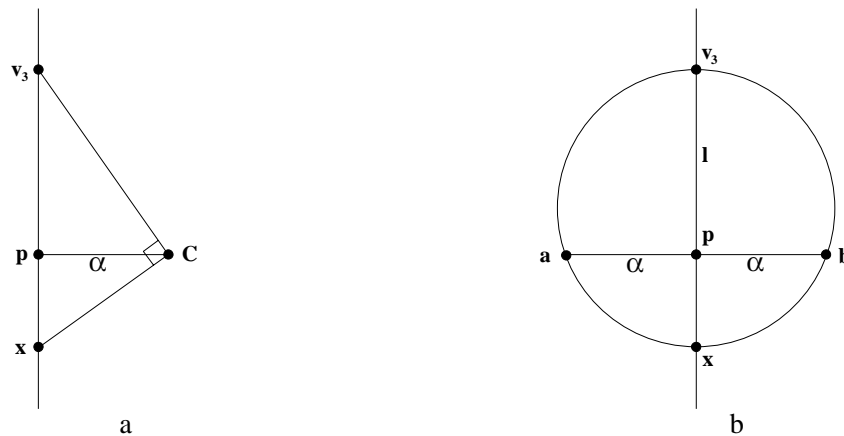


Fig. 8.24. **Geometric construction of the focal length.** (a) Consider the plane defined by the camera centre  $\mathbf{C}$ , principal point and one of the vanishing points, e.g.  $\mathbf{v}_3$  as shown in figure 8.23(a). The rays from  $\mathbf{C}$  to  $\mathbf{v}_3$  and  $\mathbf{x}$  are perpendicular to each other. The focal length,  $\alpha$ , is the distance from the camera centre to the image plane. By similar triangles,  $\alpha^2 = d(\mathbf{p}, \mathbf{v}_3)d(\mathbf{p}, \mathbf{x})$ , where  $d(\mathbf{u}, \mathbf{v})$  is the distance between the points  $\mathbf{u}$  and  $\mathbf{v}$ . (b) In the image a circle is drawn with diameter the line between  $\mathbf{v}_3$  and  $\mathbf{x}$ . A line through  $\mathbf{p}$  perpendicular to  $\langle \mathbf{v}_3, \mathbf{x} \rangle$  meets the circle in two points  $\mathbf{a}$  and  $\mathbf{b}$ . The focal length equals the distance  $d(\mathbf{p}, \mathbf{a})$ .

The principal point and focal length may also be computed geometrically in this case. The principal point is the orthocentre of the triangle with vertices the vanishing points. Figure 8.23 shows that the principal point lies on the perpendicular line from one triangle side to the opposite vertex. A similar construction for the other two sides shows that the principal point is the orthocentre. An algebraic derivation of this result is left to the exercises. The focal length can also be computed geometrically as shown in figure 8.24.

As a cautionary note, this estimation method is degenerate if one of the vanishing



Fig. 8.25. **Plane rectification via partial internal parameters** (a) Original image. (b) Rectification assuming the camera has square pixels and principal point at the centre of the image. The focal length is computed from the single orthogonal vanishing point pair. The aspect ratio of a window in the rectified image differs from the ground truth value by 3.7%. Note that the two parallel planes, the upper building facade and the lower shopfront, are both mapped to fronto-parallel planes.

points, say the vertical, is at infinity. In this case  $A$  drops rank to two, and there is a one-parameter family of solutions for  $\omega$  and correspondingly for  $K$ . This degeneracy can be seen geometrically from the orthocentre construction of figure 8.23. If  $\mathbf{v}_3$  is at infinity then the principal point  $\mathbf{p}$  lies on the line  $l_3 = \langle \mathbf{v}_1, \mathbf{v}_2 \rangle$ , but its  $x$  position is not defined.

### Example 8.28. Determining the focal length when the other internal parameters are known

We consider a further example of calibration from a single view. Suppose that it is known that the camera has zero skew, that the pixels are square (or equivalently their aspect ratio is known), and also that the principal point is at the image centre. Then only the focal length is unknown. In this case, the form of  $\omega$  is very simple: it is a diagonal matrix  $\text{diag}(1/f^2, 1/f^2, 1)$  with only one degree of freedom. Using algorithm 8.2, the focal length  $f$  may be determined from one further constraint, such as the one arising from two vanishing points corresponding to orthogonal directions.

An example is shown in figure 8.25(a). Here the vanishing points used in the constraint are computed from the horizontal edges of the windows and pavement, and the vertical edges of the windows. These vanishing points also determine the vanishing line  $l$  of the building facade. Given  $K$  and the vanishing line  $l$ , the camera can be synthetically rotated such that the facade is fronto-parallel by mapping the image with a homography as in example 8.13(p205). The result is shown in figure 8.25(b). Note, in example 8.13 it was necessary to know the aspect ratio of a rectangle on the scene plane in order to rectify the plane. Here it is only necessary to know the vanishing line of the plane because the camera calibration  $K$  provides the additional information required for the homography.  $\triangle$

#### 8.8.1 The geometry of the constraints

Although the algebraic constraints given in table 8.1 appear to arise from distinct sources, they are in fact all equivalent to one of two simple geometric relations: two points lying on the conic  $\omega$ , or conjugacy of two points with respect to  $\omega$ .

For example, the zero skew constraint is an orthogonality constraint: it specifies that

the image  $x$  and  $y$  axes are orthogonal. These axes correspond to rays with directions in the camera's Euclidean coordinate frame,  $(1, 0, 0)^\top$  and  $(0, 1, 0)^\top$ , respectively, that are imaged at  $\mathbf{v}_x = (1, 0, 0)^\top$  and  $\mathbf{v}_y = (0, 1, 0)^\top$  (since the rays are parallel to the image plane). The zero skew constraint  $\omega_{12} = \omega_{21} = 0$  is just another way of writing the orthogonality constraint (8.16)  $\mathbf{v}_y^\top \boldsymbol{\omega} \mathbf{v}_x = 0$ . Geometrically skew zero is equivalent to conjugacy of the points  $(1, 0, 0)^\top$  and  $(0, 1, 0)^\top$  with respect to  $\boldsymbol{\omega}$ .

The square pixel constraint may be interpreted in two ways. A square has the property of defining two sets of orthogonal lines: adjacent edges are orthogonal, and so are the two diagonals. Thus, the square pixel constraint may be interpreted as a pair of orthogonal line constraints. The diagonal vanishing points of a square pixel are  $(1, 1, 0)^\top$  and  $(-1, 1, 0)^\top$ . The resulting orthogonality constraints lead to the square pixel constraints given in table 8.1.

Alternatively, the square pixel constraint can be interpreted in terms of two known points lying on the IAC. If the image plane has square pixels, then it has a Euclidean coordinate system and the circular points have known coordinates  $(1, \pm i, 0)^\top$ . It may be verified that the two square pixel equations are equivalent to  $(1, \pm i, 0)^\top \boldsymbol{\omega} (1, \pm i, 0)^\top = 0$ .

This is the most important geometric equivalence. In essence an image plane with square pixels acts as a metric plane in the scene. A square pixel image plane is equivalent to a metric plane imaged with a homography given by the identity. Indeed if the homography  $H$  in the "metric plane imaged with known homography" constraint of table 8.1 is replaced by the identity then the square pixel constraints are immediately obtained.

Thus, we see that all the constraints given in table 8.1 are derived either from known points lying on  $\boldsymbol{\omega}$ , or from pairs of points that are conjugate with respect to  $\boldsymbol{\omega}$ . Determining  $\boldsymbol{\omega}$  may therefore be viewed as a conic fitting problem, given points on the conic and conjugate point pairs.

It is well to bear in mind that conic fitting is a delicate problem, often unstable ([Bookstein-79]) if the points are not well distributed on the conic. The same observation is true of the present problem, which we have seen is equivalent to conic fitting. The method given in algorithm 8.2 for finding the calibration from vanishing points amounts to minimization of algebraic error, and therefore does not give an optimal solution. For greater accuracy, the methods of chapter 4, for instance the Sampson error method of section 4.2.6(p98) should be used.

## 8.9 Single view reconstruction

As an application of the methods developed in this chapter we demonstrate now the 3D reconstruction of a texture mapped piecewise planar graphical model from a single image. The camera calibration methods of section 8.8 and the rectification method of example 8.28 may be combined to back project image regions to texture the planes of the model.

The method will be illustrated for the image of figure 8.26(a), where the scene contains three dominant and mutually orthogonal planes: the building facades on the left and right and the ground plane. The parallel line sets in three orthogonal directions de-

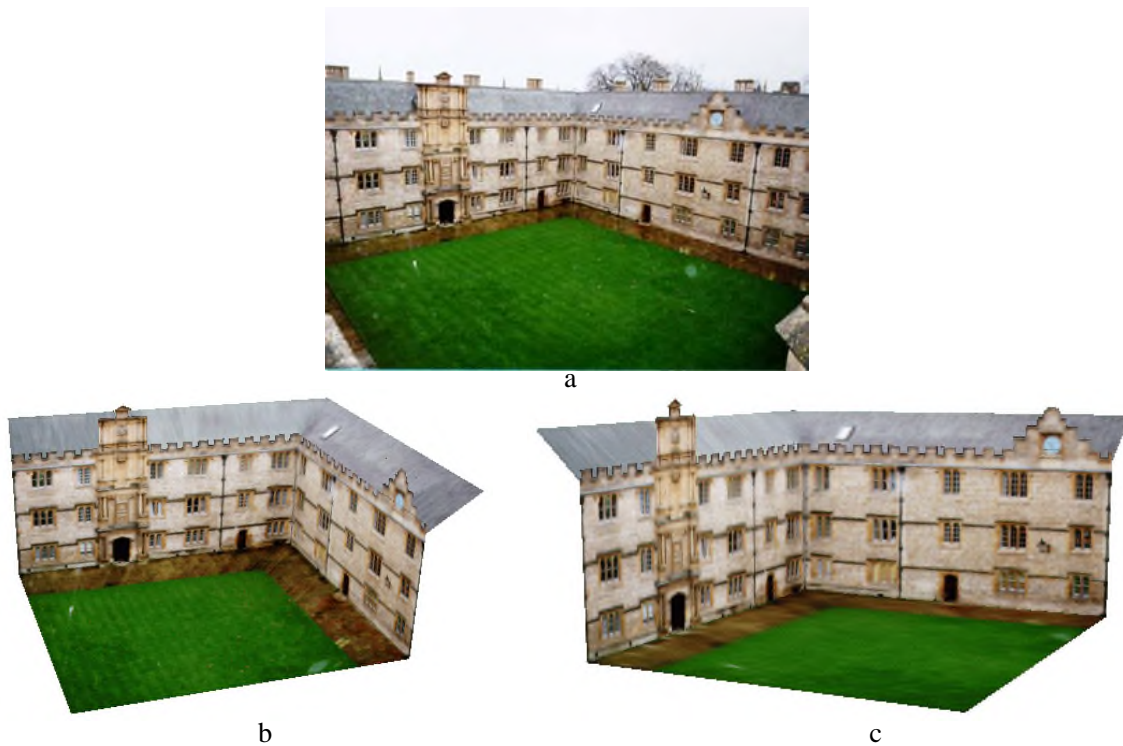


Fig. 8.26. **Single view reconstruction.** (a) Original image of the Fellows quad, Merton College, Oxford. (b) (c) Views of the 3D model created from the single image. The vanishing line of the roof planes is computed from the repetition of the texture pattern.

fine three vanishing points and together with the constraint of square pixels the camera calibration may be computed using the method described in section 8.8. From the vanishing lines of the three planes, likewise determined by the vanishing points, together with the computed  $\omega$ , homographies may be computed to texture map the appropriate image regions onto the orthogonal planes of the model.

In more detail taking the left facade as a reference plane in figure 8.26(a), its correctly proportioned width and height are determined by the rectification. The right facade and ground planes define 3D planes orthogonal to the reference (we have assumed the orthogonality of the planes in computing the camera, so relative orientations are defined). Scaling of the right and ground planes is computed from the points common to the planes and this completes a three orthogonal plane model.

Having computed the calibration, the relative orientation of planes in the scene that are not orthogonal (such as the roof) can be computed if their vanishing lines can be found using (8.14–p218). Their relative positions and dimensions can be determined if the intersection of a pair of planes is visible in the image, so that there are points common to both planes. Relative size can be computed from the rectification of a distance between common points using the homographies of both planes. Views of the model, with texture mapped correctly to the planes, appear in figure 8.26(b) and (c).

### 8.10 The calibrating conic

The image of the absolute conic (IAC) is an imaginary conic in an image, and hence is not visible. Sometimes it is useful for visualization purposes to consider a different conic that is closely related to the calibration of the camera. Such a conic is the *calibrating conic*, which is the image of a cone with apex angle  $45^\circ$  and axis coinciding with the principal axis of the camera.

We wish to compute a formula for this cone in terms of the calibration matrix of the camera. Since the  $45^\circ$  cone moves with the camera, its image is clearly independent of the orientation and position of the camera. Thus, we may assume that the camera is located at the origin and oriented directly along the Z-axis. Thus, let the camera matrix be  $P = K[I \mid \mathbf{0}]$ . Now, any point on the  $45^\circ$  cone satisfies  $X^2 + Y^2 = Z^2$ . Points on this cone map to points on the conic

$$C = K^{-T} \begin{bmatrix} 1 & & \\ & 1 & \\ & & -1 \end{bmatrix} K^{-1} \quad (8.18)$$

as one easily verifies from result 8.6(p199). This conic will be referred to as the *calibrating conic* of the camera. For a calibrated camera with identity calibration matrix  $K = I$ , the calibrating conic is a unit circle centred at the origin (which is the principal point of the image). The conic of (8.18) is simply this unit circle transformed by an affine transformation according to the conic transformation rule of result 2.13(p37): ( $C \mapsto H^{-T}CH^{-1}$ ). Thus the calibrating conic of a camera with calibration matrix  $K$  is the affine transformation of a unit circle centred on the origin by the matrix  $K$ .

The calibration parameters are easily read from the calibrating conic. The principal point is the centre of the conic, and the scale factors and skew are easily identified, as in figure 8.27. In the case of zero skew, the calibrating conic has its principal axes aligned with the image coordinate axes. An example on a real image is shown in figure 8.29.

**Example 8.29.** Suppose  $K = \text{diag}(f, f, 1)$ , which is the calibration matrix for a camera of focal length  $f$  pixels, with no skew, square pixels, and image origin coincident with the principal point. Then from (8.18) the calibrating conic is  $C = \text{diag}(1, 1, -f^2)$ , which is a circle of radius  $f$  centred on the principal point.  $\triangle$

#### Orthogonality and the calibrating conic

A formula was given in (8.9–p210) for the angle between the rays corresponding to two image points. In particular the rays corresponding to two points  $\mathbf{x}$  and  $\mathbf{x}'$  are perpendicular when  $\mathbf{x}'^T \boldsymbol{\omega} \mathbf{x} = 0$ . As shown in figure 8.13(p212) this may be interpreted as the point  $\mathbf{x}'$  lying on the line  $\boldsymbol{\omega} \mathbf{x}$ , which is the polar of  $\mathbf{x}$  with respect to the IAC.

We wish to carry out a similar analysis in terms of the calibrating conic. Writing  $C = K^{-T}DK^{-1}$ , where  $D = \text{diag}(1, 1, -1)$ , we find

$$C = (K^{-T}K^{-1})(KDK^{-1}) = \boldsymbol{\omega}S$$

where  $S = KDK^{-1}$ . However, for any point  $\mathbf{x}$ , the product  $S\mathbf{x}$  represents the reflection of

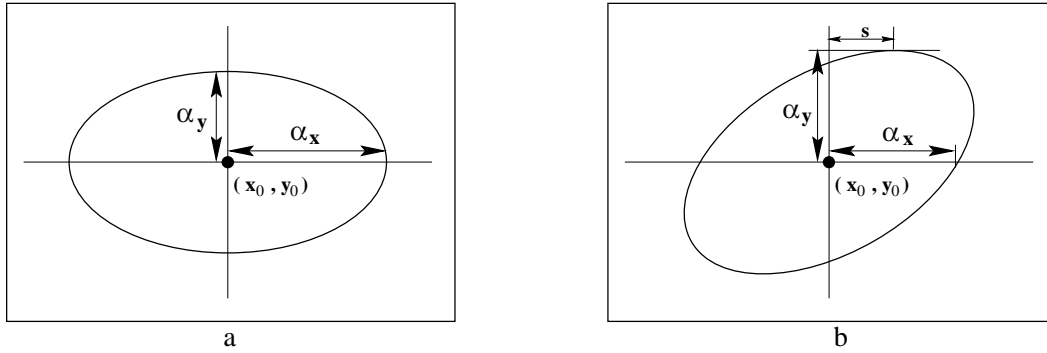


Fig. 8.27. **Reading the internal camera parameters  $K$  from the calibrating conic.** (a) Skew  $s$  is zero. (b) Skew  $s$  is non-zero. The skew parameter of  $K$  (see (6.10–p157)), is given by the  $x$ -coordinate of the highest point of the conic.

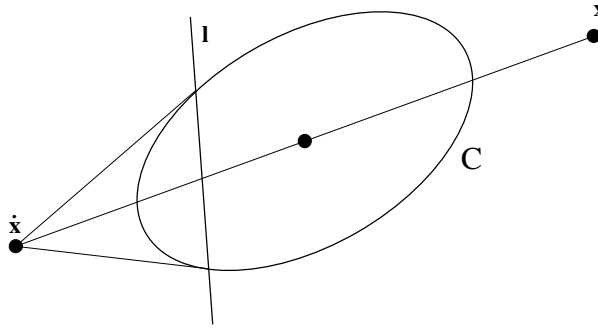


Fig. 8.28. To construct the line perpendicular to the ray through image point  $x$  proceed as follows: (i) Reflect  $x$  through the centre of  $C$  to get point  $\tilde{x}$  (i.e. at the same distance from the centre as  $x$ ). (ii) The desired line is the polar of  $\tilde{x}$ .

the point  $x$  through the centre of the conic  $C$ , that is, the principal point of the camera. Representing this reflected point by  $\tilde{x}$ , one finds that

$$\mathbf{x}'^T \omega \mathbf{x} = \mathbf{x}'^T C \tilde{\mathbf{x}} \quad (8.19)$$

This leads to the following geometric result:

**Result 8.30.** *The line in an image corresponding to the plane perpendicular to a ray through image point  $x$  is the polar  $C\tilde{x}$  of the reflected point  $\tilde{x}$  with respect to the calibrating conic.*

This construction is illustrated in figure 8.28.

**Example 8.31. The calibrating conic given three orthogonal vanishing points**

The calibrating conic can be drawn directly for the example of figure 8.22. Again assume there is no skew and square pixels, then the calibrating conic is a circle. Now given three mutually perpendicular vanishing points, one can find the calibrating conic by direct geometric construction as shown in figure 8.29.

- (i) First, construct the triangle with vertices the three vanishing points  $v_1$ ,  $v_2$  and  $v_3$ .
- (ii) The centre of  $C$  is the orthocentre of the triangle.

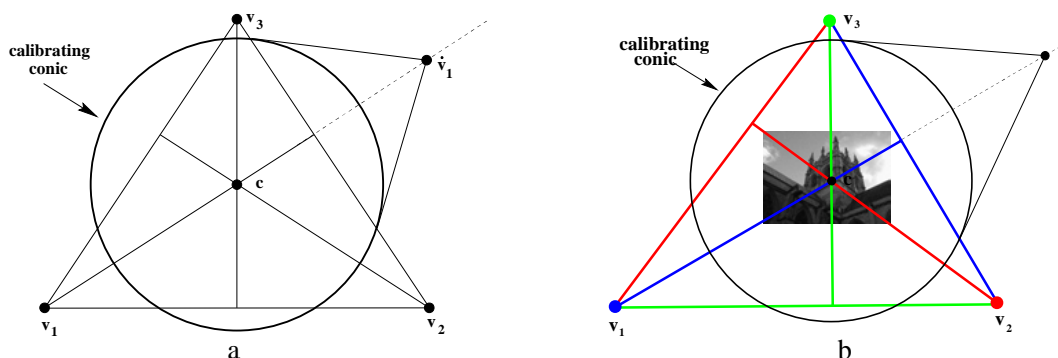


Fig. 8.29. The calibrating conic computed from three orthogonal vanishing points. (a) The geometric construction. (b) The calibrating conic for the image of figure 8.22.

- (iii) Reflect one of the vanishing points (say  $v_1$ ) in the centre to get  $\hat{v}_1$ .
- (iv) The radius of  $C$  is determined by the condition that the polar of  $\hat{v}_1$  is the line passing through  $v_2$  and  $v_3$ .

△

## 8.11 Closure

### 8.11.1 The literature

Faugeras and Mourrain [Faugeras-95a], and Faugeras and Papadopoulos [Faugeras-97] develop the projection of lines using Plücker coordinates. Koenderink [Koenderink-84, Koenderink-90], and Giblin and Weiss [Giblin-87] give many properties of the contour generator and apparent contour, and their relation to the differential geometry of surfaces.

[Kanatani-92] gives an alternative, calibrated, treatment of vanishing points and lines, and of the result that images acquired by cameras with the same centre are related by a planar homography. Mundy and Zisserman [Mundy-92] showed this result geometrically, and [Hartley-94a] gave a simple algebraic derivation based on camera projection matrices. [Faugeras-92b] introduced the projective (reduced) camera matrix. The link between the image of the absolute conic and camera calibration was first given in [Faugeras-92a].

The computation of panoramic mosaics is described in [Capel-98, Sawhney-98, Szeliski-97]. The ML method of computing vanishing points is given in Liebowitz & Zisserman [Liebowitz-98]. Applications of automatic vanishing line estimation from coplanar equally spaced lines are given in [Schaffalitzky-00b] and also [Se-00]. Affine 3D measurements from a single view is described in [Criminisi-00, Proesmans-98].

The result that  $K$  may be computed from multiple scene planes on which metric structure (such as a square) is known was given in [Liebowitz-98]. Algorithms for this computation are given in [Liebowitz-99a, Sturm-99c, Zhang-00]. The advantage in using  $\omega$ , rather than  $\omega^*$ , when imposing the skew zero constraint was first noted in [Armstrong-96b]. The method of internal calibration using three vanishing points for orthogonal directions was given by Caprile and Torre [Caprile-90], though there

is an earlier reference to this result in the photogrammetry literature [Gracie-68]. A simple formula for the focal length in this case is given in [Cipolla-99, Hartley-02b]. A discussion of the degeneracies that arise when combining multiple constraints is given in [Liebowitz-99b, Liebowitz-01]. Single view reconstruction is investigated in [Criminisi-99a, Criminisi-01, Horry-97, Liebowitz-99a, Sturm-99a].

### 8.11.2 Notes and exercises

- (i) **Homography from a world plane.** Suppose  $H$  is computed (e.g. from the correspondence between four or more known world points and their images) and  $K$  known, then the pose of the camera  $\{R, t\}$  may be computed from the camera matrix  $[r_1, r_2, r_1 \times r_2, t]$ , where

$$[r_1, r_2, t] = \pm K^{-1}H / \|K^{-1}H\|.$$

Note that there is a two-fold ambiguity. This result follows from (8.1–p196) which gives the homography between a world plane and calibrated camera  $P = K[R \mid t]$ .

Show that the homography  $x = H\tilde{X}$  between points on a world plane  $(n^T, d)^T$  and the image may be expressed as  $H = K(R - tn^T/d)$ . The points on the plane have coordinates  $\tilde{X} = (x, y, z)^T$ .

- (ii) **Line projection.**

- (a) Show that any line containing the camera centre lies in the null-space of the map (8.2–p198), i.e. it is projected to the line  $l = 0$ .
- (b) Show that the line  $\mathcal{L} = \mathcal{P}^T x$  in  $\mathbb{P}^3$  is the ray through the image point  $x$  and the camera centre. Hint: start from result 3.5(p72), and show that the camera centre  $C$  lies on  $\mathcal{L}$ .
- (c) What is the geometric interpretation of the columns of  $\mathcal{P}$ ?

- (iii) **Contour generator of a quadric.** The contour generator  $\Gamma$  of a quadric consists of the set of points  $X$  on  $Q$  for which the tangent planes contain the camera centre,  $C$ . The tangent plane at a point  $X$  on  $Q$  is given by  $\pi = QX$ , and the condition that  $C$  is on  $\pi$  is  $C^T \pi = C^T QX = 0$ . Thus points  $X$  on  $\Gamma$  satisfy  $C^T QX = 0$ , and thus lie on the *plane*  $\pi_\Gamma = QC$  since  $\pi_\Gamma^T X = C^T QX = 0$ . This shows that the contour generator of a quadric is a plane curve and furthermore, since  $\pi_\Gamma = QC$ , that the plane of  $\Gamma$  is the polar plane of the camera centre with respect to the quadric.

- (iv) **Apparent contour of an algebraic surface.** Show that the apparent contour of a homogeneous algebraic surface of degree  $n$  is a curve of degree  $n(n-1)$ . For example, if  $n = 2$  then the surface is a quadric and the apparent contour a conic. Hint: write the surface as  $F(x, y, z, w) = 0$ , then the tangent plane contains the camera centre  $C$  if

$$C_X \frac{\partial F}{\partial X} + C_Y \frac{\partial F}{\partial Y} + C_Z \frac{\partial F}{\partial Z} + C_W \frac{\partial F}{\partial W} = 0$$

which is a surface of a degree  $n-1$ .

- (v) **Rotation axis vanishing point for  $H = KRK^{-1}$ .** The homography of a conjugate rotation  $H = KRK^{-1}$  has an eigenvector  $Ka$ , where  $a$  is the direction of the rotation axis, since  $HKa = KRa = Ka$ . The last equality follows because  $Ra = 1a$ , i.e.  $a$  is the unit eigenvector of  $R$ . It follows that (a)  $Ka$  is a fixed point under the homography  $H$ ; and (b) from result 8.20(p215)  $v = Ka$  is the vanishing point of the rotation axis.
- (vi) **Synthetic rotations.** Suppose, as in example 8.12(p205), that a homography is estimated between two images related by a pure rotation about the camera centre. Then the estimated homography will be a conjugate rotation, so that  $H = KR(\theta)K^{-1}$  (though  $K$  and  $R$  are unknown). However,  $H^2$  applied to the first image generates the image that would have been obtained by rotating the camera about the same axis through twice the angle, since  $H^2 = KR^2K^{-1} = KR(2\theta)K^{-1}$ .

More generally we may write  $H^\lambda$  to represent a rotation through any fractional angle  $\lambda\theta$ . To make sense of  $H^\lambda$ , observe that the eigenvalue decomposition of  $H$  is  $H(\theta) = U \text{diag}(1, e^{i\theta}, e^{-i\theta}) U^{-1}$ , and both  $\theta$  and  $U$  may be computed from the estimated  $H$ . Then

$$H^\lambda = U \text{diag}(1, e^{i\lambda\theta}, e^{-i\lambda\theta}) U^{-1} = KR(\lambda\theta)K^{-1}.$$

which is the conjugate of a rotation through angle  $\lambda\theta$ . Writing  $\phi$  instead of  $\lambda\theta$ , we may use this homography to generate synthetic images rotated through any angle  $\phi$ . The images are interpolated between the original images (if  $0 < \phi < \theta$ ), or extrapolated (if  $\phi > \theta$ ).

- (vii) Show that the imaged circular points of a perspectively imaged plane may be computed if any of the following are on the plane: (i) a square grid; (ii) two rectangles arranged such that the sides of one rectangle are not parallel to the sides of the other; (iii) two circles of equal radius; (iv) two circles of unequal radius.
- (viii) Show that in the case of zero skew,  $\omega$  is the conic

$$\left(\frac{x - x_0}{\alpha_x}\right)^2 + \left(\frac{y - y_0}{\alpha_y}\right)^2 + 1 = 0$$

which may be interpreted as an ellipse aligned with the axes, centred on the principal point, and with axes of length  $i\alpha_x$  and  $i\alpha_y$  in the  $x$  and  $y$  directions respectively.

- (ix) If the camera calibration  $K$  and the vanishing line  $l$  of a scene plane are known then the scene plane can be metric rectified by a homography corresponding to a synthetic rotation  $H = KRK^{-1}$  that maps  $l$  to  $l_\infty$ , i.e. it is required that  $H^{-T}l = (0, 0, 1)^T$ . This condition arises because if the plane is rotated such that its vanishing line is  $l_\infty$  then it is fronto-parallel. Show that  $H^{-T}l = (0, 0, 1)^T$  is equivalent to  $Rn = (0, 0, 1)^T$ , where  $n = K^T l$  is the normal to the scene plane. This is the condition that the scene normal is rotated to lie along the camera  $Z$  axis. Note the rotation is not uniquely defined since a rotation about the plane's

normal does not affect its metric rectification. However, the last row of  $R$  equals  $\mathbf{n}$ , so that  $R = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{n}]^T$  where  $\mathbf{n}, \mathbf{r}_1$  and  $\mathbf{r}_2$  are a triad of orthonormal vectors.

- (x) Show that the angle between two planes with vanishing lines  $l_1$  and  $l_2$  is

$$\cos \theta = \frac{\mathbf{l}_1^T \boldsymbol{\omega}^* \mathbf{l}_2}{\sqrt{\mathbf{l}_1^T \boldsymbol{\omega}^* \mathbf{l}_1} \sqrt{\mathbf{l}_2^T \boldsymbol{\omega}^* \mathbf{l}_2}}.$$

- (xi) Derive (8.15–p218). Hint, the line  $l$  lies in the pencil defined by  $l_1$  and  $l_2$ , so it can be expressed as  $l = \alpha l_1 + \beta l_2$ . Then use the relations  $l_n = l_0 + n l$  for  $n = 1, 2$  to solve for  $\alpha$  and  $\beta$ .
- (xii) For the case of vanishing points arising from three orthogonal directions, and for an image with square pixels, show algebraically that the principal point is the orthocentre of the triangle with vertices the vanishing points. Hint: suppose the vanishing point at one vertex of the triangle is  $\mathbf{v}$  and the line of the opposite side (through the other two vanishing points) is  $l$ . Then from (8.17–p219)  $\mathbf{v} = \boldsymbol{\omega}^* l$  since  $\mathbf{v}$  and  $l$  arise from an orthogonal line and plane respectively. Show that the principal point lies on the line from  $\mathbf{v}$  to  $l$  which is perpendicular *in the image* to  $l$ . Since this result is true for any vertex the principal point is the orthocentre of the triangle.
- (xiii) Show that the vanishing points of an orthogonal triad of directions are the vertices of a self-polar triangle [Springer-64] with respect to  $\boldsymbol{\omega}$ .
- (xiv) If a camera has square pixels, then the apparent contour of a sphere centred on the principal axis is a circle. If the sphere is translated parallel to the image plane, then the apparent contour deforms from a circle to an ellipse with the principal point on its major axis.
- (a) How can this observation be used as a method of internal parameter calibration?
  - (b) Show by a geometric argument that the aspect ratio of the ellipse does not depend on the distance of the sphere from the camera.

If the sphere is now translated parallel to the principal axis the apparent contour can deform to a hyperbola, but only one branch of the hyperbola is imaged. Why is this?

- (xv) Show that for a general camera the apparent contour of a sphere is related to the IAC as:

$$\boldsymbol{\omega} = C + \mathbf{v}\mathbf{v}^T$$

where  $C$  is the conic outline of the imaged sphere, and  $\mathbf{v}$  is a 3-vector that depends on the position of the sphere. A proof is given in [Agrawal-03]. Note this relation places two constraints on  $\boldsymbol{\omega}$ , so that in principle  $\boldsymbol{\omega}$ , and hence the calibration  $K$ , may be computed from three images of a sphere. However, in practice this is not a well conditioned method for computing  $K$  because the deviation of the sphere's outline from a circle is small.