

University of Windsor

Scholarship at UWindor

Electronic Theses and Dissertations

Theses, Dissertations, and Major Papers

2012

The Extraction and Use of Image Planes for Three-dimensional Metric Reconstruction

Amirhasan Amintabar
University of Windsor

Follow this and additional works at: <https://scholar.uwindsor.ca/etd>

Recommended Citation

Amintabar, Amirhasan, "The Extraction and Use of Image Planes for Three-dimensional Metric Reconstruction" (2012). *Electronic Theses and Dissertations*. 410.
<https://scholar.uwindsor.ca/etd/410>

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000ext. 3208.

The Extraction and Use of Image Planes for Three-dimensional Metric Reconstruction

by

Amirhasan Amintabar

A Dissertation
Submitted to the Faculty of Graduate Studies
through Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy at the
University of Windsor

Windsor, Ontario, Canada

2011

© 2011 Amirhasan Amintabar

The extraction and use of image planes for three-dimensional metric reconstruction

by

Amirhasan Amintabar

APPROVED BY:

Dr. Robert Laganière, External examiner
University of Ottawa

Dr. Q. M. Jonathan Wu
Department of Electrical and Computer Engineering

Dr. Richard Frost
School of Computer Science

Dr. Imran Ahmad
School of Computer Science

Dr. Adlane Habed, Co-Advisor
University of Bourgogne, France

Dr. Boubakeur Boufama, Advisor
School of Computer Science

Dr. Paul Henshaw, Chair of Defence
Faculty of Graduate Studies

September 22, 2011

Declaration of Co-Authorship and Previous Publication

I. Co-Authorship Declaration

I hereby declare that this thesis incorporates material that is result of joint research with Dr. Boubakeur Boufama and Dr. Adlane Habed and under their supervision. I am aware of the University of Windsor Senate Policy on Authorship and I certify that I have properly acknowledged the contribution of other researchers to my thesis, and have obtained written permission from each of the co-authors to include the above materials in my thesis. I certify that, with the above qualification, this thesis, and the research to which it refers, is the product of my own work.

II. Declaration of Previous Publication

This thesis includes four original papers that have been previously published as follows:

Thesis Chapter	Publication title and full citation	Publication Status
3	A. Amintabar, B. Boufama, "Homography-based plane identification and matching from a pair of uncalibrated images", IEEE International Conference on Image Processing. San Diego, U.S.A, Oct. 12–15, pp. 297-300, 2008	published
3	A. Amintabar, B. Boufama, "The Distinction between Virtual and Physical Planes Using Homography", International Conference on Image Analysis and Recognition, ICIAR 2009, Halifax, Canada, pp. 727-736, 2009	published
4	A. Habed, A. Amintabar, B. Boufama, "Reconstruction-free parallel planes identification from un-calibrated images", 20th International Conference on Pattern Recognition ICPR 2010, Aug 23-26, pp. 1828-1831, 2010	published
5	A. Habed, A. Amintabar, B. Boufama, "Affine camera calibration from Homographies of parallel planes", IEEE International Conference on Image Processing ICIP 2010, Hong Kong, Sept 26-29, PP. 4249-4252, 2010	published

I certify that I have obtained a written permission from the copyright owners to include the above published materials in my thesis. I certify that the above material describes work completed during my registration as graduate student at the University of Windsor.

I declare that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owners to include such materials in my thesis.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

Abstract

The three-dimensional (3D) metric reconstruction of a scene from two-dimensional images is a fundamental problem in Computer Vision. The major bottleneck in the process of retrieving such structure lies in the task of recovering the camera parameters. These parameters can be calculated either through a pattern-based calibration procedure, which requires an accurate knowledge of the scene, or using a more flexible approach, known as camera autocalibration, which exploits point correspondences across images. While pattern-based calibration requires the presence of a calibration object, autocalibration constraints are often cast into nonlinear optimization problems which are often sensitive to both image noise and initialization. In addition, autocalibration fails for some particular motions of the camera.

To overcome these problems, we propose to combine scene and autocalibration constraints and address in this thesis (a) the problem of extracting geometric information of the scene from uncalibrated images, (b) the problem of obtaining a robust estimate of the affine calibration of the camera, and (c) the problem of upgrading and refining the affine calibration into a metric one. In particular, we propose a method for identifying the major planar structures in a scene from images and another method to recognize parallel pairs of planes whenever these are available. The identified parallel planes are then used to obtain a robust estimate of both the affine and metric 3D structure of the scene without resorting to the traditional error prone calculation of vanishing points. We also propose a refinement method which, unlike existing ones, is capable of simultaneously incorporating plane parallelism and perpendicularity constraints in the autocalibration

process. Our experiments demonstrate that the proposed methods are robust to image noise and provide satisfactory results.

Table of Contents

Declaration of Co-Authorship and Previous Publication	iii
Abstract.....	vi
List of Figures	xi
List of Tables.....	xii
List of Notations and Abbreviations	xiii
1 Introduction	1
1.1 Context and motivation	1
1.2 Scope of the thesis.....	3
1.3 Contributions	6
1.4 Outline of the thesis.....	7
2 Geometry in 3D Reconstruction.....	9
2.1 Projective Geometry	9
2.1.1 Cartesian vs. homogeneous coordinates.....	10
2.1.2 N-Dimensional Projective Space	11
2.1.3 The Dual of N-Dimensional Projective Space.....	11
2.1.4 2-D Projective Space	12
2.1.5 3-D Projective Space	13
2.2 Geometric transformations	13
2.2.1 Projective	14
2.2.2 Affine.....	15
2.2.3 Metric (Similarity)	16
2.2.4 Euclidean	16
2.3 Camera Model.....	18
2.3.1 The intrinsic parameters	21
2.3.2 Extrinsic Camera Parameters.....	22
2.3.3 Decomposition of projection matrix M	24
2.4 Epipolar geometry.....	25
2.5 Planar geometry.....	27
2.5.1 Planes at infinity.....	28
2.5.2 Inter-image homography	30
2.5.3 Homography and the plane in projective frame.....	32
2.6 3D Reconstruction Formulation.....	32

2.6.1	Metric stratum	34
2.6.2	Affine stratum	35
2.6.3	Projective stratum.....	35
3	Autocalibration constraints	37
3.1	Conics and quadrics	38
3.1.1	Conics	38
3.1.2	Quadric and its dual	40
3.2	Absolute Conic (AC) and Dual Absolute Quadric (DAQ).....	42
3.2.1	Dual Absolute Quadric	42
3.2.2	Properties of Dual Absolute Quadric	44
3.2.3	The images of AC and DAQ	45
3.3	Autocalibration	49
3.3.1	Autocalibration using Kruppa equations	50
3.3.2	Direct Autocalibration from DAQ.....	51
3.3.3	Stratified autocalibration	53
3.3.4	The modulus constraint	53
3.3.5	Chirality constraints	55
3.3.6	Constraints on intrinsic camera parameters.....	55
3.3.7	Constraints on camera motion	58
3.3.8	Scene constraints	58
4	The Distinction Between Virtual and Physical Planes	61
4.1	Problem Statement and Related Work.....	61
4.2	Identification of planes	65
4.2.1	Three-point method for homography calculation	66
4.2.2	Coplanar points identification (Steps 3-4)	69
4.2.3	Physical vs. virtual planes distinction (Steps 5-6)	69
4.2.4	Refining and merging (Steps 7-11).....	73
4.3	Experimental results and performance comparison	74
4.3.1	Performance Comparison	75
4.3.2	Noise Tolerance	77
4.4	Conclusion.....	78
5	Parallel Planes Identification	80
5.1	Problem Statement and Related Work.....	80
5.2	Background	82
5.3	Constraints due to parallel planes	84

5.4	Imposing Modulus Constraint.....	87
5.5	Proposed Parallel Planes Identification	88
5.6	Experiments	91
5.7	Conclusion.....	96
6	Combining Scene and autocalibration constraints	97
6.1	Planar scene constraints in affine calibration.....	98
6.2	Planar scene constraints in metric calibration.....	100
6.3	Proposed methods.....	102
6.3.1	A pair of parallel planes	104
6.3.2	Two pairs of parallel planes	106
6.3.3	Metric reconstruction from plane parallelism and orthogonality	108
6.4	Experiments	114
6.4.1	Using parallel planes constraints	114
6.4.2	Using parallel and perpendicular scene constraints	119
6.5	Conclusion.....	133
7	Conclusion and Remarks.....	135
	References.....	139
	VITA AUCTORIS.....	149

List of Figures

Figure 1-1. Proposed SfM pipeline.....	4
Figure 2-1. Pinhole camera model	19
Figure 2-2. Left: Skew factor, aspect ratio Right: principal point is not always located in the center of image	20
Figure 2-3. Extrinsic parameters	23
Figure 2-4. Epipolar geometry	25
Figure 2-5. Plane to image homography.....	28
Figure 2-6. Inter-image Homography	30
Figure 3-1. Calibration object	37
Figure 3-2. The absolute conic on plane at infinity.....	43
Figure 4-1. Virtual plane.....	63
Figure 4-2. Three space points define a plane and therefore a homography	67
Figure 4-3. Scenario 1, very likely physical	70
Figure 4-4. Scenario 2, likely physical.	70
Figure 4-5. Scenario 3, likely virtual.....	72
Figure 4-6. Scenario 4, very likely virtual	72
Figure 4-7. Comparing results.....	76
Figure 5-1. Two parallel planes meet in a line on plane at infinity.....	85
Figure 5-2. Planes 1, 3 and 4 are parallel.....	94
Figure 5-3. Planes 3,4 and 5 are parallel.....	94
Figure 5-4. Indoor scene, planes 1 and 3 are detected as parallel	95
Figure 5-5. Royal Victoria College (outdoor).....	95
Figure 5-6. A rock, 3 views out of 4.....	96
Figure 6-1. Two pairs of parallel planes meeting plane at infinity	107
Figure 6-2. Relative RMS 3D error using methods C1 (a and c) and D2 (b and d).	115
Figure 6-3. Two pairs of perfectly/perturbed parallel planes and 2 images (method D2).	117
Figure 6-4. Method C1 with 3 images, 3D metric reconstruction (right).	118
Figure 6-5. Method D2 with 2 images, 3D affine reconstruction (right).	119
Figure 6-6. Scene configuration for simulated data	120
Figure 6-7. Camera parameters vs image noise (pixel) for 3 to 9 views.....	122
Figure 6-8. 3D RMS error of metric reconstruction	123
Figure 6-9. Comparing initial guess with ground truth.....	124
Figure 6-10. Patterned sheets of paper pasted on the faces of packaging boxes.....	127
Figure 6-11. Planes labeled 1 and 3 are paralel. Plane labeld 2 is perpendicular to the two.....	130
Figure 6-12. Royal Victoria College in Montreal. Planes 2,3 are both orthogonal to plane 1	131
Figure 6-13. Table salt and business cards	132

List of Tables

Table 2-1. Hierarchy of geometries in 3D	17
Table 3-1. Camera calibration using calibration grid	37
Table 4-1. Intermediate results of applying the plane detection approaches	78
Table 5-1. Perfectly parallel planes.....	93
Table 5-2. Planes with 2 degrees angle.	93
Table 5-3. Planes with 4 degrees angle.	93
Table 6-1. The angle between the planes for boxes in Figure 6-10. a (in degree) using the proposed constrained method	128
Table 6-2. The angle between the planes for boxes in Figure 6-10. b (in degree) using the proposed constrained method	129
Table 6-3. The angle between the planes for boxes in Figure 6-10. b (in degree) using Pollefeys' method.....	129
Table 6-4. Experiments with real images.....	133

List of Notations and Abbreviations

\simeq	equality up to scale factor
$[v]_{\times}$	skew-symmetric matrix (3×3) of the vector
$\mathbb{P}^3, \mathbb{E}^3$	3D Projective space, 3D Euclidean space
$P_j \in \mathbb{P}^3$	a 3D scene point in projective frame
P^M	a reconstructed 3D point in metric frame
$M_i^E = K_i[R_i -t_i]$	M_i^E (3×4), Euclidean projection matrix of view i . K_i : the camera matrix. R_i (3×3): rotation matrix. t_i (1×3): translation vector.
M_i, M_i^M, M_i^A	projective, metric and affine projection matrices
$p_{i,j}, i = (1, 2, \dots, m)$	j^{th} point in the i^{th} view, in homogeneous
\tilde{p}	\tilde{p} : pixel point inhomogeneous, $\hat{p}=(x,y)$ but $p=(x,y,1)$
e_{21}	epipole in image 2, e_{21} is the projection of camera center of view 1 onto image 2
Π	world plane in projective frame $\Pi \simeq (v_{\Pi}^T \ 1)^T$
Π^E	world plane in Euclidean frame $\Pi^E = (n_{\pi}^T \ 1)^T$
$\infty \simeq (v_{\infty}^T, 1)^T$	plane at infinity in projective frame
$H_{\infty 12}$	homography of plane at infinity from view 1 to 2
C	conic, a 3×3 symmetric matrix
Q	quadric, a 4×4 symmetric matrix
Ω_{∞} or AC	Absolute Conic
Q_{∞}^* or DAQ	Dual Absolute Quadric
Q_{∞}^{*E}	DAQ in canonical form
ω or IAC	Image of the Absolute Conic
ω^* or DIAC	Dual Image of the Absolute Conic
RSS	Residual Sum of Squares (error)
RMS	Root Mean Square (error)
I	the identity matrix, (3×3)

1 Introduction

1.1 Context and motivation

Structure from motion (SfM) is the process of recovering the three-dimensional (3D) structure of a scene and camera pose from image sequences taken by either a single moving camera or by multiple cameras from multiple view points. This process, which supports 3D modeling, visualization and scene measurement [Sturm et al., 2011], is critical for the success of many applications including tele-presence [Snow, 1996, Tachakra, 2001], [Welch et al., 2011], robot navigation [Cardon et al., 2005] to archeology [Ducke et al., 2011], [Pollefeys et al., 2008] and the study of materials [Groeber et al., 2006]. In this respect, numerous approaches, such as Stereo, Shape from Shading [Bouguet and Perona, 1998], Shape from Silhouette [Kato et al., 1994] and the Grid Projection approach [Proesmans and Van Gool, 1997], [Lhuillier, 2008], with variable levels of accuracy, speed and practicality have been proposed in the literature. The applicability of each approach is dependent upon the allowed amount of user intervention and the nature of the data at hand. For instance, the relevance and significance of the 3D information that is to be extracted from the reconstructed scene depends upon whether or not the internal geometry of the camera and its relationship to the scene (which both define its calibration parameters) are known. For instance, the metric three-dimensional structure of a scene, which is a scaled version of the original structure, can be obtained from two or more images as soon as the internal geometry of the camera, i.e. its intrinsic parameters, is known. The scale ambiguity is resolved when

the camera is fully calibrated, i.e. its extrinsic parameters which define its relationship to the scene are also known exactly.

Camera calibration [Tsai, 1987] is traditionally carried out using a calibration pattern providing accurate 3D measurements and allowing the recovery of the scene-to-image mapping and hence the intrinsic and extrinsic parameters of the camera. As the presence of a pattern in the scene is often either impossible or impractical, a flexible alternative which relies solely on point correspondences across images, known as camera autocalibration, has received particular attention in the past two decades. Camera autocalibration has turned out to be among the most challenging problems in the entire SfM pipeline and, as such, numerous approaches have been proposed in the literature; for example: [Maybank and Faugeras, 1992], [Heyden and Astrom, 1996], [Triggs, 1997], [Pollefeys and Gool, 1999], [Seo and Heyden, 2004], [Habed and Boufama, 2006] and [Pollefeys et al., 2008].

Indeed, autocalibration constraints are often cast into nonlinear optimization problems which are known to be sensitive to both image noise and initialization. Methods solely based on these constraints generally require long sequences of images to help cope with various numerical issues. However, no matter how many images are used, autocalibration constraints always fail for some particular classes of camera motion [Sturm, 1997], [Kahl et al., 2000], [Sturm, 2000], and [Gurdjos et al., 2010].

The impracticality of pattern-based camera calibration on the one hand and the numerical issues of autocalibration methods on the other have led researchers to investigate an alternative approach in which scene constraints are used to support autocalibration [Liebowitz and Zisserman, 1999], [Pollefeys and Gool, 1999], [Sturm and

Quan, 1995], [Huynh and Heyden, 2005]. Such methods inherit the flexibility of autocalibration techniques as they rely on point correspondence across images and do not require the presence of a special pattern in the scene. Most of these methods are based on translating constraints (such as parallelism and orthogonality) on lines in the scene into constraints on their projections in an image. Indeed, as lines in the scene project onto lines in the image, such features can be easily identified and extracted: a quality which has made their use and investigation popular among researchers at the expense of other scene features such as planes. To our knowledge, except for the work [Huynh and Heyden, 2005] in which autocalibration and plane orthogonality are combined, no other method exploiting planes to support calibration has been reported in the literature. Unlike lines, scene planes cannot be identified from a single image since the entire scene projects onto one plane; the image plane, that is. However, planes are abundant in man-made environments and, if identified, their 3D coordinates can be calculated much more robustly (using tens or even hundreds of points) than those of lines. As a consequence, devising robust methods to identify planar structures in a scene from images as well as investigating their use to support camera calibration have emerged as both promising directions and challenging problems we have chosen to address in this thesis.

1.2 **Scope of the thesis**

Previous works exploiting scene constraints to support uncalibrated SfM have mainly used (parallel or orthogonal) lines. In this thesis, we investigate the use of planes for the metric 3D reconstruction of a scene from uncalibrated images: a problem that has been overlooked in the Computer Vision community except for the case of exploiting orthogonal planes [Huynh and Heyden, 2005]. In particular, we propose a new SfM

pipeline, depicted in Figure 1-1, which exploits constraints on planes while minimizing user intervention.

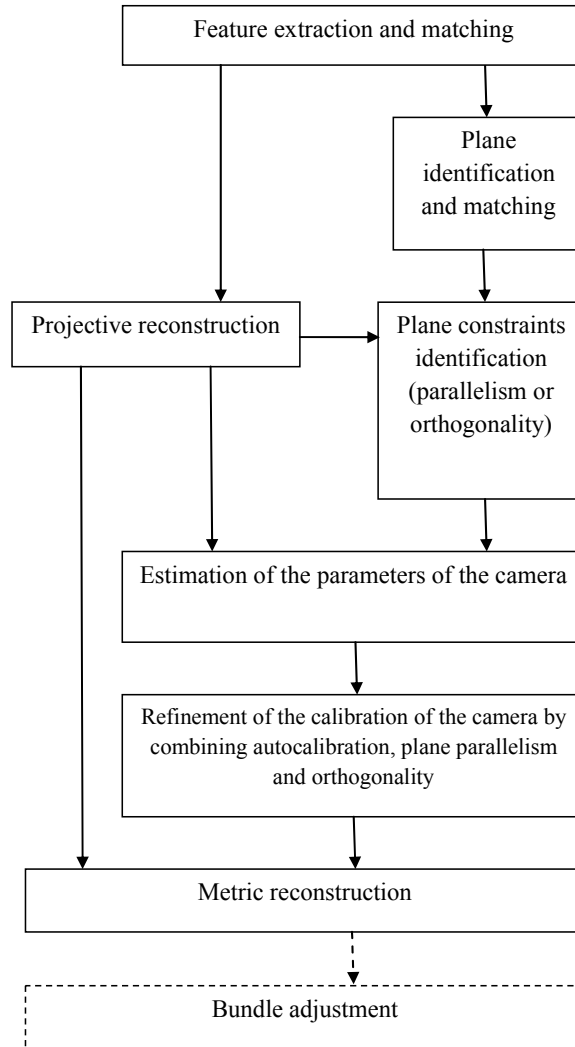


Figure 1-1. Proposed SfM pipeline

In the proposed approach, planes and constraints on these are to be identified from point correspondences across images. The identified pairs of planes exhibiting parallelism or orthogonality are used to locate the plane at infinity and hence to affinely calibrate a moving camera with constant parameters. An estimate of the camera's

intrinsic parameters can be linearly obtained at this point. Subsequently, all scene constraints (parallel planes and perpendicular ones) and camera autocalibration constraints are to be combined to obtain the optimal set of parameters of the camera. At this point the projective structure of the scene is upgraded into a metric one and a bundle adjustment procedure can optionally be carried out.

In particular, we address in this thesis the following problems which make the applicability of such an SfM approach challenging.

Identification of planar structures and their relationships: most methods incorporating scene constraints (whether using lines or planes) do so at the cost of involving the intervention of a user to identify the relationships between the features under consideration. In the case of planes, user intervention is additionally required to identify the features themselves, i.e. which image points are the projections of points that lie on the same plane in the scene. In this thesis, we address the problem of identifying planar structures in a scene using only point correspondences across images and without resorting to the intervention of a user. We also address the problem of identifying, with sufficient confidence and without user intervention, the relationship (such as parallelism or orthogonality) that may exist between planes.

Exploiting plane parallelism: line parallelism has widely been exploited in the literature. Indeed, the projections of two lines that are parallel in the scene intersect in the image in a special point known as a vanishing point. A vanishing point is the projection of a point lying on a plane located at infinity in 3-space. Identifying the plane at infinity allows, in general, the recovery of the calibration of the camera. Parallel planes also intersect at infinity; in a line, that is. However, in general, this line cannot be physically

located in the image making the problem of exploiting parallel planes seemingly unnatural and certainly harder than using lines. In this thesis, we investigate the use of plane parallelism to support autocalibration as an alternative to the method [Huynh and Heyden, 2005] which exploits orthogonality between such entities.

Incorporating both parallelism and orthogonality: existing methods for combining autocalibration and scene constraints have been designed to exploit either orthogonality or parallelism but not both at the same time. The reason is that parallelism is generally used to locate the plane at infinity and, unlike when using orthogonality, no known algebraic constraints directly relating parallelism and the parameters of the camera have been derived: a problem that we also address in this thesis.

1.3 Contributions

The main contributions presented in this thesis are as follows:

1. **Identification of planes from uncalibrated images:** we have proposed a new method [Amintabar and Boufama, 2008] for extracting the major planar structures in a scene from uncalibrated pairs of images. This method contrasts with existing methods by not making any prior assumption on the co-planarity of points. It also performs plane identification and matching defined by sets of three points only instead of four. Once all possible planes have been identified, a merging stage is carried out to make sure that the same planes are associated with a single homography.
2. **Distinguishing virtual from physical planes:** identifying planes from a pair of uncalibrated stereo images is a challenging problem as it can lead to extracting virtual planes instead of physical ones. We have proposed in [Amintabar &

Boufama 2009] a new homography-based approach to extract physical planes and to distinguish them from virtual ones in general scenarios.

3. **Identification of parallel planes:** we have also proposed in [Habed et al., 2010b] a new method for identifying parallel planes in a scene from three or more uncalibrated images. By using the fact that parallel planes intersect at infinity, we were able to devise a linear relationship between the inter-image homographies of the parallel planes and the plane at infinity. This relationship is combined with the so-called modulus constraint for identifying pairs of parallel planes solely from point correspondences.
4. **Affine camera calibration:** a new method to retrieve the affine structure of a scene from two or more images of parallel planes is presented in [Habed et al., 2010a]. The proposed approach is solely based on plane homographies calculated from point correspondences and does not require the recovery of the 3D structure of the scene. Neither vanishing points nor lines need to be extracted from images.
5. **Incorporating plane parallelism and orthogonality into autocalibration:** we also propose a new method, which we describe in Chapter 6, capable of simultaneously combining autocalibration constraints and both plane perpendicularity (orthogonality) and parallelism. The results of our experiments show that our proposed calibration method is efficient and robust to noise. Although nonlinear, our method has a very satisfactory convergence rate even when the optimization procedure is randomly initialized.

1.4 Outline of the thesis

The remainder of this dissertation is organized as follows.

- Chapter 2 provides the necessary background used to formulate the main contributions of this dissertation. It reviews concepts such as projective geometry, camera models, the relationships between multiple views, and the main reconstruction strata.
- Chapter 3 describes the fundamentals of camera calibration including conics and quadrics. It also lists and discusses the constraints used in autocalibration approaches.
- Chapter 4 deals with the identification of planes from a pair of uncalibrated images and the distinction between virtual and physical ones.
- Chapter 5 elaborates on the proposed reconstruction-free identification of parallel planes.
- Chapter 6 is dedicated to the problem of combining scene and autocalibration constraints. In this chapter, we present an affine camera autocalibration which requires only one pair of parallel planes in the scene but constant camera parameters. We also present a method which may employ different cameras altogether. Furthermore, a nonlinear metric autocalibration approach, which simultaneously combines autocalibration constraints with parallel and orthogonal planes, is presented.
- Finally, Chapter 7 concludes this dissertation with further discussions of future work.

2 Geometry in 3D Reconstruction

This chapter is an introduction to the main concepts covered in this dissertation. It introduces the projective geometry of two and three dimensional spaces. It also introduces the underlying geometric relationships between images: the homography induced by the projection of coplanar points and more general epipolar geometry relating corresponding points across pairs of images. Next, the planar geometry is described which is the basis of our contributions. The camera model and the 3D structure calculation from 2D images are also presented. This chapter is concluded by elaborating on the main 3D reconstruction strata encountered in Computer Vision.

2.1 Projective Geometry

Projective geometry is an essential tool in the modeling and analysis of 3D computer vision problems. As the main inputs to 3D computer vision algorithms are 2D images, projective geometry is widely used to model the projection of 3D scenes onto 2D image planes. Besides the linearization of the projection equations, an important advantage of embedding a scene into a projective space is that points which are infinity far from the camera may be treated as any other point that lies at a finite distance from it. In order to achieve this, geometric entities such as point, lines and planes are described by homogeneous coordinates. However, these advantages come at the expense of additional ambiguities in comparison to ordinary space. Indeed, unlike in ordinary space, in the projective space parallel lines do intersect. As a consequence, an object that is square-shaped in ordinary space and which undergoes a projective transformation may change into an arbitrary quadrangle. In fact, angles, distances and ratios are not invariant under

projective transformation. The cross-ratio (the ratio of ratios) is however invariant under such transformation and, despite added ambiguities, this property alone has turned projective geometry into a powerful tool in the field of Computer Vision.

2.1.1 Cartesian vs. homogeneous coordinates

In an n -dimensional space, a Cartesian reference frame consists of a point O , called the origin, and n linearly independent vectors $\vec{v}_1, \dots, \vec{v}_n$ called the base. In this space a point P is represented by $\vec{OP} = x_1 \vec{v}_1 + x_2 \vec{v}_2 + \dots + x_n \vec{v}_n$ where x_1, x_2, \dots, x_n are scalars known as the *Cartesian coordinates* of point P . In vector form, P is given by its Cartesian coordinate vector of *dimension* n as follows

$$\tilde{P} = (x_1, x_2, \dots, x_n)^T.$$

One problem that arises when using the Cartesian representation is that only points at a finite distance from the origin can be expressed in this coordinate system. Therefore, homogenous coordinates are used so that points at infinity can also be represented. This is done by adding one more component to the representation of points (as well as other geometric entities). In an n dimensional space, the homogeneous coordinates of point P is represented by $n+1$ components as follows

$$P \simeq (t_1, \dots, t_n, t_{n+1})^T \text{ or } P \simeq (\tilde{P}^T, 1)^T.$$

where \simeq means equality up to one scalar parameter. Thus, the Cartesian coordinates of a finite P , i.e. $H_\phi(i)$ can be recovered from their homogeneous counterparts as $\tilde{P} = (x_1, \dots, x_n)^T = (t_1/t_{n+1}, \dots, t_n/t_{n+1})^T$. In a homogeneous coordinate system spanning the n -dimensional space, a *point at infinity* P_∞ may be described by

$$P_\infty = (t_1, \dots, t_n, 0)^T. \quad (2.1)$$

Points at infinity are also referred as *ideal points* in this dissertation.

2.1.2 N-Dimensional Projective Space

In this space, geometric entities such as points, lines, or planes, are all represented in homogenous coordinates. As a result, relations between these entities can be described in simple and unified manner as opposed to the Cartesian representation. The n -dimensional projective space, which we denote \mathbb{P}^n , is the set of points represented in homogeneous coordinates by

$$\mathbb{P}^n = \{(x_1, \dots, x_{n+1})^T\} - \underbrace{\{(0, 0, \dots, 0)^T\}}_{n+1 \text{ zeros}},$$

where the sign " – " indicates that $(0, \dots, 0)^T$ does not define any point in \mathbb{P}^n , however \mathbb{P}^n includes finite points (including origin $(0, \dots, \lambda)^T$ $\lambda \neq 0$ of the reference frame as well as points at infinity (e.g. $(x_1, x_2, \dots, x_n, 0)^T$).

2.1.3 The Dual of N-Dimensional Projective Space

n linearly independent points in \mathbb{P}^n form a hyperplane. In \mathbb{P}^2 , this hyperplane is a line as it is formed by two linearly independent points. In \mathbb{P}^3 , the hyperplane is a plane as it is formed by three linearly independent points in \mathbb{P}^3 . As a point in \mathbb{P}^n is represented by an $(n + 1)$ -dimensional vector, a hyperplane defined by n linearly independent points is also represented by an $(n + 1)$ -dimensional vector. This makes points and hyperplanes indistinguishable in this space. In other words, the vector (a, b, c) can be interpreted both as a point or a line in \mathbb{P}^2 . So are points and planes in \mathbb{P}^3 .

Dual entities in n -space form a space of their own, that is the dual space generally denoted by \mathbb{P}^3 .

Thus, the dual space \mathbb{P}^{*n} is the set of all hyperplanes defined by n points taken in \mathbb{P}^n . Similarly, \mathbb{P}^n can be viewed as the set of points formed by the intersection of distinct, linearly independent sets of n hyperplanes in \mathbb{P}^{*n} .

2.1.4 2-D Projective Space

The 2-dimensional projective space, \mathbb{P}^2 , is represented by 3 homogeneous coordinates as $\mathbb{P}^2 = \{(x_1, x_2, x_{2+1})^T\} - \{(0, 0, 0)^T\}$ where points with $x_3 = 0$ (e.g. $(x_1, x_2, 0)^T$) are at infinity. These points, altogether, form a line that is called the *line at infinity*.

Duality between points and lines in 2D

Two distinct points, p_1 and p_2 in \mathbb{P}^2 define a line l in \mathbb{P}^{*2} . The homogeneous coordinate vector of line l is obtained through the cross-product of the points defining the line

$$l \simeq p_1 \times p_2, \tag{2.2}$$

where \times denotes the cross-product throughout this dissertation.

Similarly, two distinct lines, l_1 and l_2 , intersect in a unique point p with homogeneous coordinate

$$p \simeq l_1 \times l_2 = [l_1]_{\times} l_2,$$

which is the dual of equation (2.2).

Incidence in 2D

Incidence refers to the relation between a point and its dual in space, which is line in two dimensional space. Point p is on a line l iff

$$p^T l = 0, \quad (2.3)$$

where $p^T l$ denotes the inner product of p and l .

2.1.5 3-D Projective Space

The 3-dimensional projective space \mathbb{P}^3 is similarly represented by 4 homogeneous coordinates as $\mathbb{P}^3 = \{(x_1, x_2, x_3, x_4)^T\} - \{(0, 0, 0, 0)^T\}$ where points with $x_4 = 0$ (e.g. $(x_1, x_2, x_3, 0)^T$) are at infinity. These points, altogether, form a plane that is called the *plane at infinity*. The plane at infinity is denoted by ∞ throughout this dissertation.

Duality between point and planes in 3D

Three distinct and linearly independent points $P_1, P_2, P_3 \in \mathbb{P}^3$ define a unique plane Π in \mathbb{P}^3 , which is represented exactly the same way as its dual (a point):

$$\Pi \simeq (\delta_1, \delta_2, \delta_3, \delta_4)^T,$$

where $\delta_1 \dots \delta_4$ are scalars.

Incidence in 3D

Point $P \in \mathbb{P}^3$ lies on the plane Π iff

$$P^T \Pi = 0. \quad (2.4)$$

The equation (2.4) can be expressed as $\Pi^T P = 0$ as well.

2.2 Geometric transformations

Geometry may be defined as the study of properties which remain invariant under groups of transformations [Klein, 2004]. We therefore describe projective geometry through

general transformations as well as the main specializations of these including the affine, metric and Euclidean transformations. A transformation is algebraically defined by matrix. Within \mathbb{Q}_{∞}^* for instance, projective linear transformations are represented by $(n+1) \times (n+1)$ invertible matrices and form a group.

2.2.1 Projective

Projective transformations are *linear transformations* from one projective space into another or within the same space. In general, a projective transformation alters angles, distances and ratios of geometric shapes. The only properties which are guaranteed to be invariant are strangeness, incidence and cross ratios. A projective transformation T from \mathbb{P}^n to \mathbb{P}^m maps each point $P \in \mathbb{P}^n$ to a new location in \mathbb{P}^m as $P \mapsto TP$ where T is an $(m+1) \times (n+1)$ matrix.

Homography

A homography H is a projective transformation within the same space:

$$\begin{aligned} H : \mathbb{P}^n &\rightarrow \mathbb{P}^n \\ P &\mapsto HP, \end{aligned}$$

where H is a nonsingular $(n+1) \times (n+1)$ matrix. In \mathbb{P}^3 , such transformation is described by a regular 4×4 matrix having 15 degrees of freedom. Homography is used very often in this dissertation.

Homographies in dual space

When space points undergo a projective transformation H from \mathbb{P}^n to itself, the dual primitives L in \mathbb{P}^{*n} (recall lines in \mathbb{P}^{*2} or planes \mathbb{P}^{*3}), undergo a dual transformation H^* such that:

$$H^* : \mathbb{P}^{*n} \rightarrow \mathbb{P}^{*n}$$

$$L \mapsto H^{-T} L.$$

The dual transformation H^* is represented by the inverse of the transpose of the matrix representing H . In \mathbb{P}^3 , a homography transformation H , is represented by a 4×4 nonsingular matrix of the following form.

$$H \simeq \begin{bmatrix} A & t \\ v^T & 1 \end{bmatrix}, \quad (2.5)$$

where A is any 3×3 nonsingular matrix and t and v are 3-vectors.

2.2.2 Affine

An affine transformation is a special projective transformation that leaves the hyperplane at infinity globally invariant. It therefore preserves parallelism and consequently the same direction ratios too. In \mathbb{P}^3 , an affine transformation is represented by a 4×4 matrix (for convenience its inverse is also provided):

$$T_A \simeq \begin{bmatrix} A & t \\ 0^T & 1 \end{bmatrix}, \quad T_A^{-1} \simeq \begin{bmatrix} A^{-1} & -A^{-1}t \\ 0^T & 1 \end{bmatrix}, \quad (2.6)$$

where A is any 3×3 nonsingular matrix and t is a vector. T_A has 12 degrees of freedom. The last row is the coordinate vector of the plane at infinity in Euclidean space. The only difference with a projective transformation is that the vector v in (2.5) has been replaced with a null vector 0 in (2.6).

2.2.3 Metric (Similarity)

It is a specialization of affine transformations that is composed of a translation, rotation and scaling. Such transformations preserve angles and ratios in any directions. In \mathbb{P}^3 , a similarity transformation is represented by a 4×4 matrix:

$$T_M \simeq \begin{bmatrix} \lambda R & t \\ 0^T & 1 \end{bmatrix}, \quad (2.7)$$

where R is a 3×3 rotation matrix in 3D, $t = (t_1, t_2, t_3)^T$ is a translation vector, and λ is a non-zero isotropic scale factor. T_M has 7 degrees of freedom. One rotation about each of the three axes, one translation in each of the three directions, and one global scale factor.

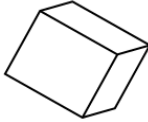

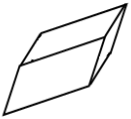
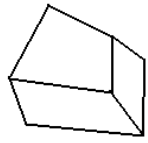
2.2.4 Euclidean

A Euclidean or Isometric transformation is a specialization of the similarity transformation. It is composed of a translation and a rotation. Euclidean transformations model the rigid transformation of an object. Such transformations preserve angles and distances in any directions. In \mathbb{P}^3 , a Euclidean transformation is represented by a 4×4 matrix:

$$T_E \simeq \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix}, \quad (2.8)$$

where R is a 3×3 rotation matrix in 3D, therefore $R^{-1} = R^T$ and $\det(R) = 1$. $t = (t_1, t_2, t_3)^T$ is a translation vector. T_E has 6 degrees of freedom. One rotation about each of the three axes, one translation in each of the three directions. The distortion effects of various transformations in 3D space are summarized in Table 2-1.

Table 2-1. Hierarchy of geometries in 3D

Group	Matrix	Distortion	Invariant properties
Euclidean 6dof	$T_E \simeq \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix}$		angles and distances in any directions
Similarity 7dof	$T_M \simeq \begin{bmatrix} \lambda R & t \\ 0^T & 1 \end{bmatrix}$		angles and ratios in any directions
Affine 12dof	$T_A \simeq \begin{bmatrix} A & t \\ 0^T & 1 \end{bmatrix}$		Parallelism of planes, volume ratios, centroids. The plane at infinity ∞
Projective 15 dof	$T \simeq \begin{bmatrix} A & t \\ v^T & 1 \end{bmatrix}$		Intersection and straightness

If we compare the projective transformation matrices with the affine one in Table 2-1, we notice that the difference is in the vector v which is null in the definition of affine transformation. Therefore, an affine transformation retains the ideal points $(x, y, z, 0)^T$ at infinity:

$$\begin{bmatrix} A & t \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \\ 0 \end{bmatrix}. \quad (2.9)$$

It also explains why an affine transformation leaves the (hyper)plane at infinity globally invariant. In the case of projectivity, the vector $v = (v_1, v_2, v_3)^T$ in (2.5) is responsible for the non-linear effects of the projectivity. If we map an ideal point $(x, y, z, 0)^T$ under a

projective transformation we see that the ideal point maps to a finite point, referred as *vanishing point*:

$$\begin{bmatrix} A & t \\ v^T & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \\ v_1x + v_2y + v_3z \end{bmatrix}. \quad (2.10)$$

This can be noticed by comparing the last row of calculated results in (2.9) with (2.10).

2.3 Camera Model

A camera maps the three-dimensional world onto a two-dimensional picture. The mapping under which a 3D scene maps onto a 2D space process is called perspective projection. This projection is described by a (3×4) matrix which formulates the relation between a scene embedded in the 3D projective space \mathbb{P}^3 and its image embedded in a 2D projective space \mathbb{P}^2 .

$$p \simeq MP, \quad M = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix}, \quad (2.11)$$

where 3D points P and their 2D projection p are in their homogenous representation and $m_{i,j}$ are real numbers, i.e. $m_{i,j} \in \mathbb{R}$.

The pinhole camera model describes the perspective projection of the scene in terms of various parameters describing the internal geometry of the camera and its relationship to the scene. This model is widely used as a good approximation to model digital cameras with CCD-like sensors. It models the central projection of 3D space

points onto a plane called *image plane* (or *focal plane* or *retina*). For simplicity we usually model a pinhole camera by placing the image plane between the center of projection (or focal point), towards which all light rays converge, and the object (frontal pinhole model), so that the formed image is not inverted (Figure 2-1).

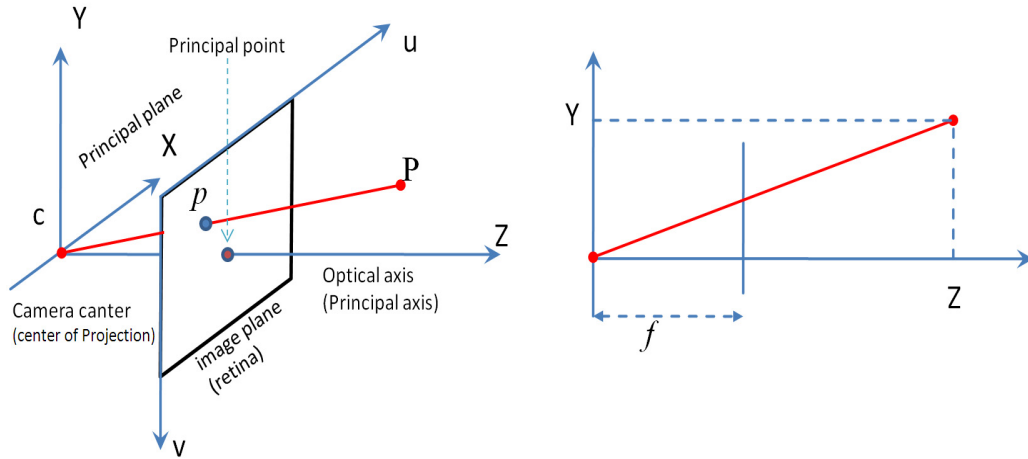


Figure 2-1. Pinhole camera model

In Figure 2-1, the camera coordinate system, $\langle XYZ, C \rangle$, is originated at the center of projection C . A scene point P is a 3D point whose definition in camera coordinate system is denoted by $P = (X_c, Y_c, Z_c)$. The line perpendicular to the image plane is called *principal axis*. The intersection of the principal axis with the image plane is called *principal point* o , which is the origin of the 2D coordinate system $\langle xy, o \rangle$. x and y axes are parallel to X and Y of camera coordinate system respectively. So, referring to the right hand side of Figure 2-1, it can be seen that a 3D point P is projected onto the point $p = (x, y)$ in the image plane by:

$$x = f \frac{X_c}{Z_c}, \quad y = f \frac{Y_c}{Z_c}, \quad (2.12)$$

where f is the distance of camera center C to the image plane, and is called focal length.

In digital cameras (e.g. CCD/CMOS cameras), the image plane is a matrix of pixels where the horizontal and vertical inter-distances (dimension of pixels) may not necessarily be the same. To consider this in our camera model, we define a metric called *aspect ratio* which is the ratio of inter-pixel distances in horizontal direction divided by inter-pixel distance in vertical direction. Also in cameras the vertical columns of pixels may not be necessarily orthogonal to the horizontal rows [Kanatani, 2008]. We therefore say the *skew factor* may not be zero (Figure 2-2). Nevertheless, in today's manufactured digital cameras the skew factor is close to zero. Another point to consider in the modeling of cameras is that the principal point is not always located in the center of the image as the latter may move during the focusing process due to the motion of the lenses (Figure 2-2).

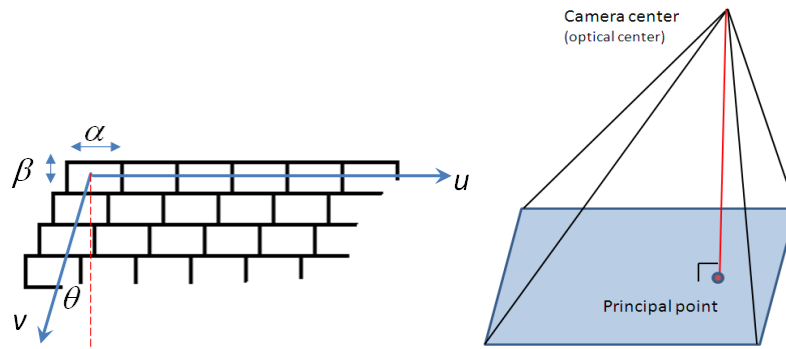


Figure 2-2. Left: Skew factor, aspect ratio Right: principal point is not always located in the center of image

In digital imaging a common convention is to describe the coordinates of image points with respect to the top-left corner of the image (Figure 2-1). Therefore the pixel point $\tilde{p}(u, v)$ is defined as

$$u = \frac{x}{\alpha} + \frac{y}{\beta} \tan\theta + u_0, \quad v = \frac{y}{\beta} + v_0, \quad (2.13)$$

where (u_0, v_0) is the location of principal point with respect to the coordinates $\langle uv, o \rangle$. α is the horizontal inter-pixel distance and accordingly β is the vertical inter-pixel distance and $\tilde{p}(u, v)$ is the Cartesian representation of point $p \simeq (u, v, 1)$. θ is the angle between u and v coordinates.

Central projection using homogeneous coordinates

If the scene and image points are represented by their homogeneous coordinates, then central projection is simply expressed as a linear mapping between them. Combining equations (2.12) and (2.13), one obtains

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \simeq \begin{bmatrix} f\frac{\beta}{\alpha} & f\frac{\beta}{\alpha}\tan\theta & u_0 & 0 \\ & f & v_0 & 0 \\ & & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}. \quad (2.14)$$

2.3.1 The intrinsic parameters

The equation (2.14) may be written as

$$p \simeq K[I \mid O]P_c, \quad (2.15)$$

where P_c and p are the homogeneous vectors of scene point P defined in camera coordinate system and image point in the image coordinate system respectively. The matrix K is derived from (2.13) and is represented by a 3×3 matrix as

$$K = \begin{bmatrix} f_{\alpha}^{\beta} & f_{\alpha}^{\beta} \tan \theta & u_0 \\ & f & v_0 \\ & & 1 \end{bmatrix}. \quad (2.16)$$

It is a common practice in textbooks to represent the matrix K in a simpler form

$$K = \begin{bmatrix} f_u & s & u_0 \\ & f_v & v_0 \\ & & 1 \end{bmatrix}, \quad (2.17)$$

where f_u and f_v represent the focal length along the axis of the image. The aspect ratio is defined by f_u/f_v , s is for skew factor and (u_0, v_0) are the pixel coordinates of the principal point. These parameters are referred to as the *intrinsic parameters* of the camera.

Distortions

The camera model described above is linear. That means it is assumed that straight lines are projected as straight lines which is not quite true for real lenses, especially cheap optics and for short focal lengths. The well-known nonlinear distortion is radial distortion discussed in [Vergauwen, 2006]. For more information on nonlinear camera models the reader can refer to [Sturm et al., 2011].

2.3.2 Extrinsic Camera Parameters

In our previous discussion, the 3D scene points $\tilde{P}_c(X_c, Y_c, Z_c)$ was assumed to be described in the camera's coordinate system. However, in general the coordinates of P are more conveniently available in another reference frame called the world (or scene)

coordinate frame ($\tilde{P}(X, Y, Z)$ in Figure 2-3). In such case, for the validity of calculations, all the points including the camera center have to be also defined in that coordinate system. Let's assume that the optical center of the camera is defined in the scene coordinate system by the Cartesian vector \tilde{C} . Then, the relation between the representation of P in the scene coordinate system and that of P in the camera coordinate system is given by.

$$\begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} = R \left(\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} - \tilde{C} \right) \quad (2.18)$$

where vector \tilde{C} is defining the center of camera frame, the origin C , with respect to world frame. R is a rotation matrix which represents the orientation of the camera coordinate system with respect to the world coordinate system.

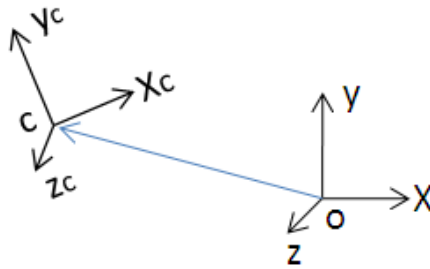


Figure 2-3. Extrinsic parameters

Equation (2.18) implies that if the description of a 3D point is available in a world coordinate system XYZ and the R and \tilde{C} are also available (Figure 2-3), then the scene points can be defined in camera coordinate system as in (2.18). The parameters $\{R, \tilde{C}\}$ are called *extrinsic parameters* of the camera and are defined with respect to camera

coordinate system. Replacing $[X_c, Y_c, Z_c]^T$ in (2.14) with its equivalent in (2.18) and combining equations (2.14), (2.11) and (2.18) we obtain

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = KR[I \mid -\tilde{C}] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix},$$

where

$$M = KR[I \mid -\tilde{C}] \quad (2.19)$$

is the factorization of the projection matrix which appears in (2.11) in term of the intrinsic and extrinsic parameters of the camera.

2.3.3 Decomposition of projection matrix M

If only the 3×4 projection matrix M is known and not the individual parameters, it is still possible to derive the intrinsic and extrinsic parameters from it. Thus we are interested in decomposing M into intrinsic camera matrix K , and extrinsic matrices $\{R, t\}$. The process is as follows [Vergauwen, 2006]. We consider the upper left 3×3 sub-matrix of M . This matrix is on the form KR . The inverse of this matrix is $R^T K^{-1}$ (as $R^T = R^{-1}$). The QR-decomposition of a 3×3 matrix of rank 3 decomposes the matrix into two matrices: an orthonormal matrix on the left and an upper triangular matrix on the right. R is a rotation matrix and thus orthonormal and K is modeled as an upper triangular matrix, so K^{-1} is an upper triangular too. Now that K and R are known, the translation vector t can be easily computed by pre-multiplying the fourth column of M with $-RK^{-1}$.

2.4 Epipolar geometry

The relationship between the projections of points in two images of the same scene is described by the epipolar geometry. We assume here that two cameras are observing a scene and that the world coordinate system is attached to the first camera, Figure 2-4. Referring to equation (2.19), the Euclidean projection matrices of the two views are in the form:

$$M_1 = [K_1 | 0] \text{ and } M_2 = K_2 R_2 [I | -t_2], \quad (2.20)$$

where t_2 describes the location of the center of second camera, O_2 , in the coordinate frame of first camera and R_2 describes the rotation of camera 2 with respect to camera 1.

Epipole e_{ij}

An *epipole* e_{ij} is a special point on the i^{th} image which is the projection of the optical center of camera j on the image plane of the camera i . Given two images of a scene, the epipole e_{12} on the first image is the projection of the optical center $O_2 \simeq (t_2^T, 1)^T$ of the second camera on the image plane \mathcal{I}_1 of the first camera $e_{12} \simeq M_1 O_2$

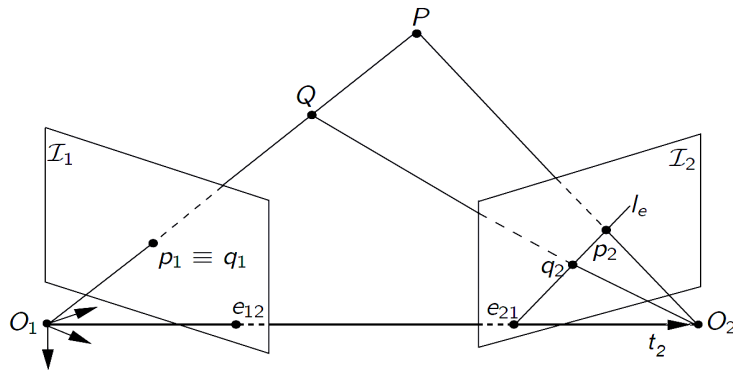


Figure 2-4. Epipolar geometry

Similarly epipole e_{21} on image 2 is obtained by projecting $O_1 \simeq (0^T, 1)^T$ on image 2 through $e_{21} = M_2 O_1$. Therefore

$$e_{12} \simeq K_1 t_2 \text{ and } e_{21} \simeq -K_2 R_2 t_2. \quad (2.21)$$

Epipolar line

Let p_1 and p_2 be the projections of a scene point P in two images. Point P lies anywhere on the optical ray $\overrightarrow{O_1 p_1}$. In turn, point p_2 must lie anywhere on the line $l_{e_{21}}$ which is the projection of $\overrightarrow{O_1 P}$ on the second image. The line l_e contains the epipole e_{21} (projection of O_1 on image 2) and point p_2 . Therefore, we have

$$l_{e_{21}} = e_{21} \times p_2 \quad (2.22)$$

The line $l_{e_{21}}$ is referred to as the epipolar line associated to point p_1 . All epipolar lines intersect in the epipole.

Fundamental Matrix and Epipolar constraint

Two points p_1 and p_2 , corresponding to the projections of the same scene point P in two images are related by

$$p_2^T F_{12} p_1 = 0 \text{ or } p_1^T F_{21} p_2 = 0, \text{ where } F_{12} = F_{21}^T. \quad (2.23)$$

The 3×3 matrix F_{12} is the so-called *fundamental matrix* [Faugeras et al., 1992]. This constraint is known as the epipolar constraint and arises from the coplanarity of the camera centers of the two views the images points and the space point, the points O_1 , O_2 , p_1 , p_2 and P in Figure 2-4.

The fundamental matrix is singular and its rank is 2 and has 7 degrees of freedom.

Considering Figure 2-4, for each point p_1 in image 1, F_{12} provides the epipolar line $l_{e21} \simeq F_{12}p_1$ in image 2 which contains the corresponding point p_2 .

Computation of fundamental matrix

The relation (2.23) provides a linear equation in the entries of the matrix F . Therefore F can be computed from a set of point correspondences using one of the following methods.

- The Eight-Point Algorithm [Longuet-Higgins, 1981] and its improved version in [Hartley, 1997].
- Non-Linear Least-Squares Approach [Luong et al., 1993] with respect to the coefficients of F , using an appropriate rank-2 parameterization.

2.5 Planar geometry

As this dissertation is concerned with the identification of planes and using them for calibration, a section of this chapter is dedicated to planar geometry. Consider that a single camera is observing a planar scene with a supporting plane not passing through the optical center of the camera. We already know that a mapping from a scene plane to image plane (2D entities), is a linear mapping from \mathbb{P}^2 to \mathbb{P}^2 and hence called a plane homography. We are particularly interested in the exact expression of the 3×3 matrix that represents such homography in terms of the intrinsic and extrinsic parameters of the camera. This will be explored in the following two cases:

- scene plane is at infinity,
- scene plane is at a finite distance from the origin of the reference frame.

2.5.1 Planes at infinity

Let $P_\infty(X_\infty, Y_\infty, Z_\infty, 0)$ be a point on the plane at infinity, Figure 2-5. Its projection onto the image plane is p and is obtained by:

$$p \simeq KR[I \mid -t] \begin{bmatrix} X_\infty \\ Y_\infty \\ Z_\infty \\ 0 \end{bmatrix} = KR \begin{bmatrix} X_\infty \\ Y_\infty \\ Z_\infty \end{bmatrix},$$

$$p \simeq H_\infty \begin{bmatrix} X_\infty \\ Y_\infty \\ Z_\infty \end{bmatrix}, \quad H_\infty \simeq KR, \quad (2.24)$$

where H_∞ is called *infinity-to-image* homography, K is the camera matrix of that view and R is the rotation matrix denoting the rotation between the camera reference frame and the scene reference frame. The projection of a 3D point P_∞ , at infinity, can be a finite point of the image.

The projection of a 3D point at infinity does not depend upon the position of the camera with respect to the world reference frame. It depends only on the orientation of the camera and its intrinsic parameters.

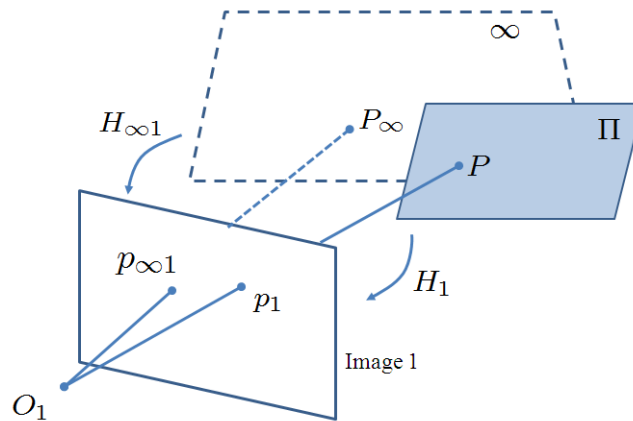


Figure 2-5. Plane to image homography

Planes at finite distance

Let Π be a scene plane observed by a camera. We assume that Π is at a finite distance $d \neq 0$ from the origin of the reference frame and n_{Π} is a normal vector of Π . Then the coordinates of Π is given by

$$\Pi \simeq (n_{\Pi}^T, -d)^T.$$

A scene point P with coordinates $P \simeq (X, Y, Z, 1)^T$ lying on Π satisfies the equation $\Pi^T P = 0$, the incidence equation (2.4). Also the projection of P is defined as $p \simeq MP$, where $M \simeq KR[I \mid -t]$. Therefore:

$$\left\{ \begin{array}{l} n_{\Pi}^T \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = d \\ d \neq 0 \\ p \simeq KR \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} - KRt \end{array} \right. \Rightarrow p \simeq KR \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} - \frac{1}{d} KRtn_{\Pi}^T \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

Thus

$$p \simeq H_{\Pi} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}, \quad H_{\Pi} \simeq KR - \frac{1}{d} KRtn_{\Pi}^T, \quad (2.25)$$

where H_{∞} is the (3×3) infinity-to-image homography matrix described in (2.24). The (3×3) matrix H_{Π} represents the homography induced by plane Π and maps each point P on this plane to its projection on the image plane. The plane Π is located at finite distance

d from the origin, however if we move it to infinite distance then the equation (2.25) changes back to

$$\lim_{d \rightarrow \infty} H_{\Pi} = H_{\infty}.$$

2.5.2 Inter-image homography

Given two images of a same plane Π (Figure 2-6), we have

$$p_1 \simeq H_{\Pi 1} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}, \text{ and } p_2 \simeq H_{\Pi 2} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}, \quad (2.26)$$

from which we can easily deduce that

$$p_2 \simeq H_{\Pi_{12}} p_1 \text{ where } H_{\Pi_{12}} = H_{\Pi 2} H_{\Pi 1}^{-1}. \quad (2.27)$$

$H_{\Pi_{12}}$ is the matrix representation of the inter-image homography between images 1 and 2 induced by plane Π . $H_{\Pi_{12}}$ is a 3×3 matrix with 8 degrees of freedom due to equality up to scale factor.

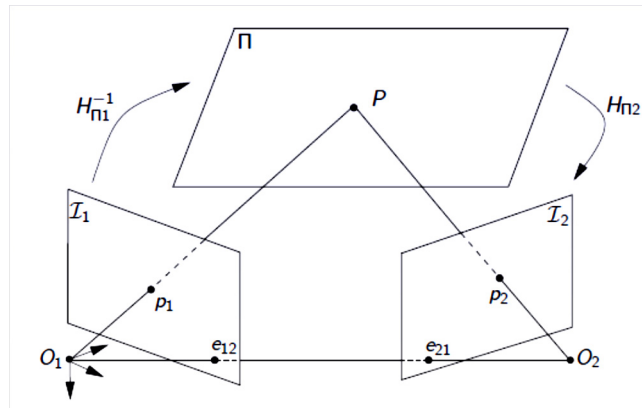


Figure 2-6. Inter-image Homography

From equations (2.27) and (2.25) it can be noticed that for any inter-image plane homography $H_{\Pi 12}$ which relates the projection of points lying on Π onto the two views we have

$$e_{21} \simeq H_{\Pi 12} e_{12}, \quad (2.28)$$

where epipole e_{21} is the projection of camera center of view 1 onto image 2.

Inter-image homography of plane at infinity

If in the equation(2.27) we replace $H_{\Pi 1}$ and $H_{\Pi 2}$ with their equivalent from (2.25) the following expression will be obtained $H_{\Pi 12} \simeq (K_2 R_2 - \frac{1}{d} K_2 R_2 t_2 n_{\Pi}^T) (K_1)^{-1}$. In this equation $d \neq 0$ is the distance of plane Π from the origin of the reference frame and n_{Π} is a normal vector to Π . The following equation can be obtained for the inter-image homography of plane at infinity

$$\lim_{d \rightarrow \infty} H_{\Pi 12} = H_{\infty 12} \Rightarrow H_{\infty 12} = K_2 R_2 K_1^{-1},$$

which implies that the inter-image homography of the plane at infinity only depends on the intrinsic parameters of the views and the rotation between them. The general form of this equation is

$$H_{\infty 1i} = K_i R_i K_1^{-1}, \quad (2.29)$$

where, R_i is the rotation matrix of i^{th} camera with respect to fist view and K_i is the camera matrix of i^{th} camera.

2.5.3 Homography and the plane in projective frame

Assume that in projective space the projection matrices of two views are available and the world reference frame is attached to the first camera. Then

$$M_1 = [I \mid 0], \quad M_2 = [A \mid a] \quad (2.30)$$

where matrix $A(3 \times 3)$ and vector $a(3 \times 1)$ are real-valued. Also consider a projective plane with coordinate $\Pi = (v^T, 1)^T$, not passing the origin, and a point P with coordinates $P = (p_1, \lambda)^T$, on this plane, i.e. $\Pi^T P = 0$. From $p_1 \simeq [I \mid 0]P$ it can be observed that P could be any point with components $P = (p_1, \lambda)^T$. At the same time P is on the plane $\Pi = (v^T, 1)^T$. Therefore

$$(v^T, 1)(p_1, \lambda)^T = 0 \Rightarrow \lambda = -v^T p_1 \Rightarrow P \simeq (p_1, -v^T p_1)^T.$$

Projecting P on the second view, we have

$$\begin{aligned} p_2 &= M_2 P \\ &= [A \mid a](p_1, -v^T p_1)^T \\ &= (A - av^T)p_1, \end{aligned}$$

which defines the homography induced by this plane, $p_2 = H_{12}p_1$, is defined by

$$H_{\Pi 12} = A - av^T. \quad (2.31)$$

2.6 3D Reconstruction Formulation

Suppose m cameras with Euclidean projection matrices $M_i^E(3 \times 4)$, $i \in \{1..m\}$ are observing n 3D scene points $P_j^E(X_j, Y_j, Z_j, 1) \in \mathbb{E}^3$, $j \in \{1..n\}$ in a scene, where the superscript E indicates that the points coordinates are in Euclidean space. Let

$p_{ij} = (u_{ij}, v_{ij}, 1)$ be the 2D projection of point P_j^E onto the image i and is defined by the linear mapping:

$$p_{i,j} \simeq M_i^E P_j^E. \quad (2.32)$$

Problem definition:

In uncalibrated 3D reconstruction, neither the projection matrices M_i^E nor the 3D points P_i^E are available. It will be shown that if the only available inputs are the 2D image points, 3D reconstruction is possible only up to a projective ambiguity. Indeed, instead of P_i^E we can only recover P_j^P which has the following relation with the original scene points

$$P_j^E \simeq T_P P_j^P,$$

where T_P remains an unknown projective matrix. We can lower the ambiguity as we get more information about the cameras or the scene. The lowest level of ambiguity is when the cameras are somehow calibrated. In this case it is possible to recover the scaled version of the original scene, P_j^M (e.g. a *metric reconstruction*): $P_j^E \simeq T_M P_j^M$, where T_M is a metric transformation matrix. The reason behind the reconstruction ambiguity is explained below.

Ambiguity in reconstruction

Let H be an arbitrary 4×4 non-singular matrix representing a projective transformation, as in equation (2.5). Let's define $M_i' \simeq M_i H$ and the projective transformation of the scene points of equation (2.32) as $P_j' \simeq H^{-1} P_j$. Then the following structure can also be recovered from set of image points p_{ij} :

$$p_{ij} \simeq M'_i P'_j.$$

This means from known 2D image points p_{ij} we can obtain solution set $\{M_i, P_j\}$ or anything else like $\{M_i H, H^{-1} P_j\}$ which both are valid solutions of the equations (2.32)! This, in other words, means that from availability of only 2D image points we can reconstruct the scene up to a projective ambiguity H , not a unique solution.

To remove this ambiguity we inevitably need some information about the parameters of the cameras used or about the scene. Once that information is available, we can compute the matrix β to eventually upgrade the projective structure to affine or ultimately to the similarity one. This reduces the projective ambiguity to a similarity ambiguity (i.e. Euclidean up to an arbitrary scale factor). In other words, a *similarity reconstruction* preserves all geometrical features of the scene except for an overall scale factor.

2.6.1 Metric stratum

A scene is referred to as metric if it differs from the true Euclidean one by a metric transformation. Therefore, if points P_j^E undergo a metric transformation T_{EM} , the projection matrices transform into T_{EM}^{-1} (to keep the relation(2.32) valid). The set

$$\{M_i^M, P_j^M\} \simeq \{M^E T_{EM}^{-1}, T_{EM} P_j^E\}$$

is called the metric reconstruction of the original scene. The notation T_{EM} refers to a metric transformation of form defined in equation (2.7) which transforms Euclidean frame to metric frame.

Metric projection matrices, canonical form

Without loss of generality one can choose the matrix T_{EM} that aligns the world reference frame with that of camera 1, thus bringing the metric projection matrices to the canonical form

$$M_1^M \simeq [K_1|0] \text{ and } M_i^M \simeq [K_i R_i | e_{i1}],$$

where R_i is the rotation matrix between the reference and the i^{th} camera. e_{i1} is the projection of the center of camera 1 onto the i^{th} image (equation (2.21)).

2.6.2 Affine stratum

A scene is referred to as affine if it differs from a metric one by an affine transformation. Therefore, if scene points P_j^M , defined in a metric frame, undergo an affine transformation $T_{MA} P_j^M$, the metric projection matrices M_i^M transform to affine ones as follows:

$$\{M_i^A, P_j^A\} \simeq \{M_i^M T_{MA}^{-1}, T_{MA} P_j^M\},$$

where the affine projection matrices of view i take the following canonical form.

$$M_1^A \simeq [I|0] \text{ and } M_i^A \simeq [H_{\infty 1i} | e_{i1}], \quad (2.33)$$

where $H_{\infty 1i}$ is the inter-image homography of the plane at infinity between the reference and the i^{th} views.

2.6.3 Projective stratum

Similarly, a projective scene differs from an affine scene by a projective transformation

T_{AP} :

$$\{M_i, P_j\} \simeq \{M_i^A T_{AP}^{-1}, T_{AP} P_j^A\}.$$

If we define the projective transformation T_{AP} as

$$T_{AP} \simeq \begin{bmatrix} I & 0 \\ v_\infty^T & 1 \end{bmatrix} \quad (2.34)$$

where the v_∞ is the 3-vector of plane at infinity in $\infty = (v_\infty^T, 1)^T$. Then the projective projection matrices of view i take the following canonical form:

$$M_1 \simeq [I|0] \text{ and } M_i \simeq [H_{\Pi 1i}|e_{i1}], \quad (2.35)$$

where $H_{\Pi 1i}$ is the inter-image homography of any arbitrary plane from image 1 to image i . Recall that the inter-image homography of the plane at infinity is related to the inter-image homography of any arbitrary plane Π between any two images i and j by the relationship

$$H_{\Pi ij} \simeq H_{\infty ij} - e_{ij} v_\infty^T. \quad (2.36)$$

Next chapter elaborates on the fundamentals of camera calibration including conics and quadrics.

3 Autocalibration constraints

Camera calibration is the process of determining internal camera parameters and is a necessary step for 3D reconstruction. A typical approach to calibrate a camera involves using one or several images of a set of 3D scene points with known Euclidean structure, called calibration pattern (Figure 3-1). Once a 3D to 2D correspondence has been established between the 3D points and their projections on the images, it is straightforward to recover the camera parameters [Boufama et al., 1993], The steps are listed in the Table 3-1.

Camera *autocalibration*, on the other hand, determines camera parameters directly from multiple uncalibrated images with no need to use any special calibration objects in the scene. Most autocalibration approaches (also referred to as or self-calibration in the literature), make use, instead of a calibration pattern, a virtual pattern which present in all scenes and visible in all images. These virtual patterns are some special conics and quadrics with close ties to the plane at infinity. They play fundamental roles which will be discussed in the following section.

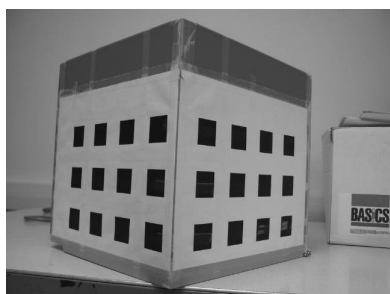


Figure 3-1. Calibration object

Table 3-1. Camera calibration using calibration grid

- 1- Manually measure the 3D position of grid points on the calibration object(5 or more general points)
- 2- Set up the 3D-2D point correspondences between the picked points and their projection onto the images
- 3- Set up the linear equations and solve using methods such as SVD.

3.1 Conics and quadrics

Conics and quadrics are important geometric entities in camera autocalibration approaches. In the following section we elaborate on conics and quadrics and their invariant properties under geometric transformations.

3.1.1 Conics

A conic is a planar 2D entity described by a second-degree equation in its supporting plane. A conic is formed by the intersection of a cone with a plane. In Euclidean space this intersection can take the form of an ellipse, parabola, hyperbola or any of their degenerate forms which are all quadratic expressions of points in \mathbb{E}^2 (2D Euclidean space). A conic C is defined as the locus of 2D points p for which the homogeneous equation below is valid:

$$p^T C p = 0, \quad (3.1)$$

where C is a 3×3 symmetric matrix which, due to representing its points $p = (x, y, 1)$ in homogeneous, is defined up to scale. Therefore instead of 6 parameters it can be defined with only 5 independent parameters.

$$\begin{aligned} [x \ y \ 1] \begin{bmatrix} a_1 & a_2 & a_3 \\ a_2 & a_4 & a_5 \\ a_3 & a_5 & a_6 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} &= 0 \\ \Rightarrow [x \ y \ 1] \begin{bmatrix} \frac{a_1}{a_6} & \frac{a_2}{a_6} & \frac{a_3}{a_6} \\ \frac{a_2}{a_6} & \frac{a_4}{a_6} & \frac{a_5}{a_6} \\ \frac{a_3}{a_6} & \frac{a_5}{a_6} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} &= 0, \end{aligned}$$

where a_6 is assumed to be non-zero. From (3.1) each point p places one constraint on the conic coefficients. Therefore, five points in general position are enough to determine a conic uniquely up to scale.

Dual Conic.

Like other entities in projective geometry, the conic has its dual too. As in \mathbb{P}^2 the dual of a point is a line, the dual of conic of points C defined in (3.1) is a conic of lines and is denoted by C^* :

$$l^T C^* l = 0 \quad (3.2)$$

where l is a set of lines tangent to the points on the conic C . C^* is the adjoint (adjugate) matrix of C . If C is a nonsingular matrix its dual C^* is described by the inverse of matrix C :

$$C^* \simeq C^{-1}. \quad (3.3)$$

Tangent to conic.

In \mathbb{P}^2 , the line l tangent to conic C at a point p on C , is given by

$$l = Cp = C^T p,$$

since C is symmetric. From the definition of tangent, p is the only common point on the conic C and the line l and therefore by referring to the equation (3.1)

$$p^T l = 0, p^T Cp = 0 \Rightarrow l = Cp.$$

Transformations of Conics

If 2D points p transform to 2D points p' under the 2D homography H , $p' = Hp$ then the conic C formed by the locus of these points transforms to $C' = H^{-T}CH^{-1}$:

$$p^T C p = 0, p = H^{-1}p' \Rightarrow p'^T H^{-T} C H^{-1} p' = 0$$

and thus

$$C' = H^{-T} C H^{-1}. \quad (3.4)$$

Similarly if points p transform to 2D points p' under the 2D homography $H_{(3 \times 3)}$, $p' = Hp$ the line formed by points p transforms to $l' = H^{-T}l$. Therefore, the line conic C^* transforms to $C^{*'} = HC^*H^T$:

$$l^T C^* l = 0, l = H^T l' \Rightarrow (H^T l')^T C^* (H^T l') = 0 \Rightarrow l'^T H C^* H^T l' = 0$$

and therefore

$$C^{*'} = H C^* H^T. \quad (3.5)$$

From equations (3.6) and (3.7) it can also be implied that $C'^* = C'^{-1}$ for nonsingular C .

3.1.2 Quadric and its dual

Quadrics

As mentioned earlier, conics represent quadratic expressions of 2D points in \mathbb{P}^2 , whose extensions to \mathbb{P}^3 are called quadrics. Quadrics are described by quadratic expressions in \mathbb{P}^3 . Therefore a quadric Q is a surface in 3D (not necessarily a flat one). When 3D points P are represented by their homogeneous coordinates, a quadric Q can be described by a

symmetric matrix. A general symmetric 4×4 matrix is described by ten parameters, but since points are represented in their homogeneous coordinates, out of ten parameters only nine are independent. A quadric Q can be described in projective space \mathbb{P}^3 as follows

$$P^T Q P = 0, \quad (3.8)$$

where Q is a *point quadric* which contains the 3D points $P \in \mathbb{P}^3$. For brevity, point quadrics are also referred as quadrics in this desertion.

Dual Quadric.

As mentioned in Section 2.1.3, the dual of points are planes in projective space \mathbb{P}^3 . Therefore the dual of a point quadric Q is defined by a set of planes tangent to the points of quadric Q . The dual of point quadric Q is a plane quadric Q^* which is the adjugate of matrix Q . In case Q is nonsingular, Q^* is the inverse of Q :

$$\Pi^T Q^* \Pi = 0, Q^* = Q^{-1} \quad (3.9)$$

where plane Π is a plane tangent to a point on Q .

Transformation of quadrics.

Transformations on quadrics and dual quadrics follow the same rules as for the conics and dual conics. Therefore if points $P \in \mathbb{P}^3$ of equation (3.8) transforms to $P' \simeq HP$, the transformation of corresponding quadric Q and dual quadric Q^* are given by

$$\begin{aligned} Q' &= H^{-T} Q H^{-1} \\ Q^{*'} &= H Q^* H^T, \end{aligned} \quad (3.10)$$

where H in here is a 4×4 3D homography as opposed to H in (3.5) and (3.4) which is 3×3 .

3.2 Absolute Conic (AC) and Dual Absolute Quadric (DAQ)

The Absolute Conic (AC) is a virtual squashed cone in 3D space located on the plane at infinity. A cone is a degenerate quadric so its representing 4×4 matrix does not have full rank. The AC was introduced to 3D Computer Vision by Faugeras [Faugeras and Maybank, 1990] and plays a key role in camera autocalibration. The recovery of the image of the AC or its dual in standard form results in the metric recovery of the entire scene structure.

We denote the absolute conic with the symbol Ω_∞ . In 3D Euclidean space \mathbb{E}^3 , the absolute conic consists of points $P = (x^T, 0)^T$ on the plane at infinity which satisfy $x^T x = 0$, where $x = (x_1, x_2, x_3)$. Therefore in Euclidean 3D space \mathbb{E}^3 , it contains only imaginary points. This conic is called absolute as it resides in the plane at infinity and consequently is invariant (and absolute) under the metric transformations:

$$T_M^{-T} \Omega_\infty T_M^{-1} \simeq \Omega_\infty, \quad (3.11)$$

where T_M is a metric transformation (equation (2.7)). The invariance of the AC under is an important property exploited in camera autocalibration as its position relative to a moving camera is constant.

3.2.1 Dual Absolute Quadric

The dual of the absolute conic is a degenerate dual quadric in 3-space and is called Dual of Absolute Quadric (DAQ) which we denote with the symbol Q_∞^* . Any plane in the

envelope of the dual of absolute quadric Q_{∞}^* is tangent to the absolute conic Ω_{∞} . (Figure 3-2)

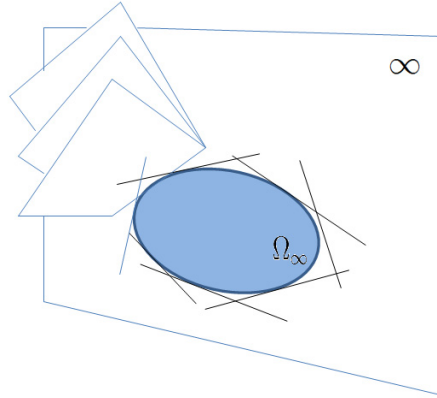


Figure 3-2. The absolute conic on plane at infinity

This terminology may sound unsuitable as the dual of a conic cannot be a dual quadric. Nonetheless, the terminology Dual Absolute Quadric is commonly used in literature and we stick with it too, otherwise Dual Absolute Conic has been used in some literature too [Cyganeck et al., 2009].

In practice it is often preferred to represent the Ω_{∞} through its dual, Q_{∞}^* . This way both the Ω_{∞} and its supporting plane Π_{∞} , are expressed in one geometric entity. Algebraically Q_{∞}^* is represented by a 4×4 matrix of rank 3. The DAQ has the following canonical form in 3D Euclidean space:

$$Q_{\infty}^{*E} = \begin{bmatrix} I & O \\ O^T & 0 \end{bmatrix}, \quad (3.12)$$

where O is a 3-null-vector.

Dual of Absolute Quadric under transformations

In the following discussion, consider the definition of geometric transformations in Table 2-1 and transformation of quadrics in equation (3.10).

Euclidean: Q_{∞}^{*E} is invariant under Euclidean transformations.

Metric: Q_{∞}^{*E} is scaled under similarity transformations (i.e. $Q_{\infty}^{*E} \simeq \lambda Q_{\infty}^{*M}$).

Affine: Q_{∞}^{*E} takes the form $Q_{\infty}^{*A} = \begin{bmatrix} A^T A & O \\ O^T & 0 \end{bmatrix}$, under affine transformations.

Projective: Under projective transformations, Q_{∞}^{*E} becomes an arbitrary *symmetric positive semidefinite* rank 3 matrix Q_{∞}^* [Triggs, 1997].

Remark: Q_{∞}^* is a projective encoding of Euclidean and Affine structure of the scene. Euclidean as it encodes the intrinsic camera parameters, and affine as it also encodes the plane at infinity. This property makes DAQ the an important geometric entity of camera autocalibration on 3D computer vision.

3.2.2 Properties of Dual Absolute Quadric

The DAQ Q_{∞}^* has the following properties which can be considered in autocalibration approaches. The dual absolute quadric Q_{∞}^* is a degenerate quadric that has 8 degrees of freedom. The reason is that a symmetric 4×4 matrix has 10 independent entries. However, a 4×4 matrix which represents a quadric Q_{∞}^* in projective space has one less degree of freedom due to the scale stem from homogeneous coordinates. Also, as the DAQ is a degenerate quadric, it again reduces the degree of freedom one less to 8. Moreover,

- Null vector of Q_{∞}^* is the plane at infinity denoted by ∞ :

$$Q_{\infty}^* \infty = 0 \quad (3.13)$$

- Q_{∞}^* is a projective entity which encodes the Euclidean angle between two planes ϕ and φ represented in projective space:

$$\cos(\theta) = \frac{\phi^T Q_{\infty}^* \varphi}{\sqrt{(\phi^T Q_{\infty}^* \phi)(\varphi^T Q_{\infty}^* \varphi)}} \quad (3.14)$$

Remark 1. The two planes ϕ and φ are orthogonal if and only if

$$\phi^T Q_{\infty}^* \varphi = 0. \quad (3.15)$$

Remark 2. Q_{∞}^* is not full rank, it is of rank 3. As a result its determinant is zero:

$$\det(Q_{\infty}^*) = 0, \quad (3.16)$$

The terms ‘‘Absolute Dual Quadric’’ and ‘‘Dual of Absolute Quadric’’ are used interchangeably in the literature [Zhu et al., 2009], [Triggs, 1997] and [Hartley and Zisserman, 2004], [Gurdjos et al., 2010], [Gurdjos et al., 2010], respectively. We stick with the latter (DAQ) in this thesis.

3.2.3 The images of AC and DAQ

As mentioned above, in Euclidean 3D space the absolute conic Ω_{∞} is consist of a set of points on plane at infinity. Therefore it contains only points with complex coordinates. Projection of any complex 3D point P of Ω_{∞} onto an image is also a complex point u on a 2D conic ω . ω is called the Image of Absolute Conic (IAC) and is represented by a full rank 3×3 symmetric matrix.

Similarly the projection of the dual of absolute quadric Q_∞^* onto an image yields a conic of lines [Hartley and Zisserman, 2004] which is the dual of image of absolute conic and is denoted by ω^* .

$$\omega_i^* \simeq M_i Q_\infty^* M_i^T \quad (3.17)$$

where M_i is the projection matrix of i^{th} view in projective space. This relationship is straightforward when the scene and cameras are given in the Euclidean or metric space. For instance, considering M_i^E , the Euclidean (or metric) projection matrix of view i and ω_i^* the projection of Q_∞^{*E} onto the image I we have:

$$\begin{aligned} M_i^E Q_\infty^* M_i^{ET} &\simeq K_i [R_i^T | -R_i^T t_i] \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 0 \end{bmatrix} [R_i^T | -R_i^T t_i]^T K_i^T \\ &= K_i R_i R_i^T K_i^T \\ &= K_i K_i^T = \omega_i^*, \end{aligned} \quad (3.18)$$

where R , the rotation matrix is an orthogonal matrix and therefore $RR^T = I$. K is the intrinsic parameters matrix described in equation (2.17). Therefore, ω^* becomes a symmetric positive semi-definite matrix. Since ω^* is a symmetric matrix, it represents a conic. The good thing about this conic is that it only depends on the camera parameters of that view and nothing else.

In projective space. In reality, the projection matrices and 3D structure of the scene that we can recover from multiple view feature matches are valid up to an arbitrary projective ambiguity. This means what we can obtain is the projective projection matrix. Projective projection matrices can be upgraded to Euclidean ones for some nonsingular H such that

$$M^E = MH, \quad (3.19)$$

where M^E is the Euclidean projection matrix which models the Euclidean camera pin-hole model that we expect to get. Likewise, the Dual of Absolute conic Q_∞^* is not recovered in its canonical position (i.e. equation (3.12)), but in its transformed position

$$Q_\infty^* = HQ_\infty^{*E}H^T. \quad (3.20)$$

We will observe that although the DAQ has undergone a projective transformation and changed to an arbitrary symmetric matrix, its projection is the same as the projection of original Q_∞^{*E} as in (3.18). In the following statements we replace M_i with $M_i = M_i^E H^{-1}$ from equation (3.19) and Q_∞^* from equation (3.20), hence

$$\begin{aligned} M_i Q_\infty^* M_i^T &= (M_i^E H^{-1}) H Q_\infty^{*E} H^T (M_i^E H^{-1})^T \\ &= M_i^E Q_\infty^{*E} M_i^{E^T} \\ &\simeq K_i K_i^T = \omega_i^*, \end{aligned}$$

which implies that *regardless of the position of the Q_∞^* , whether canonical or projective, the image of Q_∞^* is constant and depends only on the intrinsic camera parameters of that view*

$$M_i Q_\infty^* M_i^T \simeq K_i K_i^T = \omega_i^*, \quad (3.21)$$

where Q_∞^* is now in any general projective position and M_i is projective projection matrix of view i . Equation (3.21) is one of the most important equations for auto camera calibration as it relates the intrinsic camera parameters to the dual image of absolute conic.

A few remarks here:

Remark 1: From equation (3.21), Q_∞^* is the same and constant for all the views, however its image in each view might be different. In other words, the back projections of each ω_i^* is a dual quadric in space, and the dual of absolute quadric Q_∞^* must lie in the intersection of all the dual quadrics. So Q_∞^* is part of the scene and remains constant regardless of changing camera motions and parameters.

Remark 2: Q_∞^* depends on only two entities, the intrinsic parameters of the reference camera and the projective location of the plane at infinity with respect to the projective frame, that is:

$$Q_\infty^* \simeq \begin{bmatrix} \omega^* & -(\omega^* v_\infty) \\ -(\omega^* v_\infty)^T & v_\infty^T \omega^* v_\infty \end{bmatrix}, \quad (3.22)$$

where v_∞ is the 3-vector of plane $\infty \simeq (v_\infty^T, 1)^T$ and ω^* is the dual image absolute conic of reference view.

Remark 3: The dual of image absolute quadric (DIAC) in each view depends only on the intrinsic parameters of that view. Therefore if same camera with constant parameters is used to record all the images, DIACs of all views will be identical.

Remark 4: ω^* measures the 3D angle between planes that pass through the optical center of the camera. This property of ω^* in 2D projective space \mathbb{P}^2 is similar to what Dual of Absolute Quadric (DAQ) provides in \mathbb{P}^3 for Euclidean angles between planes.

Remark 5: From remark 2, the intrinsic parameter of camera K can be recovered from ω^* in the equation below, by using the Cholesky decomposition [Trefethen and Bau, 1997] :

$$\omega^* = K K^T \quad (3.23)$$

It can be observed that a plane conic ω^* has five degrees of freedom, so does the camera calibration matrix [Maybank and Faugeras, 1992].

Remark 6: The Image of Absolute Conic and its dual, DIAC, can also be mapped from one image to another image by the homography of its supporting plane, the Π_∞ :

$$\omega_j \simeq H_{\infty ij}^{-T} \omega_i H_{\infty ij}^{-1} \quad \text{or} \quad \omega_j^* \simeq H_{\infty ij} \omega_i^* H_{\infty ij}^T \quad (3.24)$$

Equation (3.23) gives a different definition to camera calibration, it implies that locating the DIAC is equivalent to calibrating the camera.

3.3 Autocalibration

Camera *autocalibration* is the process of determining camera parameters directly from multiple uncalibrated images with no special calibration object in the scene. Instead of a real calibration object, the IAC, the DIAC or the DAQ, which is naturally present in all scenes, are used to retrieve the camera parameters. The invariance properties of these special virtual objects and their relationship to the camera parameters are exploited to achieve this task. Imposing constraints in autocalibration approaches can be achieved in three ways, 1) constraints on the intrinsic camera parameters 2) constraints on motion parameters 3) scene constraints, where the contribution of this dissertation lies. The following main approaches are at the core of most autocalibration methods presented in the literature:

- using Kruppa's equations which relate DIACs in two views through epipolar geometry,
- retrieving the DAQ from equation (3.21),

- through stratified techniques by imposing the modulus constraint to locate the enclosing plane of the absolute conic, the plane at infinity.

As mentioned above, the intrinsic camera parameters are related to the Dual Image of Absolute Conic by the equation (3.23) which is expanded below

$$\omega^* = K K^T = \begin{bmatrix} f_u^2 + s^2 + u_0^2 & s f_v + u_0 v_0 & u_0 \\ s f_v + u_0 v_0 & f_v^2 + v_0^2 & v_0 \\ u_0 & v_0 & 1 \end{bmatrix}, \quad (3.25)$$

where K is the upper triangular camera matrix defined as

$$K = \begin{bmatrix} f_u & s & u_0 \\ & f_v & v_0 \\ & & 1 \end{bmatrix}.$$

Equation (3.25) camera intrinsic parameters can be uniquely computed using Cholesky decomposition [Jennings et al., 1977] which simplifies in this case to

$$\begin{aligned} u_0 &= \omega_{31}^*, \quad v_0 = \omega_{32}^*, \quad f_v = \sqrt{\omega_{22}^* - v_0^2} \\ s &= (\omega_{12}^* - u_0 v_0) / f_v, \quad f_u = \sqrt{\omega_{11}^* - (s^2 + u_0^2)} \end{aligned} \quad (3.26)$$

3.3.1 Autocalibration using Kruppa equations

Kruppa's equations were originally derived by Kruppa in 1913 [Kruppa, 1913]. However, the equations were introduced into Computer Vision in 1992 by Maybank and Faugeras [Maybank and Faugeras, 1992] for camera autocalibration. Their work is historically seen as the first autocalibration method. The Kruppa equations are two-view constraints that require only the fundamental matrix F and consist of two independent quadratic equations in the elements of DIAC. From a geometrical point of view, Kruppa's

equations impose the constraint that the epipolar planes tangent to the AC project to corresponding epipolar lines tangent to the IAC in each view. Kruppa's equations can be obtained from equation (3.24) by restricting the constraints to epipolar geometry [Pollefeys, 2004]:

$$[e_{ji}]_x^T \omega^* [e_{ji}] \simeq F_{ij} \omega^* F_{ij}^T, \quad (3.27)$$

where F_{ij} is the fundamental matrix between views i and j and e_{ji} the epipole in image j which is the projection of center of camera i . Three views are in theory sufficient to solve for the five internal parameters. However, solving the obtained six quadratic equations turned out to be difficult. Improved versions of that work can be found in [Zeller et al., 1996] and [Luong and Faugeras, 1997].

The problem with Kruppa equations is that they restrict the autocalibration constraints to the epipolar geometry which is equivalent to the removal of the plane at infinity from the equations, as seen in (3.27). This may lead to poor results [Sturm, 2000] as compared to other general autocalibration methods. For more information on Kruppa equations the reader is referred to [Hartley and Zisserman, 2004], or [Luong and Faugeras, 1997].

3.3.2 Direct Autocalibration from DAQ

In practice it is preferable to work with DAQ [Heyden and Astrom, 1996], [Triggs, 1997] as it encodes both the dual image of absolute conic and the enclosing plane of AC in one geometric entity. Working with the DAQ allows to enforce the unity of the plane on which the AC lies and therefore enforce more constraints on the sought DIAC. Another advantage of using the DAQ is that the autocalibration equations involve plane at infinity,

the DIAC and the projective projection camera as described in equation (3.21). One of the early approaches in this class is proposed in [Triggs, 1997] for fixed camera parameters.

The following are considered while taking DAQ based calibration approaches. Let's the estimated DAQ be denoted by \hat{Q} , a 4×4 symmetric matrix which has eight degree of freedom:

$$\hat{Q} = \begin{bmatrix} q1 & q2 & q3 & q4 \\ q2 & q5 & q5 & q7 \\ q3 & q5 & q8 & q9 \\ q4 & q7 & q9 & q10 \end{bmatrix} \quad (3.28)$$

The computed \hat{Q} has to meet the following properties of a DAQ matrix [Hartley and Zisserman, 2004].

Property 1. It has to be of rank 3. Due to numerical error, it is test test the rank 3 of estimated \hat{Q} . We instead, verify the necessary condition of rank deficiency, $\det(\hat{Q}) = 0$. Again, due to numerical error, we may not get exact zero. Therefore, we can check the ratio of smallest eigen value of \hat{Q} over the second smallest eigen value. This quantity has to be very close to zero. Enforcing rank 3 can be done though SVD decomposition.

Property 2. It is positive semi-definite, (or negative - depending on the homogeneous scale). All eigen values of \hat{Q} must have same sign, all positive or all negative.

It is common practice to imposed constraints on the intrinsic parameters of the camera, such as zero skew and unit aspect ratio to obtain an initial, computation of the C^* followed by a more comprehensive refinement, through a linear decomposition such as SVD or though nonlinear ones [Chandraker et al., 2007], taking into account some or all the properties of the DAQ.

3.3.3 Stratified autocalibration

In a stratified autocalibration approach, first the location of plane at infinity is identified in projective space, upon which an affine reconstruction [Habed et al., 2010a] is obtained. This can be followed by computation of the cameras' intrinsic parameters to upgrade the recovered affine structure to metric [Pollefeys and Gool, 1999]. This last step can be carried out using the inter-image homography of plane at infinity $H_{\infty ij}$. Equation (3.24) can be reconfigured to a set of linear equations in the coefficients of image of absolute conic or its dual (DIAC) [Hartley, 1994a]. Finally, the intrinsic camera parameters can be retrieved from the IAC or DIAC either through Cholesky decomposition (3.26).

For the identification of plane at infinity Moons et al. [Moons et al., 1996] proposed a method for pure translation camera motion. Later, Pollefeys et al. [Pollefeys and Gool, 1999] proposed an affine reconstruction method based on using modulus constraints for unknown but constant intrinsic camera parameters in views. Another method is introduced in [De Agapito et al., 1999] for identification of plane at infinity for a zooming camera.

3.3.4 The modulus constraint

The modulus of a real eigenvalue λ is its absolute value $||\lambda|| = \sqrt{\lambda^2}$. The modulus of a complex eigenvalue λ is the square root of multiplication of λ to its complex conjugate:

$$||\lambda|| = \sqrt{\lambda \times \lambda^*},$$

where λ^* is the complex conjugate of λ .

The modulus constraint was first used for camera calibration by Pollefeys et al. in 1996 [Pollefeys et al., 1996]. It states that *if the camera parameters remain constant for all the views, the inter-image homography of plane at infinity will all have equal moduli*.

Recall the relation in equation (2.29) between the camera matrices and inter-image homography of the plane at infinity. Now that we assumed all the camera parameters are the same, equation (2.29) simplifies to:

$$H_{\infty i} \simeq KR_iK^{-1}, \quad (3.29)$$

where R_i is the rotation matrix and K is the camera matrix. Since KR_iK^{-1} is conjugate to rotation matrix, it has the same eigenvalues as of R_i . Therefore the plane at infinity has eigenvalues $\{\lambda, \lambda e^{i\theta}, \lambda e^{-i\theta}\}$ which all have equal moduli $\{\lambda, \lambda, \lambda\}$.

Remark. If we scale $H_{\infty i}$ so that its determinant is unity,

$$H_{\infty i} \leftarrow H_{\infty i} / \det(H_{\infty i})^{1/3}.$$

its eigenvalues will be exactly those of the rotation matrix relating the considered views, i.e. $\{1, e^{i\theta}, e^{-i\theta}\}$. Thus, when a constant parameter camera is used to capture the scene, the characteristic equation

$$\det(H_{\infty i} - \lambda I) = 0 \quad (3.30)$$

must have roots (eigen values) with all equal moduli.

3.3.5 Chirality constraints

In stratified autocalibration approaches, a projective reconstruction is first found and then upgraded to metric. The problem with the projective reconstructed scene is that the scene is often split by the plane at infinity [Hartley and Zisserman, 2004]. Some of the points locate in front of the camera and some behind it. The additional constraint which bounds the points to lie in front of the camera is called chirality constraint [Hartley, 1998] and is imposed by linear inequality constraints.

This simple constraint is useful in determining quasi-affine reconstructions as an intermediate step towards affine and metric reconstruction [Hartley and Zisserman, 2004]. The quasi-affine reconstruction with respect to camera centers is a projective reconstruction that does not contain any twisted pairs [Nistér, 2004]. However, with respect to the scene, the quasi-affine reconstruction is a projective reconstruction that preserves the convex hull of the scene [Chandraker, 2009]. In brief, this additional constraints to projective reconstruction gives us a projective reconstruction which is also a quasi-affine reconstruction.

In nonlinear 3D reconstruction optimization approaches, chirality constraints are incorporated to automatically select an initial region which is guaranteed to contain the global minimum [Chandraker et al., 2010].

3.3.6 Constraints on intrinsic camera parameters

Some constraints on the intrinsic parameters of a camera can be considered (e.g. square, or rectangular pixels, principal point in the center of the image, etc) during camera autocalibration. By imposing constraints on these parameters we intend to decrease the number of unknowns so that enough equation are available to solve the set of equations

parameters. As well, we can resolve ambiguities in the case of critical (or close to critical) motions.

As introduced in Chapter 2, Equation (2.17), the five intrinsic parameters are embedded in a 3×3 matrix K with intrinsic parameters. The combination of (3.25) and (3.21), $M_i Q_\infty^* M_i^T \simeq K_i K_i^T = \omega_i^*$, brings us to the following detailed equation

$$\omega_i^* = K_i K_i^T = \begin{bmatrix} f_u^2 + s^2 + u_0^2 & s f_v + u_0 v_0 & u_0 \\ s f_v + u_0 v_0 & f_v^2 + v_0^2 & v_0 \\ u_0 & v_0 & 1 \end{bmatrix}_i \simeq M_i Q_\infty^* M_i^T, \quad (3.31)$$

which implies that imposing constraints on the entries of the intrinsic parameters matrix provides us with simpler form of the autocalibration equations, possibly linear, or with fewer unknowns to eventually estimate the eight unknowns of the symmetric matrix dual absolute quadric \mathbb{P}^3 .

Frequently used constraints on camera internal parameters are based on the following assumptions.

Same camera with varying parameters

A helpful assumption is that it is typically the same camera that is used to take all images. This assumption results in dealing with fixed aspect ratio and fixed skew factor cameras in all sequences, but the focal length and principal point may change for each view.

Same camera with constant parameters

In recent years many researchers have been working on the scenarios where all the used cameras are assumed to have unknown but the identical intrinsic parameters. The earliest work in this group belongs to [Faugeras et al., 1992] in which Kruppa's equations are

used. With the same assumption, [Triggs, 1997] proposes the use of quasi-linear constraints for the dual absolute quadric to minimize equation (3.21). Another approach is introduced in [Pollefeys and Van Gool, 1997] where the equation (3.24) is used to compute the DIAC from the homography of the plane at infinity.

Pollefeys & Gool use the constant intrinsic camera parameters in a different approach [Pollefeys and Gool, 1999], [Pollefeys et al., 1996]. They incorporate the so-called modulus constraint on the plane at infinity which is only valid upon the assumption of constant intrinsic camera parameters.

The location of the principal point is known for all cameras

A common approach in this category is to assume that the principal points are in the middle of images so that a linear initial solution can be found for the DAQ. The solution is later refined via a nonlinear optimization process during which the intrinsic parameters of the images are allowed to vary independently [Heyden and Astrom, 1996], [Heyden and Astrom, 1999].

Zero skew and unit aspect ratio

Zero skew factor is a valid assumption in digital cameras manufactured today [Seo and Heyden, 2004]. Huynh & Heyden [Huynh and Heyden, 2005] assume use of a camera with zero skew and unit aspect ratio in their autocalibration approach. They also mentioned that even if the two entities are not known, they can at least be assumed constant when same camera is used throughout the sequence. As these two entities are known to be invariant under changes of focus and can be calculated beforehand. In their work, scene constraints are also incorporated with an assumption on identified orthogonal

planes in the scene. The orthogonality of planes is used in the nonlinear optimization and the bundle adjustment phase.

3.3.7 Constraints on camera motion

Many autocalibration approaches have been proposed for restricted camera motions, such as pure translation, often combined with constraints on the camera's intrinsic parameters. In several cases, this restriction leads to simpler algorithms. However, considering the discussion on critical camera motion, not all of these algorithms could lead to a full calibration of cameras to metric [Pollefeys, 2004]. Moons et al. [Moons et al., 1996] proposed a method for pure translation camera motion for affine camera calibration. Hartley [Hartley, 1994b] reported a calibration method for pure rotations and in another approach Armstrong et al. [Armstrong et al., 1996] for planar motion [Espuny et al., 2011].

Camera motion constraints might be critical specifically for some autocalibration methods. For example in [Sturm, 2000], Sturm reports a camera motion for which the Kruppa equation approach fails but does fine for other methods. That is if the optical centers of cameras locate on a sphere and the optical axes of those cameras pass through the sphere's center. More information on relevant recent approaches can be found in [Sturm et al., 2011].

3.3.8 Scene constraints

In many scenarios, we cannot enforce enough constraints on the cameras, however some geometrical knowledge of the scene might be available to support calibration [Criminisi et al., 2000], [Gong and Xu, 2004], [Kawasaki and Furukawa, 2009]. In many approaches

the scene constraints are used in conjunction with other types of constraints such as camera constraints to autocalibrate cameras [Triggs, 1997], [Huynh and Heyden, 2005] and obtain more robust results. The constraints which can be imposed on the scene are based on the properties which are preserved under affine or metric transformations.

Affine scene constraints

Affine scene constraints are properties which are invariant under affine transformations, but may change under general projective transformations. Parallelism of planes (or lines) and relative distances between collinear points are affine scene constraints. For example, if a representation of a cube in projective frame is given. Knowing this is a cube, the three vanishing points, where parallel lines intersect, can be identified. The plane containing these 3 vanishing points is the plane at infinity. As explained earlier, once the plane at infinity is located in projective frame, the reconstruction can be upgraded to affine by the transformation which brings the plane at infinity to its canonical form.

As another affine constraint, length ratios along a straight line define the point at infinity of that line. If three of such non-collinear configurations exist in the images, the location of plane at infinity can be recovered, and so the affine reconstruction.

A recent approach is proposed in [Habed et al., 2010a] which uses parallel planes in the scene to affine calibrate the camera.

To impose the aforementioned constraints one need to know whether they are available in the scene. One way would be to assume that a human operator specifies the constraints via a GUI program. This makes the calibration based on scene constraints a time consuming process. Therefore, a more preferable way is to use techniques that automatically identify those geometric constraints in the scene. In Chapter 5 of this

dissertation we elaborate on our proposed method which automatically detects parallel planes in a planar scene [Habed et al., 2010b].

Metric scene constraints

Metric scene constraints could be any geometric entity or property which is invariant under metric transformation and variant under lower level transformations (affine, projective). Distances, angles, orthogonality of lines or planes are examples of metric constraints.

Bill Triggs proposed one of the earliest approaches of autocalibration with the incorporation of scene orthogonal planes [Triggs, 1997] in 1997. Liebowitz and Zisserman [Liebowitz and Zisserman, 1999], introduced their method to use projection of parallel and perpendicular scene lines for the estimation of vanishing points and the IAC/DIAC. Huynh & Heyden [Huynh and Heyden, 2005] propose an approach similar to [Triggs, 1997] for cameras with zero skew and unit aspect ratio. In [Huynh and Heyden, 2005] also orthogonal scene planes are used as constraints for the estimation of the DAQ followed by bundle adjustment.

4 The Distinction Between Virtual and Physical Planes

In this chapter, identification of planes from a pair of uncalibrated images will be studied. Identification of planes from a pair of uncalibrated stereo images is a challenging problem as it can lead to extracting virtual planes instead of physical ones, especially for complex scenes. We propose a new homography-based approach to extract physical planes and to distinguish them from virtual ones for general scenarios. The proposed method takes two uncalibrated images as input, extracts and matches interest points and, then performs plane identification and matching defined by sets of three points only. Based on these three points, a plane homography is then calculated. Once all possible planes have been identified, a merging stage is carried out to make sure that same planes are associated with a single homography. This work uses non-coplanar points inside a plane to decide whether the plane is physical or virtual. Depending on the distribution of the points inside the convex hull of the plane, the plane is classified as likely virtual, likely physical, very likely virtual or very likely physical. To estimate homographies, we use our method which computes the homography for three points with no necessity to assume the fourth point being coplanar with the three. Experiments on real images demonstrate the validity of the proposed approach for general scenarios.

4.1 Problem Statement and Related Work

Man-made environment is abundant with planes which represent a major share of both indoor and outdoor scenes. Identifying planes helps us processing image pixels in groups

not individually. Planes are a high level information that could lead to faster and more accurate algorithms in applications such as robot navigation [Sugimoto and Okutomi, 2007], camera calibration [Chuan et al., 2003], 3D reconstruction [Wang et al., 2005] [Pollefeys et al., 2008] and even curved surfaces partitioning [Su et al., 2007].

There has been a number of plane detection approaches proposed over the past decade. Vincent and Laganier [Vincent and Laganier, 2001] used groups of four points which are likely to be coplanar, to compute the homography. To pick the four points, the authors' strategy is to select points lying on the two different edges in an image. Therefore the success of their approach depends on the coplanarity of the four points selected and is very dependent on the results of the edge-detector. Nicolas et al [Nicolas et al., 2005] proposed a method for computing the second homography provided the first homography is already given. In [Nicolas et al., 2005] it is mentioned that no need to consider the detection of different homographies as independent processes. This is due to the fact that the knowledge of first homography eases the detection of the second homography by using only three matched points. However, this technique suffers from a major drawback, if the first homography is not very accurate, the detection of the second homography will not be successful. A different approach is proposed in [Piazzini and Prattichizzo, 2006], where normal vectors are used to detect scene planes in a pair of stereo images. When two adjacent planes have the same normal they are merged. However, this method requires that the two cameras to be aligned to the same orientation with the same calibration parameters. In [He and Chu, 2006] a method is proposed to detect major planes in stereo images using the epipolar geometry and plane homography which is calculated using the 3 points and the epipole. The homography calculated with

using the epipole as 4th point could be very unstable as the epipole location is not usually accurate. In addition, their method did not address the issue of virtual planes.

A major issue with the existing homography-based plane detection approaches is that the detected plane could be a virtual plane not physical (Figure 4-1. Virtual plane). In such approaches, homographies are generally computed from matched interest points of a pair of stereo images and are used to verify coplanarity of points. Then, each group of coplanar points is considered as a physical plane of the scene [Choi et al., 2007], [He and Chu, 2006], which is not always true. In fact, a group of scene interest points may be coplanar; however they may not lie on a physical plane of the scene. Few applications may work fine even if the extracted plane is virtual[Wang et al., 2008]. However, many applications such as navigation [Simond08], detection of independent moving objects [Kirchhof, 2008] and 3D reconstruction[Kawakam et al., 2006] fail and do not produce reliable results.



Figure 4-1. Virtual plane

In [Simond and Rives, 2008], extraction of physical planes is essential. Simond and Rives propose detection of the urban road plane from a sequence of uncalibrated

stereo images. For the initialization step, the authors assume that the detected plane with highest number of inliers is the road plane. This assumption may work for special scenes but is not a valid solution for general cases, such as if the road markers are blocked by cars ahead. In [Wang et al., 2005] the authors use the four point method [Hartley and Zisserman, 2004] with RANSAC to compute homographies. Lines are extracted from images and used for verification of plane boundaries. One of the problems with using lines [Zeng et al., 2008] is that if extracted lines are parallel with the epipolar lines, coplanarity of lines cannot be accurately verified.

With a few exceptions [Kawakam et al., 2006], [Kirchhof, 2008] most of the existing plane extraction approaches did not elaborate on the possibility of existence of virtual planes [Choi et al., 2007], [He and Chu, 2006]. Kawakami et al [Kawakam et al., 2006], propose an approach which is based on the idea that “if there is a plane in a scene, the points on that plane may be projected onto a small region in the image” and thus feature points in a cluster of points are likely laid on a physical plane. Therefore, for each region of points, the authors [Kawakam et al., 2006], pick different combinations of four points, compute the homography and keep the one which includes the most number of coplanar points. Finally, these coplanar points are excluded and the procedure is repeated for the rest of interest points. Two problems would appear to exist with this approach. First, experiments have shown that a cluster of feature points may be composed of both virtual and physical planes. Secondly, removing the inliers for the computation of subsequent planes would degrade the plane detection, as a point can belong to more than one physical plane at the same time. For example points on the intersection of two walls.

In our previous work [Amintabar and Boufama, 2008], we proposed a plane extraction approach which discards virtual planes by applying the idea that if any non-coplanar point exists in the convex hull [Chan, 1996] of a plane, then the plane is virtual. However, the scheme would fail if a dent region exists in the physical plane. For example, for a scenario in which a thick book is placed on a desk, the approach [Amintabar and Boufama, 2008], would imply that the desk's plane is virtual and would discard it. In this chapter we extend our previous work [Amintabar and Boufama, 2008], so that it avoids virtual planes for more general scenes. Depending on the configuration of the points inside the convex hull of the plane, a few situations happen which include all general scenarios and lead to classifying the plane in four categories: likely virtual, very likely virtual, likely physical and very likely physical.

4.2 Identification of planes

This section presents our scheme to distinguish between virtual and physical planes in plane extraction approach. Although we have extracted and matched the interest points using Zhang[Zhang95] technique, pixel matching itself is not the primary goal of this work and therefore, we assume throughout this chapter that matched interest points are given. The proposed method extracts the planes using our method [Amintabar08], then verifies if the plane is physical and finally merges physical planes to improve the robustness. The steps can be summarized as follows:

1. Find next set of three points (good triangle)
2. Compute homography for the three points
3. Calculate the homography-mapping errors for all the interest points.
4. If the plane includes only small number of points goto 1.

5. If the good triangle is virtual goto step1
6. If the plane is virtual, goto step1
7. Refine the plane using DLT (Direct Linear Transformation).
8. If the new plane became virtual goto Step 1
9. Terminate if either for most of the feature points (e.g. %90) the underlying physical planes are determined or all the good triangles are visited.
10. Merge the patch-planes of Steps (1-9) and re-compute the homography

We first pick three points (good triangle) which meet the criteria explained in [Amintabar and Boufama, 2008] and one additional condition described in this chapter. It states that inside the triangle there has to be at least one feature point. This additional condition 1) helps improving the detection and discard of virtual planes in the subsequent steps of proposed approach, 2) saves CPU time as homography computation is done for fewer triangles.

4.2.1 Three-point method for homography calculation

This step of the proposed approach calculates the homography for a plane defined by a set of three points as opposed to a many existing approaches where four points are picked [Menudet et al., 2008]. Since computation of homography using only three points is not possible, like other methods all the points are used. However, the difference is that in our method we do not need to make any assumption on the coplanarity of a fourth point with the other three. In other words, we can pick any point from the rest of the interest points as the fourth point, form the projective basis and together with the rest of the points accurately compute the homography for the three points.

Let p_1, p_2 and p_3 be 3 points in the left image with corresponding points in the right image being p'_1, p'_2 and p'_3 . These image points are the projections of three space points, call them P_1, P_2 and P_3 (See Figure 4-2).

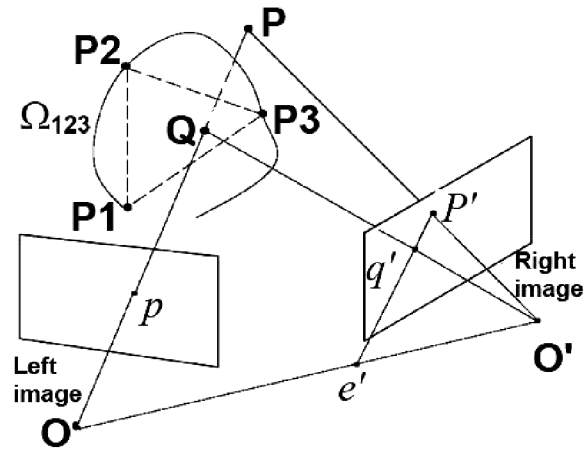


Figure 4-2. Three space points define a plane and therefore a homography

Consider a 3D point P and let the virtual point Q be the intersection of the plane Ω_{123} with the line $\langle OP \rangle$. The projection of Q on the right image, q' , is given by Hp (p is the projection of Q on the left image and H is the plane homography mapping the left image of Ω_{123} to its right image). For each couple (p_i, p'_i) , the relation $Hp_i \simeq p'_i$ yields two independent linear equations in the nine unknown coefficients of H (only eight of which are independent). Thus, P_1, P_2 and P_3 provide six linear equations that can be used to constrain and simplify H . By using two particular coordinate systems in the two images such that:

$$p_1 = (0, 0, 1)^T, p'_1 = (0, 0, 1)^T, p_2 = (1, 0, 0)^T, p'_2 = (1, 0, 0)^T,$$

$$p_3 = (0, 1, 0)^T, p'_3 = (0, 1, 0)^T, p_4 = (1, 1, 1)^T, p'_4 = (1, 1, 1)^T, H \text{ simplifies to}$$

$$H = \begin{pmatrix} \alpha & & \\ & \beta & \\ & & \gamma \end{pmatrix}. \quad (4.1)$$

Note that (p_4, p'_4) was not used to simplify H since the space point P_4 is not assumed to be on Ω_{123} . The above matrix has three parameters, but only two of them are independent. Furthermore, none of these three parameters can be zero for a nonsingular homography. So, in the following we set $\gamma = 1$.

Let $(x, y, t)^T$ and $(x', y', t')^T$ be the known homogeneous coordinates of p and p' , respectively. Then the coordinates of q' are given by

$$Hp \simeq q' \simeq (\alpha x, \beta y, t) \quad (4.2)$$

Let $(e'_x, e'_y, e'_t)^T$ be the unknown homogeneous coordinates of the epipole e' in the right image. It is clear from Figure 4-2 that q' belongs to the line $(e'p')$ and therefore we have

$$(e' \times p') \cdot q' = 0 \quad (4.3)$$

where \times is the cross product and \cdot the scalar product.

By expanding equation (4.3) and using the coordinates of q' given in equation (4.2), we obtain the following equation

$$(e'_t y' - t' e'_y) \alpha x + (t' e'_x - e'_t x') \beta y + (e'_y x' - y' e'_x) t = 0. \quad (4.4)$$

Equation (4.4) has five unknowns; α , β , e'_x , e'_y and e'_t , only four of which are independent. So, in addition to the three couples of matched points used to simplify H , at least four couples of matched points in the two images are necessary to solve for the four

independent unknowns of equation (4.4). A linearization of the above equation can be done by adding one extra parameter. More details on this linearization can be found in [Amintabar and Boufama, 2008] where we have used the parallax idea to calculate the epipolar geometry.

4.2.2 Coplanar points identification (Steps 3-4)

A point p , mapped by a calculated homography H to the point Hp in the other image, is considered coplanar with the initial three points if the Euclidean distance $d(p', Hp)$ is less than a threshold (e.g. 2 pixels), where p' is the match of p . In this step, for each homography, a plane is formed which consists of a set of matched interest point couples $\{(p, p'), \dots\}$ that have passed the mentioned mapping error test. Then only planes containing enough number of interest points are kept [Amintabar and Boufama, 2008]. The surviving planes must be checked to see whether they are virtual or physical planes.

4.2.3 Physical vs. virtual planes distinction (Steps 5-6)

First, the additional condition for a good triangle is used to see if the 3 points are potentially on a physical plane. It states that, if points inside the triangle are not coplanar with the 3 vertices then the resulting plane will not be physical, and no need to do further test on the entire plane (saving CPU time). Next, the distribution of the points inside the plane determine whether the plane is likely virtual, very likely virtual, likely physical or very likely physical.

Scenario 1. There is more than one point inside the convex hull of the plane and all are coplanar with the plane then the plane is very likely physical. (Figure 4-3)

$$\begin{cases} n_{cop} > 1 \\ n_{noncop} = 0 \end{cases} \Rightarrow \text{Very likely physical} \quad (4.5)$$

where n_{cop} denotes the number of coplanar points inside the convex hull of the plane (excluding the vertices of the convex hull). Accordingly n_{noncop} denotes the number of non-coplanar points inside the convex hull.

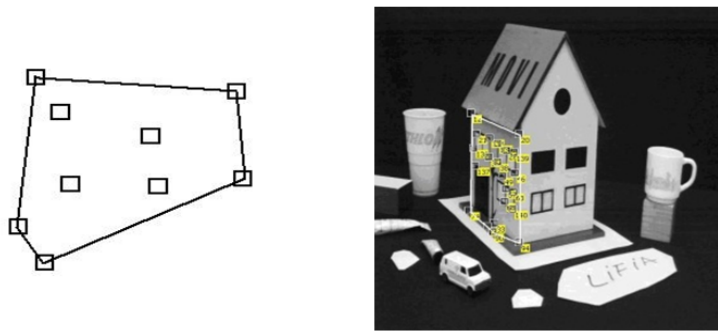


Figure 4-3. Scenario 1, very likely physical

Scenario 2. Inside the convex hull of plane there are more than one coplanar point and also one non-coplanar point. Then that single non-coplanar point might have been there due to computation error or noise, and therefore the plane is still physical but with less probability than that of Scenario 1, it is likely physical (Figure 4-4)

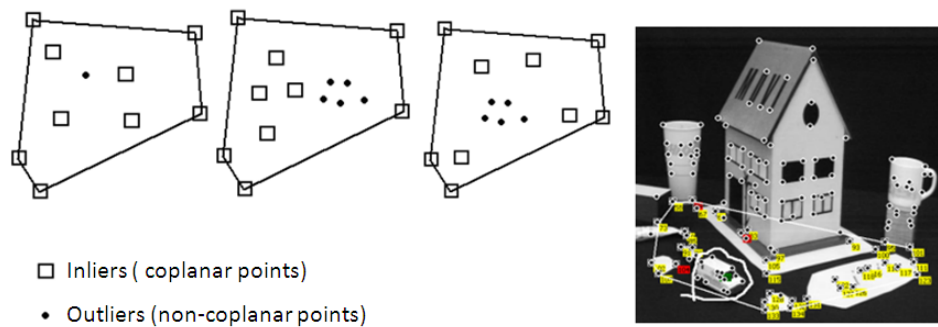


Figure 4-4. Scenario 2, likely physical.

Scenario 3. Inside the convex hull of the plane there are coplanar points and also non-coplanar points, but if the non-coplanar points form a single cluster completely isolated from planar points then the plane is *likely physical*. As shown in Figure 4-4, non-coplanar points are spread only in a cluster and isolated from planar points, then probably there was an object on that plane. The past two scenarios can be expressed as follows:

$$\left. \begin{array}{l} n_{cop} > 1 \\ n_{noncop} = 1 \end{array} \right\} \text{ or } \left. \begin{array}{l} n_{cop} > 1 \\ n_{noncop} > 1 \\ COP \cap cHall(NON) = \emptyset \end{array} \right\} \Rightarrow \text{ Likely physical,} \quad (4.6)$$

where COP denotes the set of coplanar points, NON refers to the set of non-coplanar points and $cHall(NON)$ denotes the convex hull of non-coplanar points inside the plane. To detect this situation, the proposed method works in the following way. First, it computes the convex hull of non-coplanar points which are located inside the plane. Then, if inside the convex hull of non-coplanar points there is no coplanar point then the plane is likely physical, otherwise likely virtual.

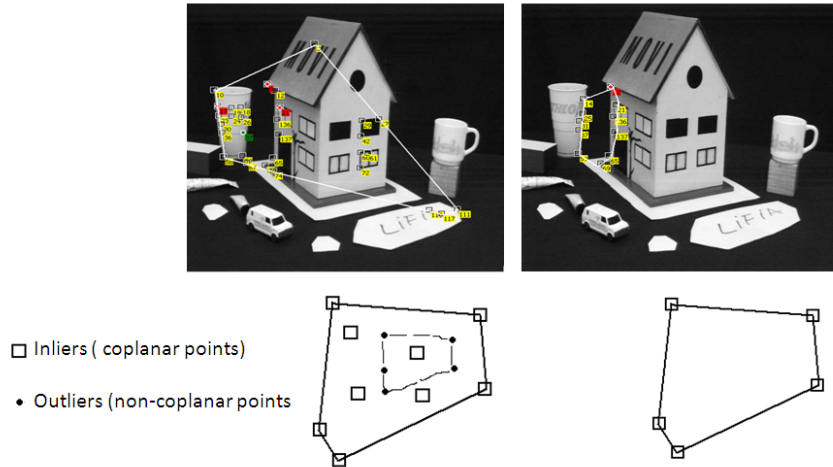


Figure 4-5. Scenario 3, likely virtual

Scenario 4. Both non-coplanar and coplanar points exist inside the convexhull of plane. However, non-coplanar points cannot be isolated from coplanar ones by the convexhull of non-coplanar points. In this situation the plane is likely virtual (Figure 4-5).

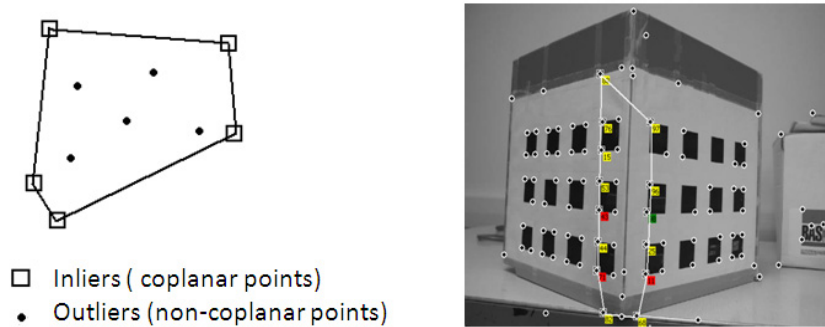


Figure 4-6. Scenario 4, very likely virtual

Scenario 5. There is no point inside the plane, being coplanar or non-coplanar. This situation happens very rare in real images, nonetheless we elaborated on that to cover all the possible scenarios. If this happens, the plane is likely virtual (Figure 4-6). therefore the situations in which the plane is likely virtual are as follows:

$$\left. \begin{array}{l} n_{cop} = 0 \\ n_{noncop} = 0 \end{array} \right\} \text{ or } \left. \begin{array}{l} n_{cop} > 1 \\ n_{noncop} > 1 \\ COP \cap cHall(NON) \neq \emptyset \end{array} \right\} \Rightarrow \text{Likely virtual} \quad (4.7)$$

Scenario 6. If all the points inside the plane are non-coplanar, then the plane is very likely virtual.

$$\left. \begin{array}{l} n_{cop} = 0 \\ n_{noncop} \neq 0 \end{array} \right\} \Rightarrow \text{Very likely virtual} \quad (4.8)$$

It can be noticed that, here as opposed to (5) with more confidence the approach lists the plane as physical. An example of this scenario is shown in Figure 4-6 where the virtual plane formed as a result of intersection of a virtual plane with the 3D object, the cube.

4.2.4 Refining and merging (Steps 7-11)

The aim of this step is to refine the plane homography of steps 1 to 4 to add as many points as possible to the plane. We first re-computed the homography this time by using DLT [Hartley and Zisserman, 2004] method including only the coplanar points from Step 4 in computation. Then the homography-mapping error test is performed for all the neighboring feature points. That is the inliers from Step 3 and the rest of feature points which are close to these points to avoid extrapolation problem [Hartley and Zisserman, 2004]. Then, again we check if this new plane is still physical or otherwise discard it. The final step is to merge the obtained sub-planes. Steps 1-9 are repeated for all the good triangles or until a desired number of interest points are included in planes (e.g. 90% of the interest points). As mentioned earlier, unlike many other methods [He and Chu, 2006, Kawakam et al., 2006], we do not exclude the inliers each time a plane is computed, as a

point can belong to more than one different physical planes at the same time. Removal of inliers at each iteration is necessary for existing approaches in [He06], [Kawakam06], as it makes the plane detection process converge. However, not removing inliers brings the advantage that same plane (or different patches of the same plane) may be computed several times with different sets of three initial points (good triangles). In [Amintabar and Boufama, 2008] we have described a novel method to merge patches of same plane to improve the robustness and accuracy of the algorithm. Finally, once a final plane has been constructed its associated homography is re-calculated with all its interest points.

4.3 Experimental results and performance comparison

In the experiments, the performance of the proposed method is compared with that of a general implementation of existing approaches. To highlight the improvement of the new approach the results are also compared with the output of our previous method [Amintabar and Boufama, 2008]. Finally, the robustness of proposed method was investigated by adding Gaussian noise to the feature points coordinates.

The aim of the experiments is to extract physical planes which consist of at least 15 interest points. Mapping error threshold is kept to 2 pixels, which were obtained through experiment with a range of input pair images. The proposed method is implemented in Borland C++ Developer Studio 2006 on Windows XP operating system. Same platform is also used for a general implementation of existing approaches [Kawakam06] with the following characteristics: 1) the plane with the highest number of inliers is picked first, 2) each time a plane is detected, its inliers are removed for next iteration. Extracted planes by applying three approaches are shown in Figure 4-7 and the intermediate results are presented in Table 4-1.

4.3.1 Performance Comparison

Figure 4-7 compares the result of applying three different plane identification approaches on a pair of stereo images of an outdoor scene with three major planes. Figure 4-7.a shows the interest points on the right pair of stereo images. Figure 4-7.b depicts the three planes identified by a general implementation of [Kawakam et al., 2006]. Figure 4-7.c provides the identified planes using our previous approach [Amintabar and Boufama, 2008] and finally Figure 4-7.d presents the result of our new approach. As seen in Figure 4-7.b, each time a plane is identified the inliers of that plane are removed for next iteration. Therefore, some of the points common with the second plane were also removed. As a result, the right edge of the second plane could not accurately be identified. Same thing happened to the third plane. Results improved in Figure 4-7.c as the merging process was applied and in 1.d even better result appeared. The reason is that in Figure 4-7.d each patch-plane has been refined before undergoing the merging step. In one case we can say that the traditional approaches performed better. That is for the second plane which was identified better than that of Figure 4-7.c and Figure 4-7.d, where that part of plane from the farther building is not included.

Figure 4-7.e-h present a more complex scene with a texture-less background. Figure 4-7.f shows the results of applying a general implementation of existing approaches in which a plane with highest number of interest points is picked first and its inliers are removed for the identification of subsequent planes. As a result, after the identification of first plane next plane with highest number of inliers turned out to be virtual. So, the algorithm fails to identify more physical planes from this scene.



Figure 4-7. Comparing results

Figure 4-7 compares the results as follows. a,e) interest points on second image. b,f) planes identified using [He and Chu, 2006], [Kawakam et al., 2006] on left pair. c,g) using our previous method [Amintabar and Boufama, 2008] d,h) planes identified by the proposed approach. It presents the result of applying our previous method [Amintabar & Boufama, 2008] in which a plane with even one non-coplanar points inside its convex hull is considered virtual. Due to consideration of distribution of points, results are improved in our new approach. As Figure 4-7.h shows, the plane of the left side of the house also improved. That's because of the Step 7 which was added to the algorithm.

4.3.2 Noise Tolerance

We add Gaussian random noise with zero mean ($\mu = 0$) and standard deviation (σ) 0.6, 1.0 (pixel) to each coordinate of the points independently to test the robustness of proposed approach for noisy inputs. The results are presented in the last four columns of Table 4-1 for the two scenarios, the building and the house. Table 4-1 also includes the intermediate results for all the three approaches. The number of interest points, number of good triangles whose homography estimated, number of patch planes which are merged to form final planes, number of final planes detected and the number of interest points which lie on the detected physical planes.

It can be implied that the algorithm is robust to small noise in point coordinates. As seen, for $\sigma = 0.6$ small changes happen to the outputs. Although fewer numbers of points are listed with physical planes, the detected planes are almost the same. The results change and fewer physical planes can be detected when more noise is added to the points coordinates ($\sigma = 1$ pixel and above). Our experiments also showed that the proposed approach degrades in complex scenes with too many interconnected planes. In this situation, only few of the physical planes can be detected and the algorithm rejects most of the detected planes as it thinks they are virtual planes.

Table 4-1. Intermediate results of applying the plane detection approaches

	Method	Building	House	Building (+ noise)		House (+ noise)	
Image size	used	569×461	511×514	$\sigma = 0.6$	$\sigma = 1.0$	$\sigma = 0.6$	$\sigma = 1.0$
Feature points		143	141	143	143	141	141
Good triangles	Previous	2001	7698				
	New	1329	2453	1407	1598	2805	3780
Patch planes	Previous	92	54				
	New	61	32	53	39	28	16
Physical planes extracted	Existing	3	1				
	Previous	3	4				
	New	3	4	3	2	4	3
Feature points included by physical planes	Existing	113	32				
	Previous	96	103				
	New	104	119	95	66	108	91

A general observation on the results in Table 4-1 and Figure 4-7 implies that the proposed plane identification approach outperforms the other two methods. First, no virtual plane detected wrongly instead of physical planes. Second, the planes' boundaries have been identified better. Also the algorithm can tolerate noisy inputs for pixel errors up to 0.6 pixels with no considerable change in its performance.

4.4 Conclusion

A general solution is needed for the detection of physical planes in a scene using uncalibrated images for applications ranging from scene modeling to robot navigation. Unfortunately, in addition to real planes, non-desirable virtual planes are also extracted from the images, making any further processing more complex and more error-prone. We have presented a new scheme, in conjunction with homography-based plane detection

methods, for the identification of virtual plane so that they can be removed. The proposed approach uses non-coplanar points inside an extracted plane to decide whether the plane is physical or virtual. We have used our own method to estimate plane-homographies, where only three points are used for the calculation of each homography without the need for a fourth coplanar point. As a consequence, we were able to compute the homographies for all potential physical planes whose three points meet our criteria. After the removal of virtual planes, the remaining planes must undergo a merging process in order to remove redundancy and to improve the robustness of the results.

The proposed approach does not give absolute answer to whether a plane is virtual or physical as this is not always possible. However, it does classify a plane into one out of four categories, i.e., likely virtual, very likely virtual, likely physical or very likely physical. The results of our experiments have shown that the proposed method was able to identify physical planes even for complex scenes, where the existing approaches may fail by detecting virtual planes instead. It has been also shown that by not removing inliers, each time a plane is detected, we were able to obtain a better identification for plane boundaries. Accurate identification of plane scenes is useful for our future work where edge points will be used to extract the actual facets of objects in the scene. Once such facets are identified, they can be reconstructed either in the 3D projective space or in the 3D Euclidean space (if some kind of camera calibration is available) to allow for some 3D modeling of scene.

5 Parallel Planes Identification

This chapter proposes a new method for identifying parallel planes in a scene from three or more uncalibrated images. By using the fact that parallel planes intersect at infinity, we are able to devise a linear relationship between the inter-image homographies of the parallel planes and the plane at infinity. This relationship is combined with the so-called modulus constraint for identifying pairs of parallel planes solely from point correspondences.

5.1 Problem Statement and Related Work

As mentioned in earlier chapters, man-made environments are abundant with planes, some of which are parallel in both indoor and outdoor scenes, e.g. walls of hallways, buildings along a street. Consequently, the detection of parallel planes can highly contribute to the understanding of 3D structures of such scenes. In many scenarios, there is not much information available about the used camera. However, some geometrical knowledge of the scene such as parallelism of lines [Criminisi et al., 2000],[Kawasaki and Furukawa, 2009], [Xu et al., 2006], [Tournaire and Papanastasiou, 2009],[Gasparini and Sturm, 2009], perpendicularity of planes [Heyden and Huynh, 2002] or plane parallelism [Tebaldini et al., 2008] are available. Khan and Shah [Khan and Shah, 2009] use multiple imaginary planes parallel to a reference plane to tackle occlusion in tracking people in video surveillance applications. Cui and Zhu [Cui and Zhu, 2009] propose a calibration method for optical triangular profilometry in which the test planes need to be parallel. Common to all these works are the uncalibrated cameras and the a priori knowledge requirement about parallel planes. Tebaldini et al. [Tebaldini et al., 2008]

utilize parallel planes in the scene to retrieve physically valid views for registration applications. In all of these approaches, parallel planes are either modeled mathematically or, if present in the scene, are assumed to be already known. Hence, to our knowledge none elaborated on the identification of such scene geometry, the plane parallelism.

While line parallelism has been thoroughly investigated in the literature through the detection of vanishing points (see, for instance [Seo et al., 2006] and the references therein), we have found no work about the identification of the scene's parallel planes from images. When cameras are calibrated, a 3D reconstruction of the scene makes it possible to identify such planes. However, when cameras are not calibrated, the scene can be reconstructed only up to a projective ambiguity that does not preserve parallelism.

In this work, we investigate the automatic identification of a pair of parallel planes from images of a scene captured by a moving camera with unknown but constant intrinsic parameters. In particular, we show that there exists a linear relationship between the coordinates of parallel planes and the ones of the plane at infinity. This relationship remains true in the projective space and extends to the inter-image homographies of these planes. Consequently, we propose an image-based parallel planes identification method that combines the latter relationship and the so-called modulus constraint [Pollefeys and Gool, 1999] on the homography matrix of the plane at infinity. Our method uses only matched pixels across the images and neither calibration nor 3D reconstruction are needed. In this chapter we propose a novel approach to detect parallel planes from a sequence of images taken by a constant parameter camera. The camera is observing a scene in which a pair or more of parallel planes exist. Our method identifies those planes

even with the presence of noise. The proposed method is based on the moduli constraint of infinite homographies and is detailed below.

5.2 Background

Let the (3×4) matrices $M_i, (i = 1..m)$, represent the projective projection matrices for a sequence of images taken by a moving camera. Using existing methods all m projection matrices can be calculated only from point matching between views [Rothwell et al., 1995]. Assuming the scene is embedded in the a 3D projective space \mathbb{P}^3 and the reference frame is attached to the first camera, M_1 simplifies to $M_1 = [I \mid 0]$ with 0 being the null 3-vector and I is the (3×3) identity matrix. All other projection matrices can be calculated up to the same projective ambiguity [Hartley and Zisserman, 2004] in which case the set of matrices is referred to as a *consistent set*. That is, the projective projection matrices are defined as

$$M_1 = [I \mid 0], M_2 = [A_2 \mid a_2] \dots M_m = [A_m \mid a_m], \quad (5.1)$$

where $A(3 \times 3)$ and $a(3 \times 1)$ are a matrix and a vector with real number elements.

Let $\Pi \simeq (v_{\Pi}^T \ 1)^T$ be the coordinate vector of a scene plane $\Pi \in \mathbb{P}^3$ not containing the origin of the reference frame (If the plane contains one of the camera centers then the induced homography is degenerate). Let A $P \in \mathbb{P}^3$ be a 3D scene point which lies on the plane Π . P is projected on the first view as $p_1 \simeq [I \mid 0]P$. This implies that P could be any point with components

$$P = (p_1, \lambda)^T,$$

where λ is a scalar. At the same time P is on the plane Π , so it satisfies:

$$\Pi^T P = 0.$$

Therefore

$$(v^T, 1)(p_1, \lambda)^T = 0 \Rightarrow \lambda = -v^T p_1 \Rightarrow P \simeq (p_1, -v^T p_1)^T.$$

Now P projects on the i^{th} view as

$$\begin{aligned} p_i &\simeq M_i P \\ &\simeq [A_i \mid a_i](p_1, -v_{\Pi}^T p_1)^T \\ &\simeq (A_i - a_i v_{\Pi}^T) p_1 \end{aligned}$$

Then

$$p_i \simeq [I \mid -v_{\Pi}] M_i^T p_1 \quad i = 2 \dots m, \quad (5.2)$$

Note that in the equation (5.2) the only unknown is the 3-vector v_{Π} . Recall that P is only one of the multiple inliers of plane Π , and p_1, \dots, p_m are its projections on the m views. Using the projections of all such points we can compute the 3-vector linearly using, for example, SVD.

Similarly, Let the 3-vector v_{Π} of a plane Π is given. The induced inter-image homography of Π between the reference camera and the camera i is given by

$$H_{\Pi i}^T \simeq [I \mid -v_{\Pi}] M_i^T \quad (5.3)$$

where $H_{\Pi i}^T$ is a (3×3) homography matrix which relates the projections p_1 and p_i of any 3D point P lying Π onto images of reference view and the i^{th} view respectively.

$$p_i \simeq H_{\Pi i} p_1.$$

Combining the equations (5.3) and the above relation provides us a set of linear equations to solve for the 3-vector v_{Π} :

$$p_i \simeq [I \mid -v_{\Pi}] M_i^T p_1. \quad (5.4)$$

5.3 Constraints due to parallel planes

The plane at infinity which we denote by ∞ is of particular interest to our work. In particular, when the intrinsic parameters of the camera are kept unchanged in two images, the eigenvalues of the inter-image homography of the plane at infinity have all equal moduli [Pollefeys and Gool, 1999].

We show that there exist a linear relationship between parallel planes and the plane at infinity in \mathbb{P}^3 . This relationship extends to the planes' inter-image homographies making it possible to identify the plane at infinity when the modulus constraint is included.

Consider a 3D scene consisting of two distinct scene planes Π_E and Φ_E which, for starters, are assumed to be embedded in the Euclidean space \mathbb{E}^3 . $\Pi_E \simeq (n_{\Pi}^T \ 1)^T$ and $\Phi_E \simeq (n_{\Phi}^T \ 1)^T$ are parallel to each other if and only if their respective normal directions n_{Π} and n_{Φ} satisfy

$$n_{\Pi} \simeq n_{\Phi}. \quad (5.5)$$

When the scene is observed by two or more uncalibrated cameras, the coordinates of the scene planes can be obtained only up to a projective ambiguity v_{Π} and v_{Φ} . Thus the reconstructed scene is embedded in the projective space \mathbb{P}^3 in which parallelism is not preserved. Therefore the equation (5.5) is not valid for projective representation of

parallel planes Π_E and Φ_E , which we denote with Π and Φ respectively. However, even in projective space, since Π and Φ are parallel, they meet at infinity. Therefore planes Π , Φ and the plane at infinity which we denote with ∞ meet in a line (Figure 5-1).

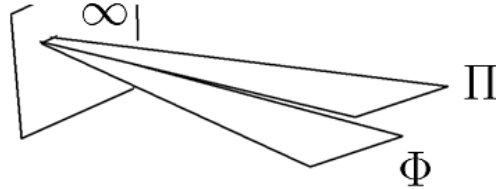


Figure 5-1. Two parallel planes meet in a line on plane at infinity

The most we can get from uncalibrated images is a projective reconstruction of the scene with unknown projective mapping G from the actual Euclidean reconstruction that we expect to get. When the scene points undergo an unknown projective mapping G , planes are mapped to their new locations by its dual G^{-T} . For a 3D point $P_E \in \mathbb{E}^3$ on plane Π_E and its transformed one, the 3D point $P \in \mathbb{P}^3$ on the plane Π

$$\begin{cases} P^T \Pi_E = 0 \\ P = GX_E \end{cases} \Rightarrow (G^{-1}P)^T \Pi_E = 0 \Rightarrow P^T (G^{-T} \Pi_E) = 0,$$

and thus

$$\Pi = G^{-T} \Pi_E. \quad (5.6)$$

Hence, based on equation (5.6), points at the intersection of planes Π_E , Φ_E and ∞_E are mapped to the intersection of the new locations $\Pi \simeq (n_{\Pi}^T \ 1)^T G^{-T}$, $\Phi \simeq (n_{\Phi}^T \ 1)^T G^{-T}$ and $\infty \simeq (0^T \ 1)^T G^{-T}$ of these planes. Thus ∞ , Π and Φ also intersect in a line, but this time in \mathbb{P}^3 , i.e., their coordinate vectors are linearly dependent in this space:

$$\infty \simeq \alpha\Pi + \Phi,$$

where α is a scalar.

Remark. Let $\Pi \simeq (v_{\Pi}^T \ 1)^T$, $\Phi \simeq (v_{\Phi}^T \ 1)^T$ and $\infty \simeq (v_{\infty}^T \ 1)^T$ be the coordinate vectors of our planes in \mathbb{P}^3 (none of Π and Φ and (obviously) ∞ contains the origin), then we have

$$v_{\infty} \simeq \alpha v_{\Pi} + v_{\Phi} \quad \left| \begin{array}{l} \alpha \neq -1 \\ \alpha \neq 0 \end{array} \right., \quad (5.7)$$

where α is a scalar considering the following principals.

- If $\alpha = -1$, then ∞ would contain the origin, which is not possible.
- If $\alpha = 0$, then Φ would coincides with the plane at infinity ∞ which is against the assumption that Φ is a scene plane.

From (5.3), (5.7) and a set of consistent projective camera matrices $P_{i(3 \times 4)}$, the following relationship

$$H_{\infty i} \simeq \alpha H_{\Pi i} + H_{\Phi i} \quad \left| \begin{array}{l} \alpha \neq -1 \\ \alpha \neq 0 \end{array} \right. \quad (5.8)$$

holds for the same value of the scalar α throughout the sequence provided $H_{\Pi i}$ and $H_{\Phi i}$ are calculated as follows.

$$\begin{cases} H_{\Pi i}^T \simeq [I \mid -v_{\Pi}]P_i^T \\ H_{\Phi i}^T \simeq [I \mid -v_{\Phi}]P_i^T \end{cases} \quad (5.9)$$

Homography $H_{\Pi i}$ is induced from plane Π and relates projections of plane Π onto the reference image and i^{th} image.

Remark. To make sure the same α verifies equation (5.8) for all images, $H_{\Pi i}$ and $H_{\Phi i}$ must be obtained from projection matrices in which case v_{Π} and v_{Φ} are calculated through point correspondences. We will explain how it is possible only through point matching.

5.4 Imposing Modulus Constraint

The modulus constraint is a constraint on the projective location of the plane at infinity ∞ in \mathbb{P}^3 in the case of constant intrinsic camera parameters. When the intrinsic parameters of the moving camera are constant, the eigenvalues of $H_{\infty i}$ have equal moduli [Pollefeys and Gool, 1999]. For each camera i , except the reference one, the characteristic equation of $H_{\infty i}$ is given by

$$\det(\alpha H_{\Pi i} + H_{\Phi i} - \lambda I) = 0,$$

which expands to

$$a_i(\alpha)\lambda^3 + b_i(\alpha)\lambda^2 + c_i(\alpha)\lambda + d_i(\alpha) = 0 \quad (5.10)$$

where the coefficients a, b, c and d are in terms of the real unknown α (considering $\alpha \neq -1$, $\alpha \neq 0$ required by equation (5.8)). A necessary condition for the eigenvalues of $H_{\infty i}$ to have equal moduli [Pollefeys and Gool, 1999] can be expressed as:

$$\left\{ \begin{array}{l} a_i(\alpha)c_i^3(\alpha) - b_i(\alpha)d_i^3(\alpha) = 0 \\ \alpha \text{ has to be a real number} \\ \alpha \neq -1 \\ \alpha \neq 0 \end{array} \right. \quad (5.11)$$

Each image i , $i \neq 1$, (except the first one chosen as reference), provides a sextic polynomial equation (5.11) in the unknown α , the solutions of which are the eigenvalues of the associated 6×6 companion matrix. The sought α is at the intersection of the solution sets of (5.11) obtained over all pairs of images.

5.5 Proposed Parallel Planes Identification

Briefly, we first identify major planes of the scene using our method [Amintabar and Boufama, 2008]. We then examine every pair of identified planes to see if they are parallel. The examination is based on the fact that if two planes are parallel, there exists a linear relationship between their homographies and the homography of plane at infinity. Therefore, for every picked pair of planes, we try to find such linear relationship. If that relationship exists then the two planes are our candidates. In order to do that calculation, we use the so called modulus constraints [Pollefeys and Gool, 1999] which is valid only if a camera with constant parameters was used to capture all the images. In such case, the main property of an infinite homography is that all its three moduli are equal (Section 3.3.4). Therefore, if a linear combination of any two planes' homographies has equal moduli, the two planes are potentially parallel. This necessary condition may be met by many non-parallel planes in the first two views of a typical planar scene. However, as we add more views at the end only the pair of parallel planes meet this condition.

In the following, we assume a consistent set of projection matrices has been calculated [Hartley and Zisserman, 2004] and image points belonging to each plane have been identified as projections of co-planar 3D points. The latter assumption is possible using the method in [Amintabar and Boufama, 2009] which additionally distinguishes

physical from virtual planes. Our method seeks, for each pair of planes Π and Φ , the value α , in (5.11), that minimizes the deviation from unity of the moduli of the eigenvalues of all the normalized inter-image homographies of the plane $\alpha\Pi + \Phi$. The pair of planes with the smallest deviation is identified as parallel.

Deviation

Given a set S of planes, a pair of planes $(\Pi, \Phi) \in S \times S, \Pi \not\cong \Phi$, is identified as parallel, if, for some α satisfying (5.11), the deviation from 1 of the moduli of the eigenvalues calculated over all inter-image homographies,

$$e(\Pi, \Phi, \alpha, l) = \sum_{j>i, (i,j) \in imgs \times imgs} \sum_{k=1,2,3} (\|\lambda_{k(i,j)}\| - 1)^2 \quad (5.12)$$

is minimal. $\|\lambda\|$ denotes 's modulus and $\lambda_{k(i,j)}$ ($k = 1, 2, 3$) are the eigenvalues of

$$(\alpha H_{\Pi j} + H_{\Phi j})(\alpha H_{\Pi i} + H_{\Phi i})^{-1} a$$

normalized such that its determinant is 1. Finally, $imgs$ is the set of all available images.

Choosing α

In the presence of pixel noise, finding a common value of α satisfying (5.11) often fails even if the considered planes are parallel. This could be either due to the fact that α is found with a different accuracy for each image or, worse, it turned into a complex solution for some of the images. Therefore, in the absence of knowledge about the two planes and when failing to locate a common α , no conclusion can really be drawn on whether the planes are parallel or not. Therefore, instead of seeking a common α , one can solve (5.11) for a number of images over a subset $\mathcal{I} \subseteq \{imgs\}$ of the set $\{imgs\}$ (set of

all images). While a large \mathcal{I} increases the chances of getting the right α , it significantly slows the algorithm down. For "short" image sequences, $\mathcal{I} = \{imgs\}$ is the safest choice while for longer ones, the size of \mathcal{I} should be determined experimentally.

Identifying parallel planes

In order to find a pair of planes that minimizes the deviation (5.12) we maintain a $|S| \times |S|$ (where $|S|$ is the number of planes at hand) matrix D of minimal deviations. The deviation $D[\Pi, \Phi]$ associated to the two planes Π and Φ is updated every time a smaller deviation, for some value of α in the solution set obtained from the images in \mathcal{I} , is found. The pair of planes for which $D[\Pi, \Phi]$ is minimal is identified as parallel. If more than one pair of parallel planes is sought, then it is identified by the next smallest deviation in L .

Sequence length

Given a sequence of three images taken by a constant intrinsic camera parameters it is possible to locate the plane at infinity in projective space. This involves for solving for the three parameters of plane at infinity $\infty = (p_1, p_2, p_3, 1)^T$. Geometrically this is the problem of intersecting three quartic surfaces in projective space, so one should expect to get 64 solutions ($64 = 4 \times 4 \times 4$) [Pollefeys et al., 1996]. However, in [Schaffalitzky, 2000] it is shown that for a triplet of images, the number of planes whose inter-images homographies satisfy the modulus constraint does not exceed 21(including complex planes) [Schaffalitzky, 2000]. Any two scene planes intersecting one of those 21 planes in the same line could be mistakenly identified as parallel. Each additional image reduces, in general, the number of such planes until the plane at infinity is the only one left satisfying the modulus constraint for all inter-image homographies.

5.6 Experiments

The goal of these experiments, using both simulated and real data, was to identify a pair of parallel planes from images using our proposed method.

Experiments using simulated data: We have conducted two kinds of experiments with simulated data each with a varying number of images and noise added to pixel coordinates. In the first kind of experiments, the sought pair of planes was perfectly parallel. In the second one, noise was added to the angle between two initially parallel planes. Our method was tested using randomly generated scenes each consisting of 10 planes two of which were either perfectly parallel or close-to-parallel. Each scene was created with its own set of randomly generated sequence of cameras. 100 such scenes/cameras were created for each number of cameras in the range 3 to 10. The identification of the pair of parallel (or close-to-parallel) planes was then carried out for varying levels of pixel noise.

Scenes and cameras: Each of the 10 scene planes consisted of 50 randomly generated scene points scattered in a disc of radius 1. The first plane was placed on the $Z = 0$ plane of the reference frame with its points centered at the origin. The second plane was created parallel to the first one and located at a mean distance of 0.5 units from it with a 0.15 standard deviation. For experiments with close-to-parallel planes, we have considered two scenarios where the second plane was rotated by a zero-mean randomly generated angle. The standard deviation of the angle was 0.5 degrees (about each axis) for one scenario and 1 degree for the other. In all the experiments, the center of the disc on each of the remaining 8 planes was placed at random locations, drawn from a uniform distribution on the unit interval, and orientations obtained by the Cayley transform of

randomly generated vectors (uniform distribution). The cameras were generated at a mean distance of 3 units and a 0.3 standard deviation from the origin of the scene's reference frame. The optical axis of each camera was first oriented so it contains the origin before the camera was rotated by a zero-mean randomly generated angle with 3 degrees of standard deviation about each axis. All cameras have been created with square pixels, no-skew, 800 pixels focal length and the principal point centered in a 512×512 image. Zero-mean Gaussian noise with standard deviation from 0 to 2 pixels (with 0.5 step) was added to pixel coordinates.

The results. Tables 5-1, 5-2 and 5-3 summarize the results of our simulations with perfectly parallel (Table 5-1) and close-to-parallel (Tables 5-2 and 5-3) planes for varying sequence lengths and levels of pixel noise. For a given number of images and noise, the corresponding value in each table represents the number of scenes (out of the 100 generated) for which the sought pair of parallel (or close-to-parallel) planes was successfully identified. In Table 5-2 and 3, the results were obtained after one of the parallel planes was rotated about each axis by an angle drawn from a zero-mean Gaussian distribution of 0.5 degrees (resp. 1 degree) standard deviation.

The resulting planes made an angle of up to 2 degrees (resp. 4 degrees). The reported results in all tables confirm that chances of identifying a pair of parallel planes increase with the number of images and decrease with localization errors in the image. While the success rate of the method is quasi-perfect even with high levels of noise for longer sequences, it is also very good with short ones (86% on perfectly parallel planes with only 3 images and 2 pixels of noise). Also, the further our planes are from perfectly parallel, the more the method fails to identify them as such. However, even with angles of

2 and 4 degrees between the planes and high levels of noise, the success rate is very satisfactory.

Table 5-1. Perfectly parallel planes.

Pixel noise	Number of images							
	3	4	5	6	7	8	9	10
0.00	100	100	100	100	100	100	100	100
0.50	95	99	100	100	100	99	99	100
1.00	92	97	99	100	98	100	100	100
1.50	86	96	97	98	97	98	99	99
2.00	86	97	91	97	98	100	98	99

Table 5-2. Planes with 2 degrees angle.

Pixel noise	Number of images							
	3	4	5	6	7	8	9	10
0.00	87	95	96	98	96	96	98	98
0.50	87	96	97	98	95	95	97	97
1.00	82	93	94	98	96	95	95	97
1.50	77	92	93	97	95	94	96	98
2.00	78	91	96	96	93	93	93	96

Table 5-3. Planes with 4 degrees angle.

Pixel noise	Number of images							
	3	4	5	6	7	8	9	10
0.00	80	88	90	93	89	90	89	90
0.50	77	87	90	93	88	87	90	93
1.00	71	88	89	90	85	89	90	88
1.50	72	84	88	9	88	88	88	89

Experiments using real images. We provide here five examples, Figure 5-2 to Figure 5-6.

Figures 5-2 and 5-3 each with a contrived scene consisting of four letter size white sheets

of paper with black patterns. Only the planes labeled 1 and 3 (resp. 3, 4 and 5) are parallel in Figure 5-2 (resp. Figure 5-3) example. In each example, four 640×480 images were captured by a low-end digital camera (Sony Cyber-shot DSC-S930). Figures 5-2 and 5-3 show 3 out of the 4 images used in each example. The example in Figure 5-2 (resp. Figure 5-3) aimed at identifying one pair (resp. three pairs) of parallel planes. Only 12 (resp. 16) points were detected (left image) and matched across the sequence. All four planes have been successfully identified using [Amintabar and Boufama, 2009] (middle image). Planes labeled 1 and 3 (resp. (3,4), (3,5) and (4,5)) were rightfully identified as parallel (right image).

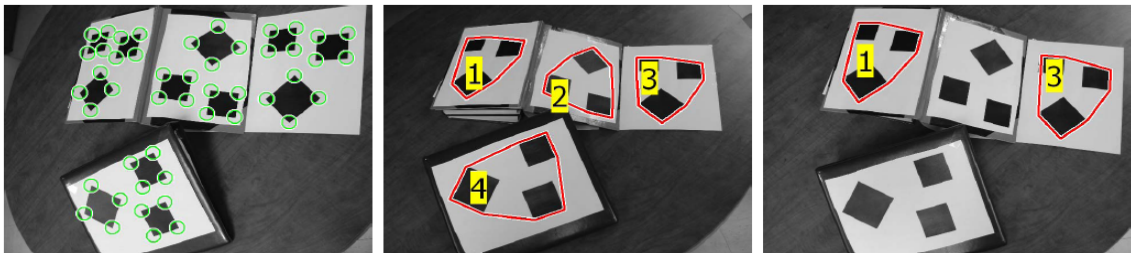


Figure 5-2. Planes 1, 3 and 4 are parallel

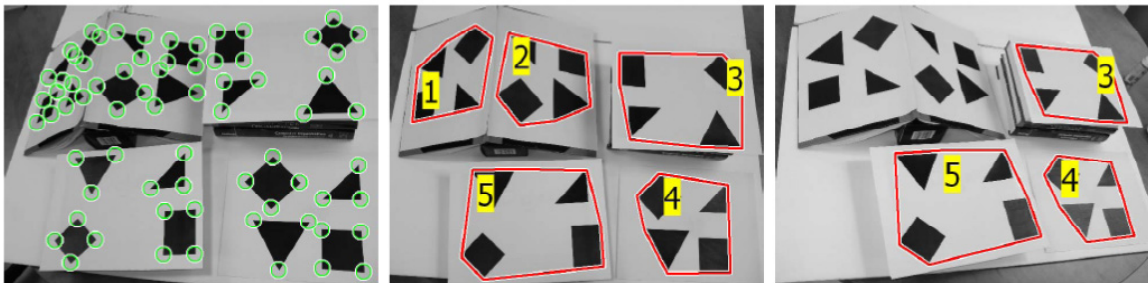


Figure 5-3. Planes 3,4 and 5 are parallel

The indoor scene in Figure 5-4 is an office desk area of which 32 points (left image) lying on four distinct planes have been successfully detected and matched across four images. All four planes in Figure 5-4 have been identified (middle image) and the planes labeled 1 and 3 have been found parallel to each other (right image).

The outdoor scene in Figure 5-5, the Royal Victoria College in Montreal. In this example, 33 points (left image) lying on three planes (middle image) have been matched across three images only. Yet our method has identified the planes labeled 2 and 3 to be parallel (right image).

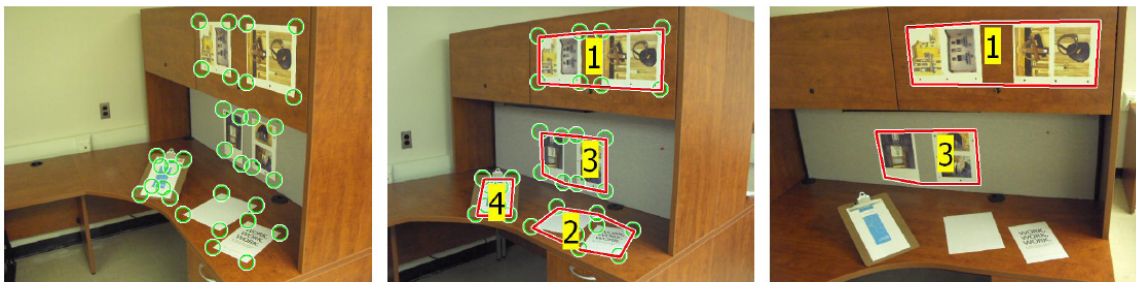


Figure 5-4. Indoor scene, planes 1 and 3 are detected as parallel

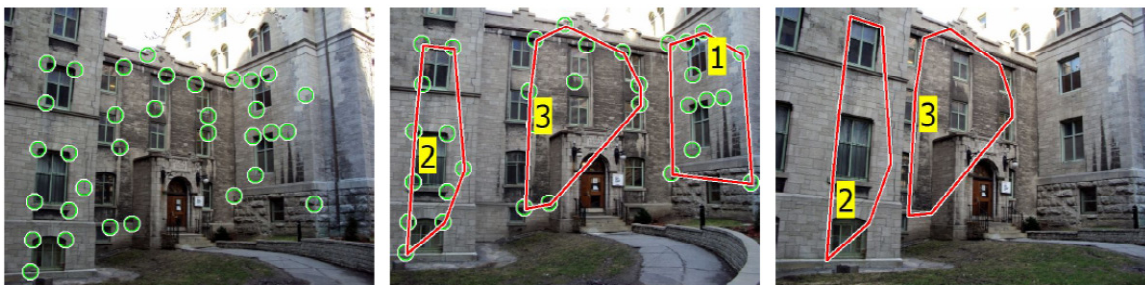


Figure 5-5. Royal Victoria College (outdoor).

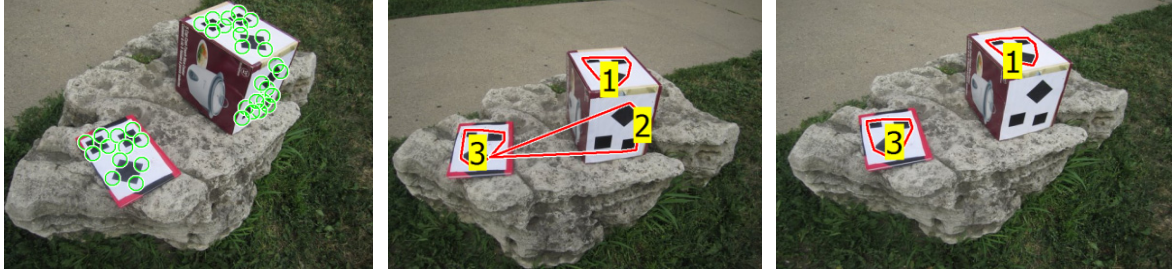


Figure 5-6. A rock, 3 views out of 4.

Another outdoor scene is presented in Figure 5-6, the University of Windsor campus, in front of Erie Hall. In this example, 36 points (left image) lying on three planes (middle image) have been matched across four images. The pictures were taken using Canon powershot SD780IS camera was used. Our method has identified the planes labeled 1 and 3 to be parallel (right image). It can be seen that the plane was misled by the algorithm [Amintabar and Boufama, 2009] to include one spot of plane 3 as its inliers.

5.7 Conclusion

In this chapter, we have proposed a novel method for parallel planes identification from point correspondences across three or more images. Our method neither requires the camera to be calibration nor a 3D reconstruction to be calculated. It relies on prior detection of planes and point correspondences only. The results of the numerous experiments we have conducted show that, the proposed method achieves its goal with a very high success rate even for high levels of noise on both image coordinates and angle between planes.

6 Combining Scene and autocalibration constraints

Autocalibration approaches are known to be challenging due to their nonlinear equations and sensitivity to noise. In autocalibration, the goal is to locate the plane at infinity and the camera intrinsic parameters, without using any calibration pattern. A classical approach to autocalibration, in the case of a single moving camera with constant but unknown parameters, is based on solving the Kruppa equations [Maybank and Faugeras, 1992], which is known to be sensitive to noise [Luong and Faugeras, 1997]. One way to tackle the noise in autocalibration approaches is to incorporate scene constraints [Zisserman et al., 1998]. In this chapter we propose affine autocalibration methods which incorporates planar scene constraints and are robust to noise.

Aside from the problem of sensitivity to noise, the applicability of camera autocalibration approaches highly depends on the issue of initialization. Since a nonlinear optimization is always required [Fusiello, 2001], convergence may be reached only if the cost function is initialized within a proper basin of attraction. A number of techniques are proposed for linear initialization of nonlinear calibration methods. These techniques are often based on assuming that principal points are located on the center of images [Hartley and Zisserman, 2004]. Such assumption can lead to a linear solution which is used as a starting point for the nonlinear optimizations.

To address this issue, we propose a metric autocalibration method which does not require an accurate estimation of starting points, and converges even with a very inaccurate ones from random guesses. Thus, it does not need any assumption about the

principal points being in the center of images, which is usually used for estimating a linear starting point for nonlinear camera calibrations. It uses perpendicularity and parallelism of planes in the scene to impose constraints on the values of camera parameters. In particular, the proposed method combines the two constraints into a single nonlinear framework. The experimental results show that our approach successfully converges even with the presence of two pixel noise.

The proposed calibration approaches are applicable to urban area scenes which are rich in planar structures, e.g., buildings, streets and certain kinds of manufactured objects such as furniture. In particular, those urban scenes are abundant with parallel and perpendicular planes.

6.1 Planar scene constraints in affine calibration

Affine calibration (or affine reconstruction) is equivalent to locating the plane at infinity in projective space. This is a challenging task as it is very sensitive to noise [Hartley et al., 2002]. Most existing affine calibration methods are based on scene constraints or special motions [Manning and Dyer, 2001]. These motions are generally translations [Ruf et al., 1998], [Hammarstedt et al., 2006] as these make it possible to extract vanishing features from images. In general, the affine calibration of cameras is possible as soon as three vanishing points (in addition to the epipoles) can be extracted and matched across two or more images. Alternatively, one vanishing line and one vanishing point can also be used [Hartley and Zisserman, 2004]. It has already been shown in [Pollefeys and Gool, 1999], [Sturm and Quan, 1995] that the affine calibration of a moving camera with constant intrinsic parameters is possible using only two images as soon as two vanishing points can be detected and matched across the views. In the latter case, a constraint on the

moduli of the eigenvalues of the homography of the plane at infinity compensates for the missing point.

Vanishing points cannot always be extracted from images as they require the presence of parallel lines in the observed scene [van den Heuvel, 1999]. When these special points can be extracted, they often lack accuracy as small errors on the localization of (parallel) lines in images often result in large localization errors on the vanishing points. Best results are obtained when relying on a large number of parallel lines [Seo et al., 2006]. Most scenes, however, contain only a limited number, if any, of parallel lines. A vanishing line can be obtained from a pair of vanishing points in which case such line is poorly localized. Alternatively, a vanishing line can be obtained from the intersection of two parallel planes. The projection of the "intersection" line of these planes cannot be calculated directly in the images. Instead, it is obtained by reconstructing, through triangulation, the parallel planes in a 3D projective space, and their intersection line is back-projected onto the images [Sturm and Quan, 1995]. In addition to localization errors on points matched across images, the coordinates of the projected line may very well be further deteriorated by triangulation errors.

Another way to position the plane at infinity is to enforce the modulus constraint which was discussed in Section 3.3.4 of this dissertation. In [Pollefeys et al., 1996] one of the first approaches is proposed to compute all the 64 solutions to modulus constraints. However, out of these solutions only 21 could physically be valid [Schaffalitzky, 2000]. The author's approach is to imposing cheirality constraints on the points and camera centers. Cheirality constraints is imposed to bound a rectangular region inside which the 3-vector v_∞ representing the plane at infinity $\infty = (v_\infty^T, 1)^T$ must lie. Then a brute force

search is undertaken to locate the elusive plane at infinity in \mathbb{P}^3 within this region [Hartley et al., 2002], [Chandraker, 2009].

To address the problem of accurate identification of the plane at infinity, we propose a method that uses parallel planes in the scene. This is justified by the fact that man-made environments are abundant with geometric structures where parallel planes are very common (walls of hallways, buildings on both sides of streets, . . . etc.). In this work, we investigate the use of parallel planes for the affine calibration of cameras. In particular, we propose two methods that only use the inter-image homographies of parallel planes in the scene and do not require 3D reconstruction. Both methods require neither the extraction nor the detection of vanishing features (neither points nor lines) in images.

We only rely on point correspondences of the projections of scene points across images with the sole assumption that parallel planes are identified. Our first method deals with the case where at least two images of only one pair of parallel planes are available and the camera parameters are constant. The second method makes no assumption whatsoever about the cameras as these can be altogether different. The latter method is linear and assumes the presence of two or more pairs of parallel planes (with different directions) in the scene. In the third method we use a pair of parallel planes and a pair of perpendicular ones.

6.2 Planar scene constraints in metric calibration

In metric autocalibration approaches (or simply autocalibration) one more step needs to be taken compared with affine calibrations. As well as locating the plane at infinity, the camera intrinsic parameters are also required to be computed during the calibration

process. One way is to directly compute the Dual of Absolute Quadric [Heyden and Astrom, 1996]. This approach is of interest as it encodes both the Dual Image Absolute Conic (DIAC) and its enclosing plane, the plane at infinity, in one geometric entity, (3.22). Recall that the DIAC itself embodies the camera intrinsic parameters, (3.21).

One of the first approaches in this class is proposed in [Triggs, 1997]. One of the problem with Triggs' method and similar ones is that it would not converge if a combination of enough rotation about two non-parallel axes and translation in sequences of images is not available [Hassanpour and Atalay, 2004].

To address the above issue and provide more robust results Zisserman et al. [Zisserman et al., 1998] suggest to use all the available geometric constraints. Fortunately, geometric scene constraints are abundant in man-made environments. Examples are perpendicularities [Boufama and Habed, 2007] and planar constraints [Liebowitz and Zisserman, 1999, Menudet et al., 2008, Cui and Ngan, 2010]. One of the earliest planar based calibration methods is proposed in [Triggs, 1998] which requires at least five views of a planar scene. In our work we focus on the scenes with parallel and perpendicular planes.

Huynh and Heyden [Huynh and Heyden, 2005] incorporated orthogonal scene plane constraints in the dual of the absolute conic, which is a parameterization originally introduced by Triggs [Triggs, 1997]. Incorporation of the scene orthogonality is conducted through a nonlinear optimization framework. However, the numerical minimization requires an starting point, which is obtained from a linear solution. The linear solution itself is obtained from the assumption that the principal point is in the middle of the image.

In general, since the camera autocalibration is nonlinear, the problem of choosing the initial values of parameters is often difficult. In addition, initializing the optimization procedures close to the ground truth does not necessarily guarantee the convergence to the desired solution [Boufama and Habed, 2007]. For instance, when the candidate plane at infinity contains one of the camera centers, the optimization fails even if the motion of the camera is not degenerate. Fusiello et al. [Fusiello, 2001] proposed a globally convergent method that uses interval analysis in order to bound the values of the camera parameters. However, the excessive running time of this method makes it inappropriate for most applications.

To address the issues above, we propose a method to use geometric constraints to improve the stability of nonlinear calibration yet not requiring any estimation of starting point. Our method uses perpendicularity and parallelism of planes in the scene to impose constraints on the values of camera parameters. The proposed method combines the two constraints into a single nonlinear framework. The method does not require accurate starting points and convergences even with a very inaccurate ones from random guesses. Thus, it does not need any assumption about the principal points being in the center of images. The results of our experiments show that our proposed method converges almost all the times.

6.3 Proposed methods

Throughout this discussion we assume that a projective reconstruction of the scene described by

$$p_{i,j} \simeq M_i P_j \tag{6.1}$$

is already available as follows

- n reconstructed 3D points in projective space P_j , $j = 1..n$,
- projective projection matrices of m views M_i , $i = 1..m$,
- projection of j^{th} point onto i^{th} view denoted by $p_{i,j}$.

Also, world coordinate is attached to the first view coordinate so $M_1 \simeq [I \mid 0]$. The projective projection of the rest of views is defined with respect to the first view as $M_i \simeq [A_i \mid a_i]$, where A_i is a 3×3 matrix and a_i is a 3- vector.

In section 5.3 we showed that the linear dependency of planes can be translated into a linear relationship between the inter-image homographies induced by the corresponding planes. Three planes are linearly dependent if and only if they intersect in a line. If Γ , Π and Φ are three such planes, the linear relation

$$\Gamma = c_1\Pi + c_2\Phi, \quad (6.2)$$

is valid for the scalars c_1 and c_2 . Therefore, same relation is valid for their induced homographies

$$H_{\Gamma i} = c_1 H_{\Pi i} + c_2 H_{\Phi i}, \quad (6.3)$$

where the matrices $H_{\Gamma i}$, $H_{\Pi i}$ and $H_{\Phi i}$ are those of the inter-image homographies induced by Γ , Π and Φ respectively, between the reference image and the i^{th} one.

Remark 1. Three parallel planes also meet in a line; that is, a line that lies on the plane at infinity.

Remark 2. The scalars c_1 and c_2 in equations (6.2) have the same value in equation (6.3) provided the homographies are calculated through the steps explained in Section 5.2. This helps to calculate the scalars from

Remark 3. The equations (6.2) and (6.3) can also be written in the following forms:

$$\Gamma \simeq (c_1/c_2)\Pi + \Phi, H_{\Gamma i} \simeq (c_1/c_2)H_{\Pi i} + H_{\Phi i},$$

where c_1/c_2 is replaced by α in the subsequent discussions.

The relationship (6.3) is true for intersecting planes and parallel planes as well. Because, two parallel planes intersect the plane at infinity in a line, these planes and the plane at infinity, ∞ , are linearly dependent and so are their inter-image homographies.

For two orthogonal planes, there is no such linear relationship. However, the two are related by the dual absolute quadric, Q_∞^* in the following manner:

$$\Pi_1 Q_\infty^* \Pi_2^T = 0, \tag{6.4}$$

where Π_1 and Π_2 are two orthogonal planes.

In the subsequent discussions we propose approaches to incorporate the scene constraints imposed by parallel planes and orthogonal ones in autocalibration approaches. Parallel planes are affine invariant scene constraints which are used in affine calibrations. Similarly, orthogonal planes remain orthogonal only under metric transformations. Therefore plane orthogonality are used in the proposed metric camera calibration approach.

6.3.1 A pair of parallel planes

Affine camera calibration is equivalent to locating the plane at infinity or calculating its homography. In this section, we present two such calibration methods using only parallel planes in the scene. Our first method is dedicated to the case where only one pair of parallel planes is available and images are captured by a moving camera with constant intrinsic parameters.

One pair, Π and Φ , of parallel scene planes defines the plane at infinity (up to an arbitrary scale factor) up to a one-parameter family of planes $\infty \simeq \alpha\Pi + \Phi$. With $\infty = (v_\infty^T \ 1)^T$ in \mathbb{P}^3 , we have

$$(v_\infty^T \ 1)^T \simeq \alpha(v_\Pi^T \ 1)^T + (v_\Phi^T \ 1)^T \Rightarrow v_\infty \simeq \frac{1}{1+\alpha}(\alpha v_\Pi + v_\Phi)$$

or

$$v_\infty \simeq \alpha v_\pi + v_\phi \quad \left| \begin{array}{l} \alpha \neq -1 \\ \alpha \neq 0 \end{array} \right. \quad (6.5)$$

where α is a scalar considering the following principals.

- If $\alpha = -1$, then ∞ would contain the origin, which is not possible.
- If $\alpha = 0$, then Φ would coincides with the plane at infinity ∞ which is against the assumption that Φ is a scene plane.

$$H_\infty(i) \simeq \alpha H_\pi(i) + H_\phi(i) \quad \left| \begin{array}{l} \alpha \neq -1 \\ \alpha \neq 0 \end{array} \right. \quad (6.6)$$

holds for the same value of the scalar α throughout the sequence provided $H_\pi(i)$ and $H_\phi(i)$ are calculated as follows.

$$\begin{cases} H_\pi^T(i) \simeq [I \mid -v_\pi]P_i^T \\ H_\phi^T(i) \simeq [I \mid -v_\phi]P_i^T \end{cases} \quad (6.7)$$

Homography $H_\pi(i)$ is induced from plane Π and relates the two projections of plane Π onto the reference image and i^{th} image. Additional images do not introduce new unknowns as long as the homographies are scaled as suggested in Remark 2. Assuming

the parameters of the camera are kept constant, the eigenvalues of $H_\infty(i)$ must have equal moduli [Pollefeys and Gool, 1999]. For each camera i , except the reference one, the characteristic equation of $H_\infty(i)$ is given by $\det(\alpha H_\pi(i) + H_\phi(i) - \lambda I) = 0$ which expands to

$$a_i(\alpha)\lambda^3 + b_i(\alpha)\lambda^2 + c_i(\alpha)\lambda + d_i(\alpha) = 0 \quad (6.8)$$

where the coefficients a, b, c and d are in terms of the real unknown α (considering $\alpha \neq -1$, $\alpha \neq 0$ required by equation (5.8)). A necessary condition for the eigenvalues of $H_\infty(i)$ to have equal moduli [Pollefeys and Gool, 1999] can be expressed as:

$$\left\{ \begin{array}{l} a_i(\alpha)c_i^3(\alpha) - b_i(\alpha)d_i^3(\alpha) = 0 \\ \alpha \text{ has to be a real number} \\ \alpha \neq -1 \\ \alpha \neq 0 \end{array} \right. \quad (6.9)$$

Each image i , $i \neq 1$, (except the first one chosen as reference), provides a sixth polynomial equation (5.11) in the unknown α , the solutions of which are the eigenvalues of the associated 6×6 companion matrix. The sought α is at the intersection of the solution sets of (5.11) obtained over all pairs of images.

6.3.2 Two pairs of parallel planes

Our second affine calibration method can be used as soon as two pairs of parallel planes are present in the scene and makes no assumption about the camera parameters.

When two images, captured by two different cameras, the modulus constraint and the method described in previous section is no longer applicable. In the following, we

propose a method that makes use of two pairs of parallel planes observed by two or more cameras with possibly distinct intrinsic parameters.

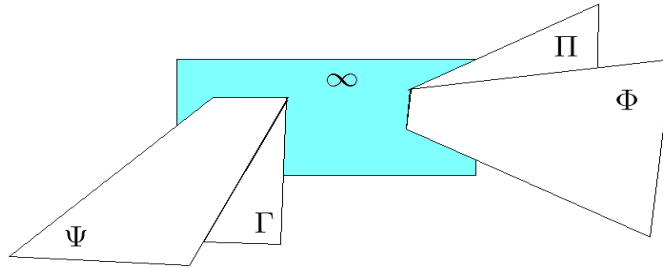


Figure 6-1. Two pairs of parallel planes meeting plane at infinity

If three planes, two of which are parallel, are linearly dependent, then the third plane either is parallel to the other two or coincides with the plane at infinity. Consider two pairs (Π, Φ) and (Ψ, Γ) of parallel planes: Π parallel to Φ and Ψ parallel to Γ but neither Γ nor (obviously) Ψ are parallel to any of Π or Φ . Furthermore, consider a fifth plane Ξ that is simultaneously linearly dependent upon (Π, Φ) and (Ψ, Γ) .

Assuming Ξ not at infinity, it suggests that Ξ must be simultaneously parallel to two unparallel planes which is obviously impossible. Therefore, Ξ must necessarily coincide with the plane at infinity, $\Xi \equiv \infty$. As a result, only the plane at infinity satisfies $\infty = \alpha\Pi + \beta\Phi$ and $\infty = \gamma\Gamma + \delta\Psi$. Thus, given the coordinates of Π, Φ, Ψ and Γ in \mathbb{P}^3 , the unknown coordinates ∞ and the values of scalars α, β, γ and δ can be obtained by solving a linear system of equations (for example through singular value decomposition). Alternatively, the homography of the plane at infinity can also be obtained from the ones induced by our four planes:

$$\left\{ \begin{array}{l} M_i \begin{bmatrix} I \\ -v_\infty^T \end{bmatrix} = \alpha H_{\Pi_i} + \beta H_{\Phi_i} \\ M_i \begin{bmatrix} I \\ -v_\infty^T \end{bmatrix} = \gamma H_{\Psi_i} + \delta H_{\Gamma_i} \end{array} \right. \quad (6.10)$$

Therefore, v_∞ , α , β , γ and δ can be obtained by solving a linear system of equations (also using SVD in practice). For a given pair of planes, each image does not introduce new unknowns (see Remark 2). However, each additional pair of planes introduces its own unknown coefficients that also need to be calculated. Fortunately, most scenes contain only a limited number of parallel planes.

6.3.3 Metric reconstruction from plane parallelism and orthogonality

The recovered structure described in (6.1) is projective. That means for any pair of $\{M_i, P_j\}$, the pair $\{M_i T, T^{-1} P_j\}$ also satisfies the equation (6.1):

$$p_{i,j} \simeq M_i T T^{-1} P_j,$$

where T is any nonsingular 4×4 matrix. To search for a particular rectifying matrix T which upgrades the recovered projective structure to metric, there are two main approaches, stratified and direct. In stratified one, we first affine reconstruct the scene by locating the plane at infinity and the upgrade it to metric by computing the image of absolute conic [Pollefeys and Gool, 1999]. In direct approaches, we compute both the location of plane at infinity and the image of absolute conic at the same time. Direct methods are generally done by calculation of dual absolute quadric which embodies both entities [Triggs, 1997].

In projective space the DAQ is not in its canonical position Q_∞^{*E} given by equation (3.12) but in a transformed position:

$$Q_\infty^* \simeq TQ_\infty^{*E}T^T. \quad (6.11)$$

Therefore, we compute DAQ in projective space first and then decompose it to equation (6.11) to obtain that rectifying matrix T which takes the projective DAQ back to its canonical form. The obtained T also upgrades the entire scene to metric [Hartley and Zisserman, 2004].

To compute DAQ in projective space, we start from the following relation $K_i K_i^T = \omega_i^* \simeq M_i Q_\infty^* M_i^T$ between the absolute dual quadric Q_∞^* and its images ω_i^* projected on views. Since we want the modulus constraints to be valid, we assume the camera parameters are the same for all the views. Therefore, all views will have the same dual of image of absolute conic and the mentioned relation is simplified to:

$$K K^T = \omega^* \simeq M_i Q_\infty^* M_i^T \quad (6.12)$$

where K denotes the camera matrix and M_i is the projective projection matrix for view i . There are two types of solutions to computer Q_∞^* from (6.12), the linear solution and the nonlinear one which is the main contribution of this work.

Existing linear solution

The goal in this approach is to obtain a set of linear relations from equation (6.12) which can be solved using linear approaches such as SVD. One way is to assume that the principal point of each view is in the middle of the image of that view [Pollefeys et al.,

2004]. Using at least 3 views the dual absolute conic Q_∞^* can be linearly computed as a 4×4 symmetric matrix:

$$Q_\infty^* = \begin{bmatrix} q1 & q2 & q3 & q4 \\ q2 & q5 & q5 & q7 \\ q3 & q5 & q8 & q9 \\ q4 & q7 & q9 & q10 \end{bmatrix}$$

such that $\begin{cases} Q_\infty^* \text{ is positive semidefinite} \\ Q_\infty^* \text{ is rank 3} \end{cases}$ (6.13)

which is defined by 10 unknowns out of which only 8 are independent.

A *positive semidefinite* matrix [Marcus and Minc, 1988] is a Hermitian matrix all of whose eigenvalues are nonnegative. \hat{Q} is a symmetric matrix, therefore $\hat{Q} = \hat{Q}^T$, and that means \hat{Q} is a Hermitian matrix. Thus, the positive semidefiniteness of a symmetric matrix is met when all the eigen values have the same sign.

The *rank deficiency* is verified when the $\det(Q_\infty^*) = 0$. This can be forced through SVD decomposition of Q_∞^* where the smallest eigen value is set to zero. The ratio of smallest eigen value to the second smallest eigen value is an indication of the accuracy of the estimated Q_∞^* .

Once the linear solution is available it can be used as an starting point for nonlinear approaches which usually deliver better accuracy. This way of obtaining linear solution is used in [Huynh and Heyden, 2005] as a starting point for the nonlinear optimization where orthogonal scene plane constraints are incorporated.

Proposed approach

In our proposed nonlinear approach, as opposed to existing approaches [Huynh and Heyden, 2005], [Pollefeys et al., 2004] we do not need any initial linear solutions, neither

do we need to make any assumption about the location of principal points. In this approach a pair of scene parallel planes and orthogonal planes are incorporated in the autocalibration equations described in (6.12). We solve for the dual absolute quadric (DAQ) via the optimization of the following nonlinear system:

$$f(u_0, v_0, f_x, f_y, s, v_\infty) = \sum_{i=2}^m \frac{K K^T}{\|K K^T\|_F} - \frac{M_i Q_\infty^* M_i^T}{\|M_i Q_\infty^* M_i^T\|_F}, \quad (6.14)$$

$$\text{such that } \begin{cases} \text{para}(\Pi_1, \Pi_2) \\ \text{orth}(\Pi_1, \Pi_3) \end{cases}$$

where Π_1 and Π_2 are a pair of parallel plane in the scene. There is a third plane, Π_3 , which is orthogonal to the previous ones. Instead of the third plane being orthogonal to the other two, we could also assume there are two separate orthogonal planes. $\|\cdot\|$ denotes the Frobenius norm of the matrix concerned. The planes are defined in projective frame as follows:

$$\Pi_1 \simeq (v_1, 1)^T, \quad \Pi_2 \simeq (v_2, 1)^T, \quad \Pi_3 \simeq (v_3, 1)^T.$$

The unknowns u_0, v_0, f_x, f_y, s are the camera intrinsic parameters and $v_\infty = (p_1, p_2, p_3)$ is the 3-vector of plane at infinity in $\infty \simeq (v_\infty^T, 1)^T$. The two functions $\text{orth}(\Pi_1, \Pi_3)$ and $\text{para}(\Pi_1, \Pi_2)$ define the nonlinear constrains which we elaborate more.

From the previous discussions The DAQ can be parameterized by the DIAC of the reference view ω^* and the plane at infinity of that view $\infty \simeq (v_\infty^T, 1)^T$ as:

$$Q_\infty^* \simeq \begin{bmatrix} \omega^* & -(\omega^* v_\infty) \\ -(\omega^* v_\infty)^T & v_\infty^T \omega^* v_\infty \end{bmatrix}, \quad (6.15)$$

where $v_\infty = (p_1, p_2, p_3)$, which means 3 unknowns. However, we already assumed that the two planes Π_1 and Π_2 are parallel. Hence, from our discussion in previous chapter, this relation holds:

$$\infty \simeq \Pi_1 + \alpha \Pi_2 \Rightarrow v_\infty = \frac{1}{1 + \alpha}(v_1 + \alpha v_2), \quad (6.16)$$

which transforms a relation with three unknown to same relation with one unknown only. This reduces the number of unknowns in the equation (6.14) to six only. Hence, in our proposed approach we solve the following nonlinear constrained equation:

$$f(u_0, v_0, f_x, f_y, s, \alpha) = \sum_{i=2}^m \frac{K K^T}{\|K K^T\|_F} - \frac{M_i Q_\infty^* M_i^T}{\|M_i Q_\infty^* M_i^T\|_F}, \quad (6.17)$$

$$\text{such that } \begin{cases} \text{para}(\Pi_1, \Pi_2) \\ \text{orth}(\Pi_4, \Pi_3) \end{cases}.$$

In other words, our proposed method combines planes orthogonality and parallelism in a single optimization framework.

The set of nonlinear equations are formed by the expansion of the equation (6.17) as follows. From $K K^T = \omega^* \simeq M_i Q_\infty^* M_i^T$ and considering K as

$$K = \begin{bmatrix} f_u & s & u_0 \\ & f_v & v_0 \\ & & 1 \end{bmatrix}$$

we get

$$\begin{bmatrix} f_u^2 + s^2 + u_0^2 & s f_v + u_0 v_0 & u_0 \\ s f_v + u_0 v_0 & f_v^2 + v_0^2 & v_0 \\ u_0 & v_0 & 1 \end{bmatrix} \simeq M_i \begin{bmatrix} \omega^* & -(\omega^* v_\infty) \\ -(\omega^* v_\infty)^T & v_\infty^T \omega^* v_\infty \end{bmatrix} M_i^T, \quad i : 2..m, \quad (6.18)$$

where the v_∞ is replaced by its equivalent in equation (6.16).

Nonlinear constraints

The scene planes parallelism and perpendicularity were transferred to the calibration equations through the following nonlinear constraints

$$Para(\Pi_1, \Pi_2) = \|Q_\infty^* \Pi_1 - Q_\infty^* \Pi_2\|^2 \quad (6.19)$$

as $\infty \simeq \Pi_1 + \alpha \Pi_2$, $Q_\infty^* \infty = 0$. The equation (6.19) enforces constraints on the two planes by bounding the value of function $Para(\Pi_1, \Pi_2)$ around zero. Similarly, the value of the function below in equation (6.20) is hold close to zero by enforcing orthogonality constraints on planes during the optimization process.

$$Orth(\Pi_4, \Pi_3) = \frac{\Pi_4^T Q_\infty^* \Pi_3}{\sqrt{\Pi_4^T Q_\infty^* \Pi_4 \cdot \Pi_3^T Q_\infty^* \Pi_3}} \quad (6.20)$$

Planes orthogonality constraint could be implemented in an easier form used in existing approaches [Huynh and Heyden, 2005, Triggs, 1997] as simple as $\Pi_4^T Q_\infty^* \Pi_3$. However, we found (6.20) more stable to noise (due to the division by the expersion at the denominator) .

6.4 Experiments

We have conducted experiments using both simulated data and real images. The goal in all the experiments was to assess the quality of the affine and metric 3D reconstruction obtained using each of the three methods proposed. The results of experiments are demonstrated in two sections. First using only parallel constraints, second using both parallel and orthogonal scene constraints. The implementation is done in Matlab using the default optimization toolbox.

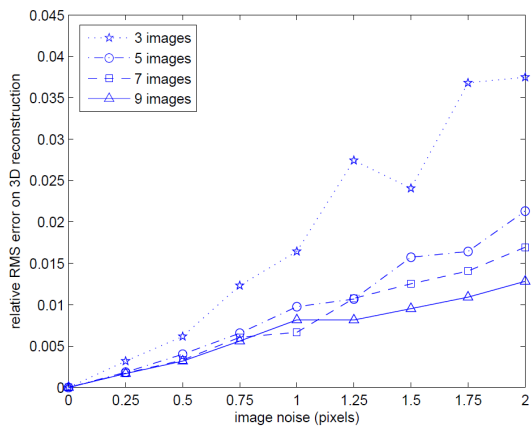
6.4.1 Using parallel planes constraints

In the following, the affine calibration method (described in Section 4.1) using a camera with constant parameters is referred to as C1. The method (in Section 4.2) that considers cameras with possibly different intrinsic parameters will be referred to as D2.

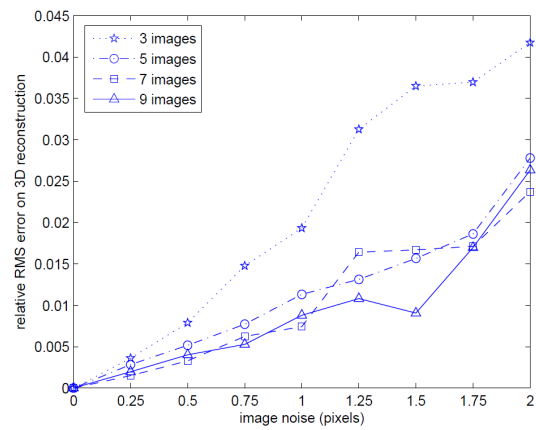
Simulated data: Our experiments with simulated data were carried out for a varying number of images and various levels of noise added to image coordinates. We have tested our methods C1 and D2 on perfectly parallel planes and with noise added to the angles between pairs of parallel planes. In each experiment, a scene, consisting of randomly generated pairs of parallel planes, was projected onto the image planes.

Each plane consisted of 50 randomly generated scene points scattered in a disc of radius 1. Parallel planes were generated at a mean distance of 0.75 units with a 0.25 standard deviation from each other. In each pair of parallel planes, the disc of one plane was centered in the origin of the scene's reference frame. Method C1 was tested on scenes each consisting of only one pair of parallel planes. Method D2 was tested with two pairs of parallel planes making an angle of 25 degrees about each axis with 5 degrees standard deviation. Noise on the angle between a pair of parallel planes was added by rotating one

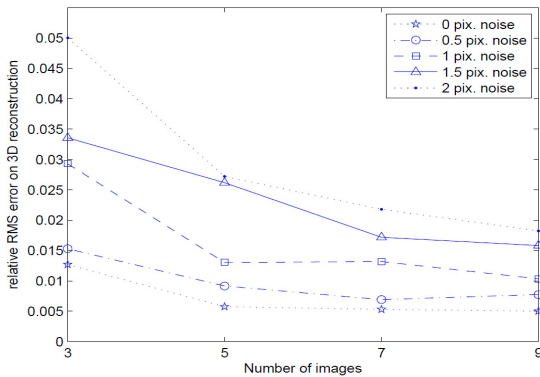
of the planes by a zero-mean randomly generated angle with 1 degree standard deviation about each axis. The cameras were generated at mean distance of 2 units and a 0.3 standard deviation from the origin of the scene. The camera's were roughly oriented towards the origin of the scene's reference frame. Zero-mean Gaussian noise with standard deviation in the range 0 to 2 pixels was added to the pixel coordinates. Camera parameters were kept constant for experiments using method C1. Using D2, they were varied by a random amount in the range -15% to +15% of their original values.



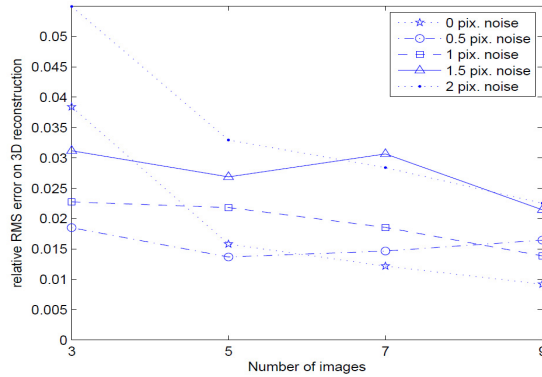
(a) C1/parallel.



(b) D2/parallel.



(c) C1/perturbed.



(d) D2/perturbed.

Figure 6-2. Relative RMS 3D error using methods C1 (a and c) and D2 (b and d).

The recovered affine 3D structure was aligned with the original data and the relative RMS error was used as a quality measurement of our reconstruction. The results of our experiments using methods C1 and D2 with perfectly parallel planes are reported in Figure 6-2.a and Figure 6-2.b respectively. Those of our experiments with noise on the angles between parallel planes are reported in Figure 6-2.(c) and Figure 6-2.d for methods C1 and D2 respectively. Method D2 was also tested in minimal conditions with 2 images and the results are reported in Figure 6-3 for both perfectly parallel planes and perturbed ones.

In all these figures, each point is the average of 1000 independent trials for a given noise level in pixel coordinates and a given number of images (3, 5, 7 and 9 images). The results we have obtained show that, for both methods, the quality of the affine reconstruction improves with the increasing number of images and progressively deteriorates with the amplitude of errors in point localization in images (image noise). However, the relative errors remain within acceptable bounds even with 3 images and 2 pixels of noise: about 4% with both methods when using perfectly parallel planes. The results of our experiments with noise added to the angles between parallel planes (Figure 6-2.c and Figure 6-2.d) suggest that the planes need not be perfectly parallel for both methods to provide acceptable results.

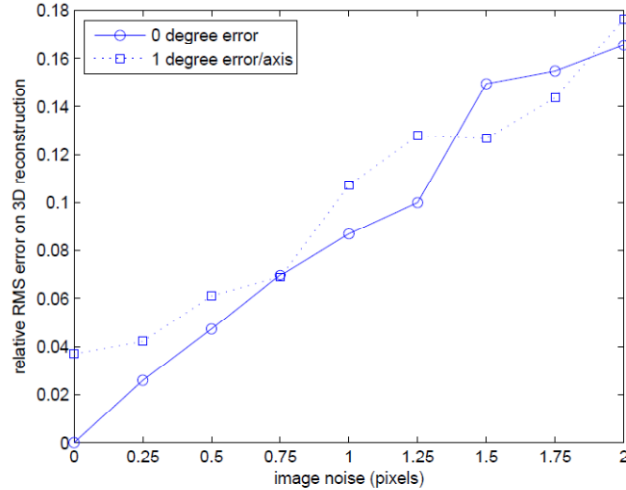
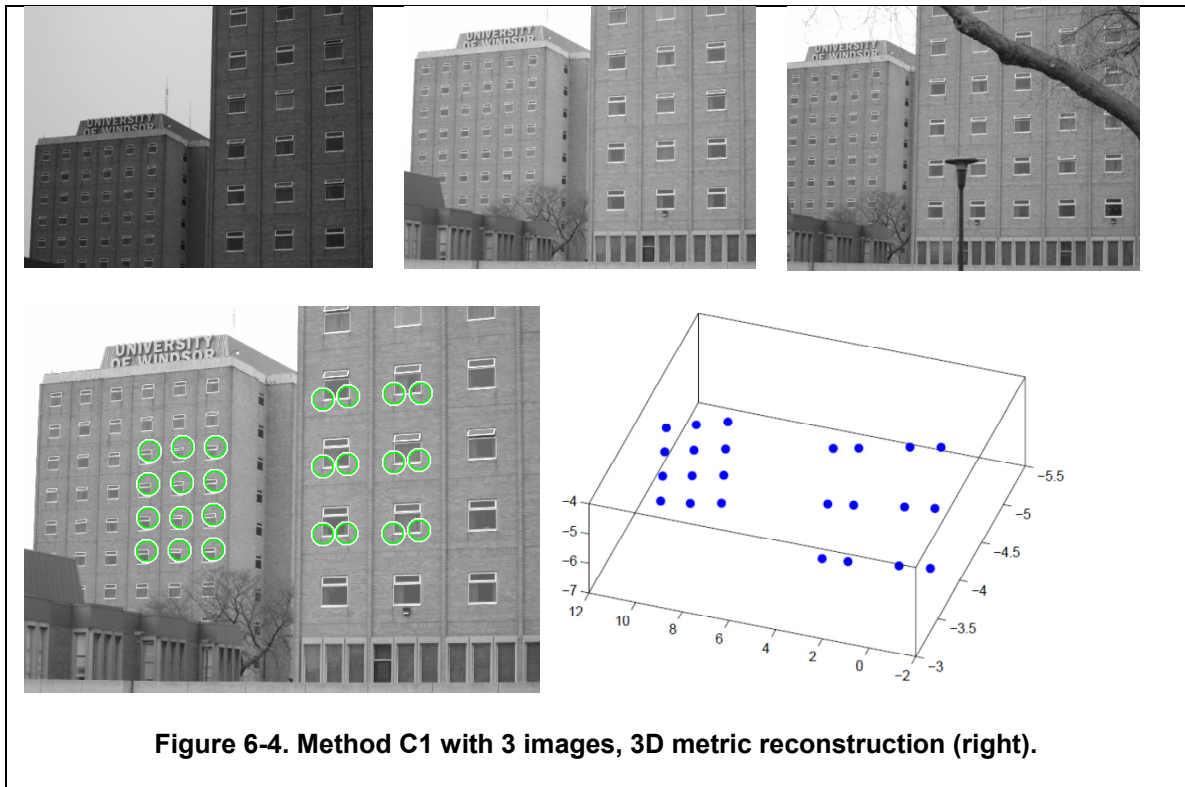


Figure 6-3. Two pairs of perfectly/perturbed parallel planes and 2 images (method D2).

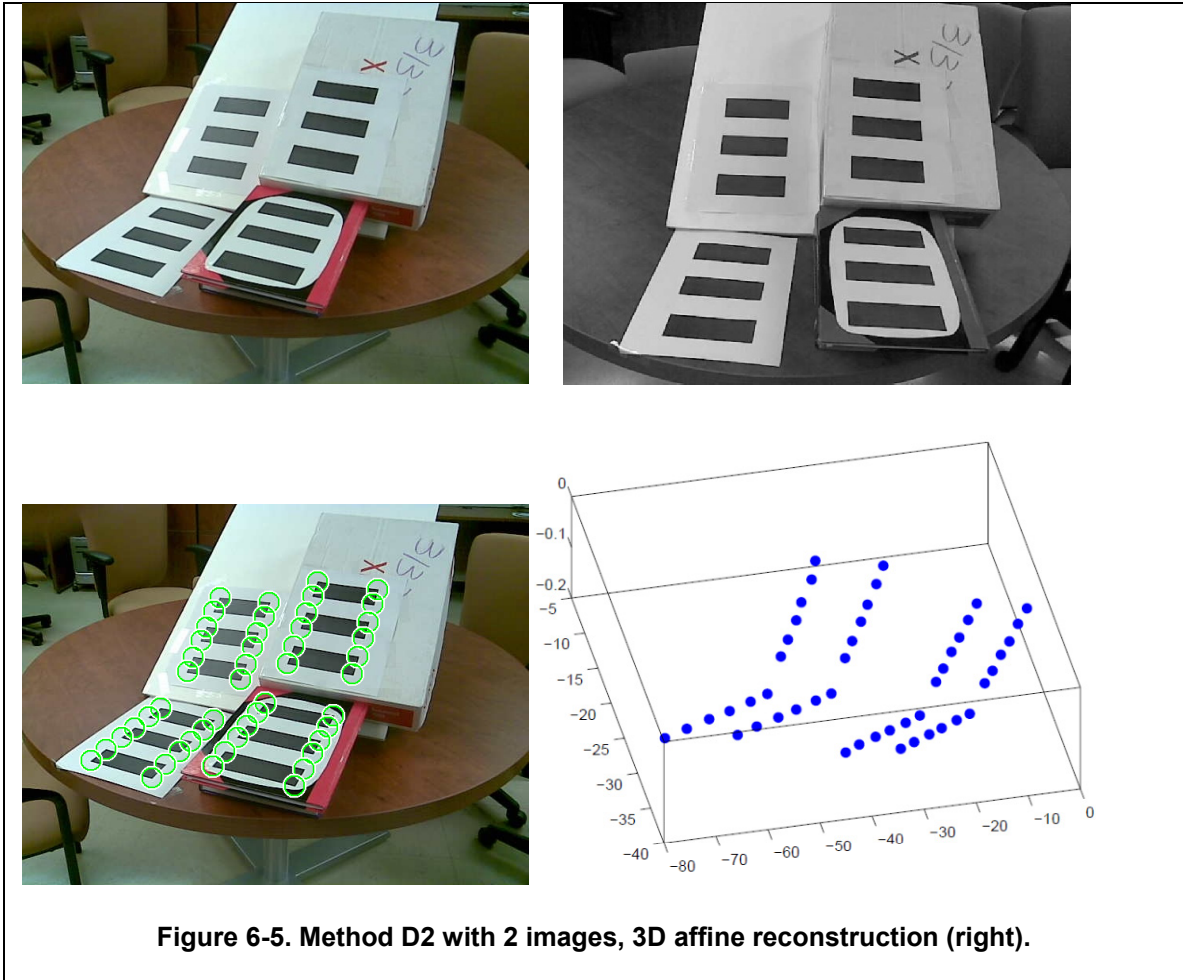
Method D2 is more sensitive to errors on the angles as, unlike C1, it relies on plane parallelism only. When using only 2 images with method D2, Figure 6-3., the 3D errors have reached 18% with 2 pixels of noise with perturbed parallel planes and a little more than 16% with perfectly parallel planes.

Real images

We have used the parallel front walls of two buildings, Figure 6-4, to affinely calibrate a low-end Sony Cybershot DSC-S930 camera from 3 images with our C1 method. We have successfully recovered the 3D affine structure of the scene and upgraded it to a metric one (Figure 6-3. right). The reconstructed points were detected and matched across all three images (12 points on each wall as marked in Figure 6-3. left).



Our second example, is a scene consisting of two pairs of parallel sheets of paper with three black rectangles on each (12 points on each plane as marked in Figure 6-5.left). We have used method D2 to obtain the affine 3D reconstruction of this scene from only two images. Two different cameras have been used: the camera used in the "buildings" example and the camera of a cell phone. The resulting affine reconstruction is given in Figure 6-5.right. The 3D reconstruction results we have obtained in these examples and others as well were very satisfactory.



6.4.2 Using parallel and perpendicular scene constraints

The performance of the proposed camera calibration method using parallel and perpendicular scene constraints is investigated under extensive simulations with synthetic data and with real uncalibrated images from low-end cameras. The goal in these experiments was to investigate that whether incorporating planar scene constraints allows our method to converge from pretty much anywhere, which is not the case when using the unconstrained method.

Simulated data

In these set of experiments with simulated data, the scene configuration is very similar to that of Section 6.4.1 except that the scene contains orthogonal planes too. The

experiments were carried out for a varying number of images and various levels of noise added to image coordinates. Therefore, zero-mean Gaussian noise with standard deviation in the range 0 to 2 pixels was added to the pixel coordinates. In each experiment, a scene, consisting of randomly generated pair of parallel planes and a third plane which was orthogonal to the two was generated and projected onto the image planes.

Each plane consisted of 50 randomly generated scene points scattered in a disc of radius 1. Parallel planes were generated at a mean distance of 0.5 units with a 0.2 standard deviation from each other. In each pair of parallel planes, the disc of one plane was centered in the origin of the scene's reference frame. A third plane was placed perpendicular to the other two with the distance of 0.2 standard deviation from the scene origin (Figure 6-6).

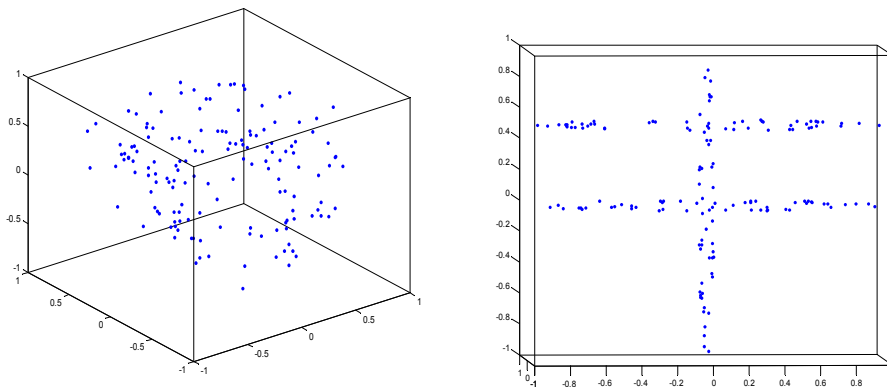


Figure 6-6. Scene configuration for simulated data

The cameras were located at mean distance of 2 from the origin with standard deviation 0.3. The camera's were roughly oriented towards the origin of the scene's reference frame so that the entire scene is visible for all cameras. Zero-mean Gaussian noise with

standard deviation in the range 0 to 2 pixels was added to the pixel coordinates. The image sizes were 640×480 for all the views during the entire experiment. The camera intrinsic parameters were set known values and kept constant during each trial to assess the quality of results afterwards. The focal length was 900 units and the aspect ratio varied from 0.5 and 2. The principal point (u_0, v_0) was randomly picked from anywhere within the average distance of 50 pixels from the middle of the image. The skew factor is generally very close to zero especially for recently manufactured digital cameras. However, to cover the exceptional cases in our simulation we let the skew factor vary from -0.5 to 0.5.

As quality measurements to evaluate our calibration method, the relative RMS error of internal camera parameters from the true values were calculated. Also the recovered 3D structure was aligned with the original data to measure the 3D RMS reconstruction error.

The first set of results of conducted experiments is reported in Figure 6-7, where the camera internal parameters are plotted in 2D graphs. In all these figures, each point is the average of 200 independent trials for a given noise level in pixel coordinates and a given number of images (3, 5, 7 and 9 images). The obtained results show that the quality of the metric camera calibration improves with the increasing number of images and progressively deteriorates with the amplitude of errors in point localization in images (image noise). However, the relative RMS errors remain within acceptable bounds even with 3 images and 2 pixels of noise at about 8%.

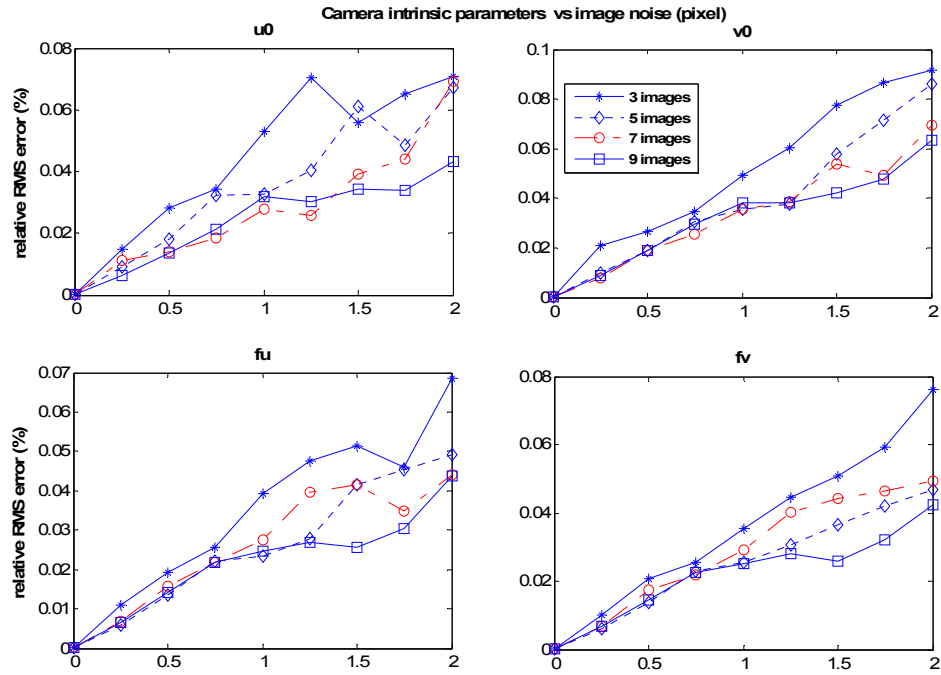


Figure 6-7. Camera parameters vs image noise (pixel) for 3 to 9 views

The 3D reconstruction RMS error was also measured during same experiment and the results are reported in Figure 6-8. It can be seen that the proposed approach can recover the 3D structure with acceptable accuracy even with the presence of two pixel noise which is too high for real world scenarios.

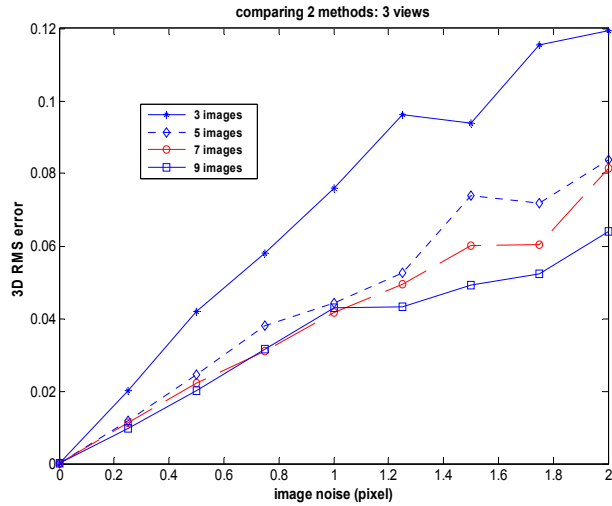


Figure 6-8. 3D RMS error of metric reconstruction

To investigate the effect of incorporating planar scene constraints on calibration we conducted the following experiment.

1- The optimization was run for both constrained method that we proposed and unconstrained method starting from ground truth.

2- The optimization was also run for both method starting from initial guesses. In the case of starting from initial guess, if a method converges towards the same result as starting from ground truth, this is counted as a success, a failure otherwise. The success rate of both methods is reported in Figure 6-9. The results show that starting from anywhere, with our proposed method, is pretty much like starting from ground truth.

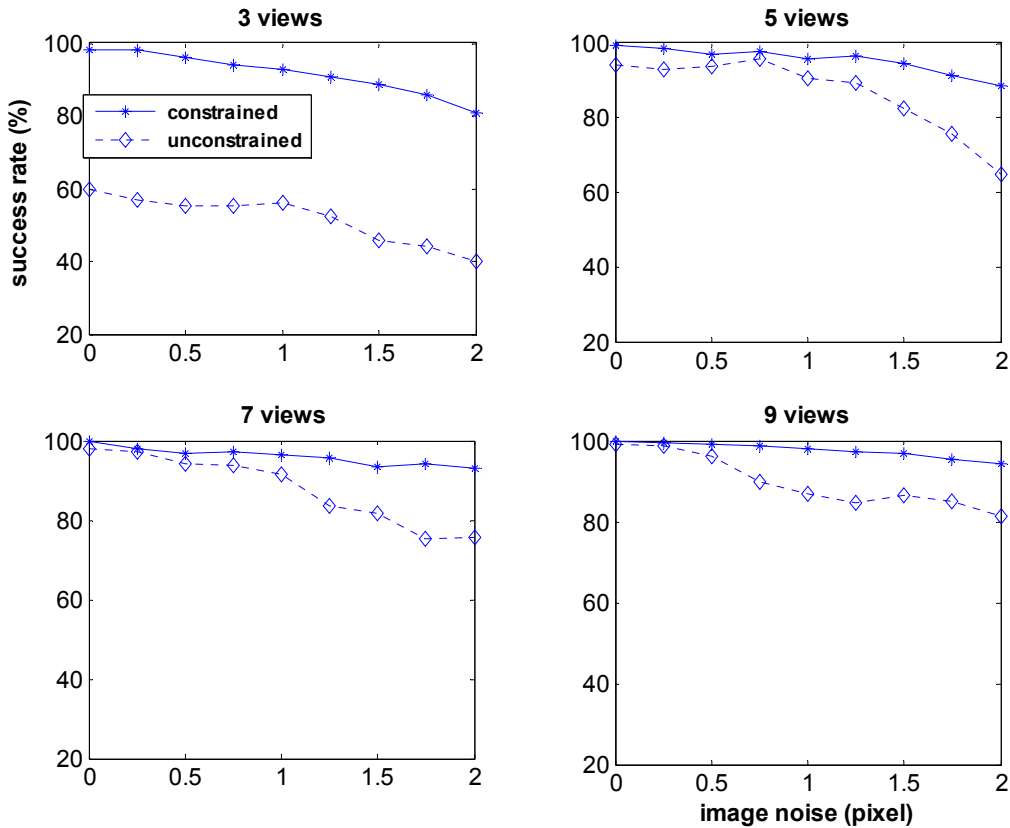


Figure 6-9. Comparing initial guess with ground truth

Real images

To evaluate the performance of the proposed metric calibration method in real world applications, we conducted experiments using real images as well. To evaluate the accuracy of recovered 3D model, we use same method applied in [Lourakis and Deriche, 2000]. That is the validity of results is demonstrated by calculating angles in the recovered 3D model, and comparing to the real angles of the actual scene. We provide four examples here, Figures 6-10 to 6-13.

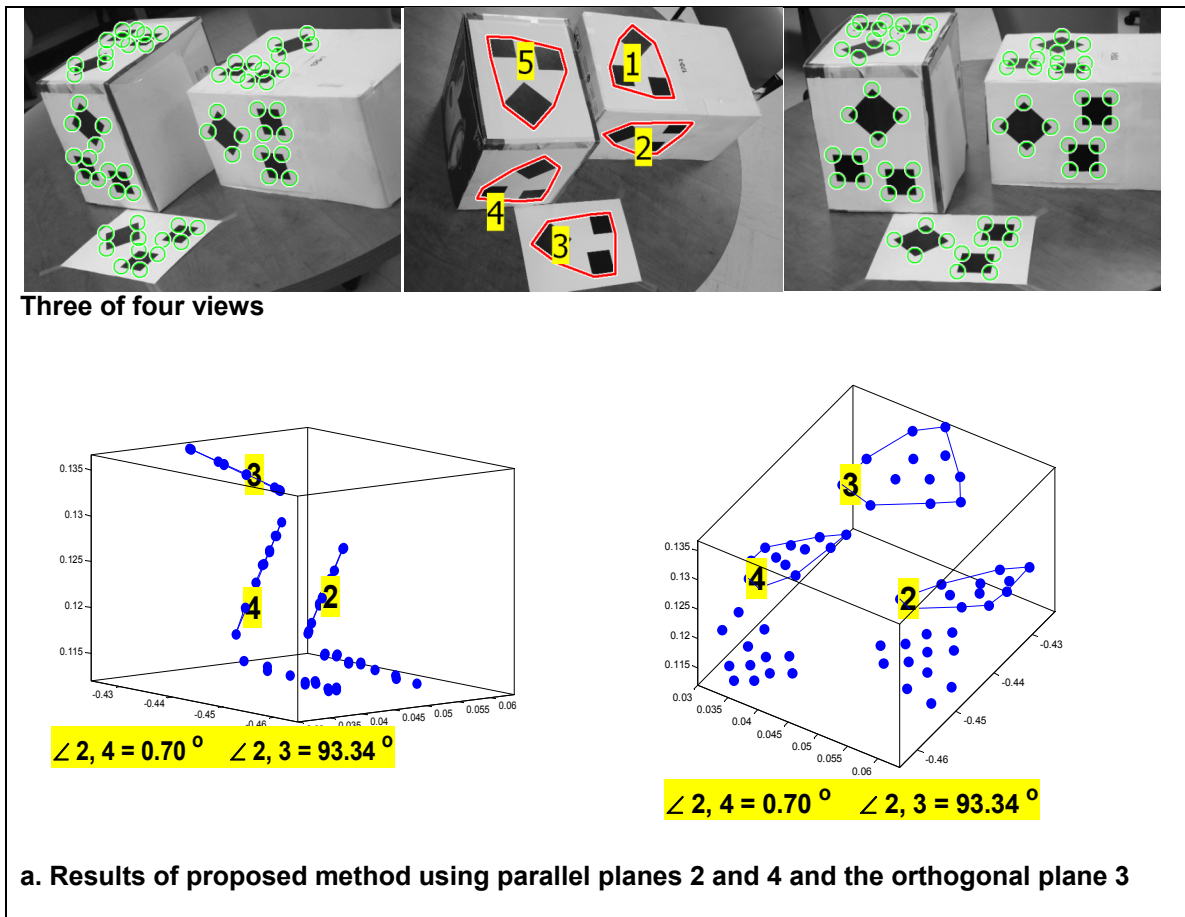
Boxes. Figure 6-10 presents a contrived scene consisting of five letter size white sheets of paper with black patterns pasted on the faces of two packaging boxes. In this example, 60

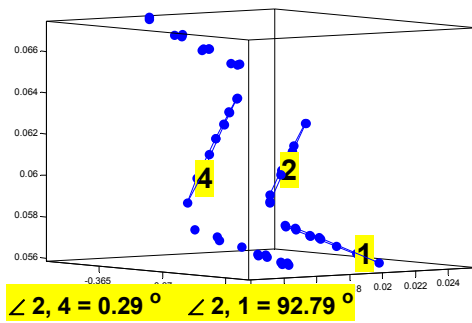
points (left image) lying on five planes (middle image) have been matched across four images. The planes labeled 1 and 3 and 5 are parallel to each other and perpendicular to the other two planes labeled 2 and 4. The four 640×480 images were shot by a low-end digital camera (Sony Cyber-shot DSC-S930). Since this scene contains multiple parallel and perpendicular planes, several experiments were conducted each time using different groups of planes.

In the first experiment with this scene, planes labeled 2 and 4 were taken as input parallel planes Π_1 , Π_2 and the plane labeled 3 as the plane Π_3 perpendicular the other two. The angles between the planes of the reconstructed model were computed and reported in the Figure 6-10.a. As seen, the computed angles between the two planes 2 and 3 are 93.34 degrees which is close to orthogonal. However the computed quantity was not supposed to be exactly 90 degrees, as the actual angle is in fact 92 degrees (Table 6-4).

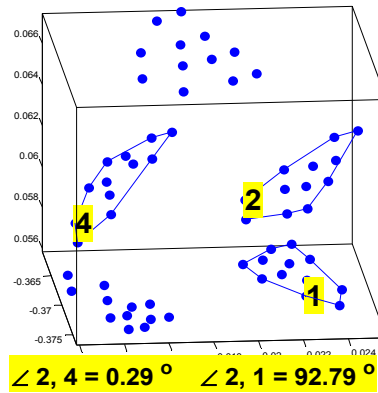
In another experiment, Figure 6-10.b, same planes labeled 2 and 4 were taken as input parallel planes Π_1 , Π_2 but this time plane labeled 1 was taken as Π_3 . It can be noticed that in first experiment the angle between planes labeled 2 and 4 was computed 0.70 degrees and this time the computed angle between same planes was reported 0.29 degrees. We expect the two quantities to be same, however picked planes are not perfectly parallel neither perfectly perpendicular. Yet, the results are satisfactory and better than that of Pollefeys method [Pollefeys, 2004], where scene constraints were not incorporated (Table 6-4). Same experiment was repeated this time for difrent planes and results are demonstrated in Figure 6-10.c. In Figure 6-10.d, the reconstructed model from the implementation of Pollefeys method [Pollefeys, 2004] is illustrated as a reference point to evaluate the performance of our method.

Table 6-4 Repost the accuracy of the proposed method. It computed angles from the reconstructed models with the corresponding actual angles of the real scene, which were measured manually. It also compares the accuracy of our proposed method with the method proposed in [Pollefeys, 2004] and nonlinear unconstrained method as well. Results clearly demonstrate better performance of the calibration algorithm with the way we incorporated the scene constraints.



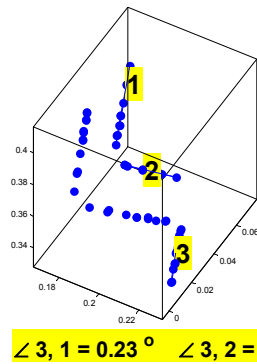


Side view

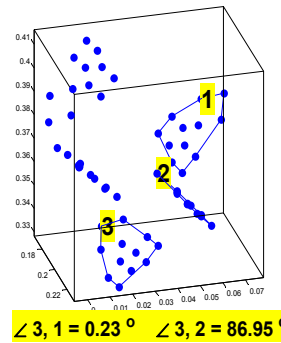


Top view

b. Results of proposed method using parallel planes 2 and 4 and the orthogonal plane 1

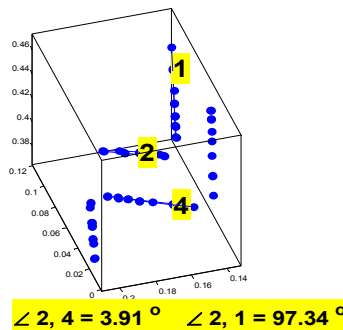


Side view

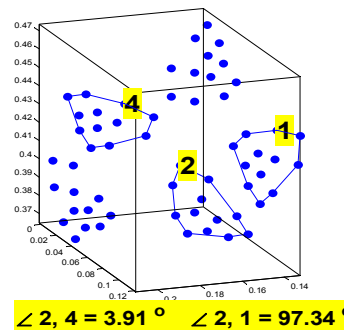


Top view

c. Results of proposed method using parallel planes 1 and 3 and the orthogonal plane 3



Side view



Top view

d. Results of Pollefeys method [Pollefeys, 2004]

Figure 6-10. Patterned sheets of paper pasted on the faces of packaging boxes

The three tables Table 6-1 to Table 6-1 report the angle between entire scene planes of scenario 1 in Figure 6-10. Table 6-1 reports the results when planes 2 and 4 are picked as parallel ones and the pair of planes 2 and 3 are picked as the perpendicular ones. Table 6-1 reports the results when the pair of planes 2 and 4 are assumed to be parallel and the planes 2 and 1 are assumed to be perpendicular. To compare the performance of the proposed algorithm the results of applying Pollefeys' method to same scene is reposted in Table 6-1 As seen, the results of the proposed method are clearly better than those obtained from Pollfeys' method in [Pollefeys, 2004].

Table 6-1. The angle between the planes for boxes in Figure 6-10. a (in degree) using the proposed constrained method

Plane#	2	4	3	1	5
2	0	0.7	93.34	98.99	99.95
4	0.7	0	92.7	98.37	99.33
3	93.34	92.7	0	6.14	7.58
1	98.99	98.37	6.14	0	1.62
5	99.95	99.33	7.58	1.62	0

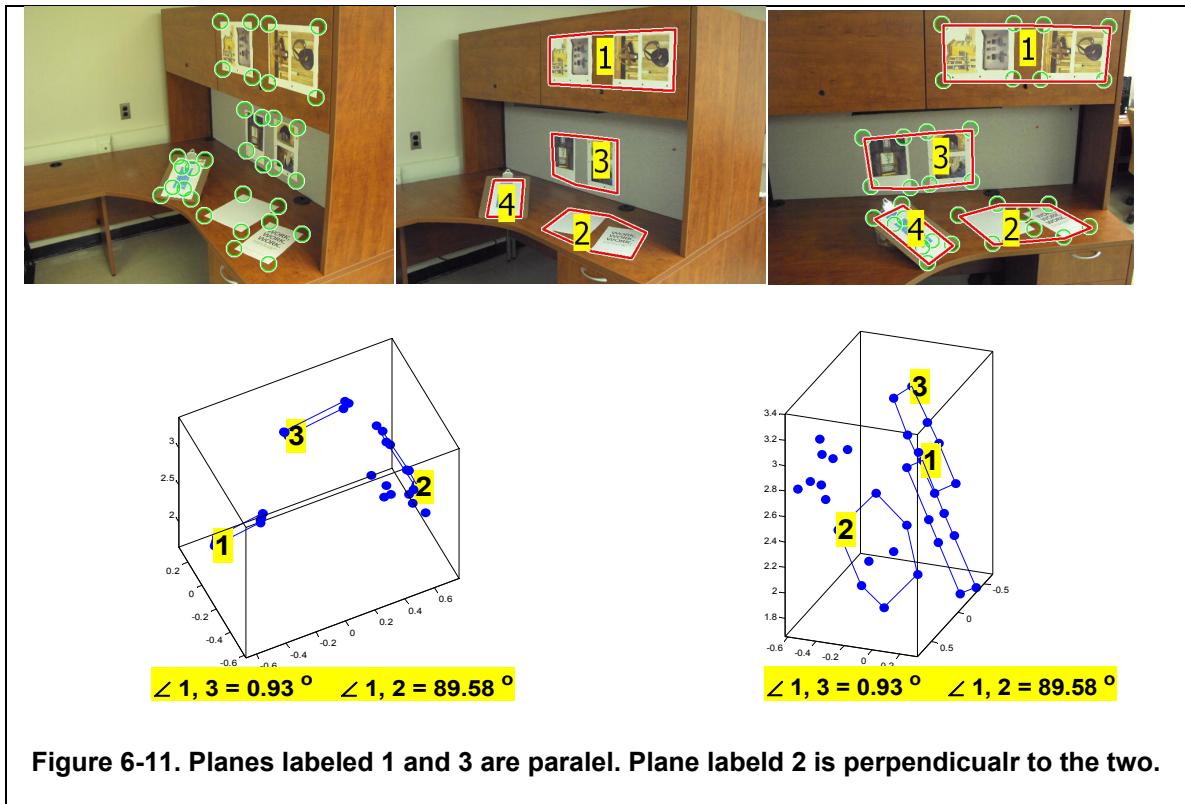
Table 6-2. The angle between the planes for boxes in Figure 6-10. b (in degree) using the proposed constrained method

Plane #	2	4	1	3	5
2	0	0.29	92.79	87.59	93.96
4	0.29	0	93.02	87.82	94.2
1	92.79	93.02	0	6.07	1.66
3	87.59	87.82	6.07	0	7.7
5	93.96	94.2	1.66	7.7	0

Table 6-3. The angle between the planes for boxes in Figure 6-10. b (in degree) using Pollefeys' method

Plane #	2	4	1	3	5
2	0	3.91	97.34	98.85	96.21
4	3.91	0	100.22	101.73	99.09
1	97.34	100.22	0	1.53	1.13
3	98.85	101.73	1.53	0	2.65
5	96.21	99.09	1.13	2.65	0

Office. In another experiment an indoor scene was tested on the proposed metric calibration approach. It is an office desk area of which 32 points (left image) lying on four distinct planes have been successfully detected and matched across four images. All four planes in Figure 6-11 have been identified. Planes labeled 1 and 3 are parallel and the plane label 2 is perpendicular to the two. Here also the results demonstrate the success of the proposed approach to deliver an acceptable calibration. 0.93 degree was obtained for the angle between parallel planes and 89.58 for the perpendicular ones. The unconstrained method failed on this scene (Table 6-4).



College. The outdoor scene in Figure 6-12, the Royal Victoria College in Montreal. In this example, 33 points (left image) lying on three planes (middle image) have been matched across three images only. Yet, our method has recovered a satisfactory metric

reconstruction of the real scene. The unconstrained nonlinear approach failed on this scenario and did not return meaningful results. For this scene, the actual angles of the real scene are not available, however it would appear that the angle between the perpendicular walls is closer to 90 degrees than 77 which was obtained from Pollefeys' method (Table 6-4).

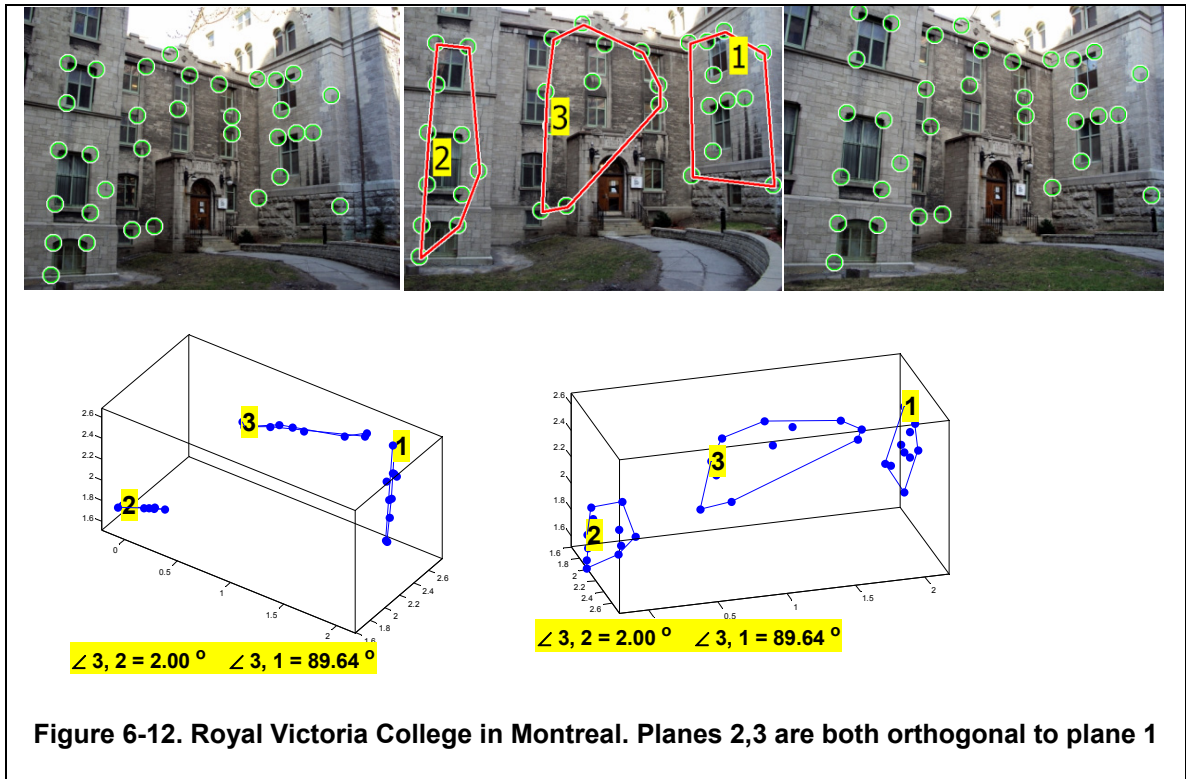


Table salt and business cards.

The outdoor scene in Figure 6-13, the table salt box and the business cards. In this example, 36 points (left image) lying on three planes (middle image) have been matched across four images. The pictures were taken using iPhone 4 rear camera. The images were originally 2592×1936 which were down sized to 478×640 to fit into our program's input. Our method has recovered a satisfactory metric reconstruction of the

real scene in this scenario as well. The unconstrained nonlinear approach and Pollefeys' method also delivered acceptable results in this particular experiment.

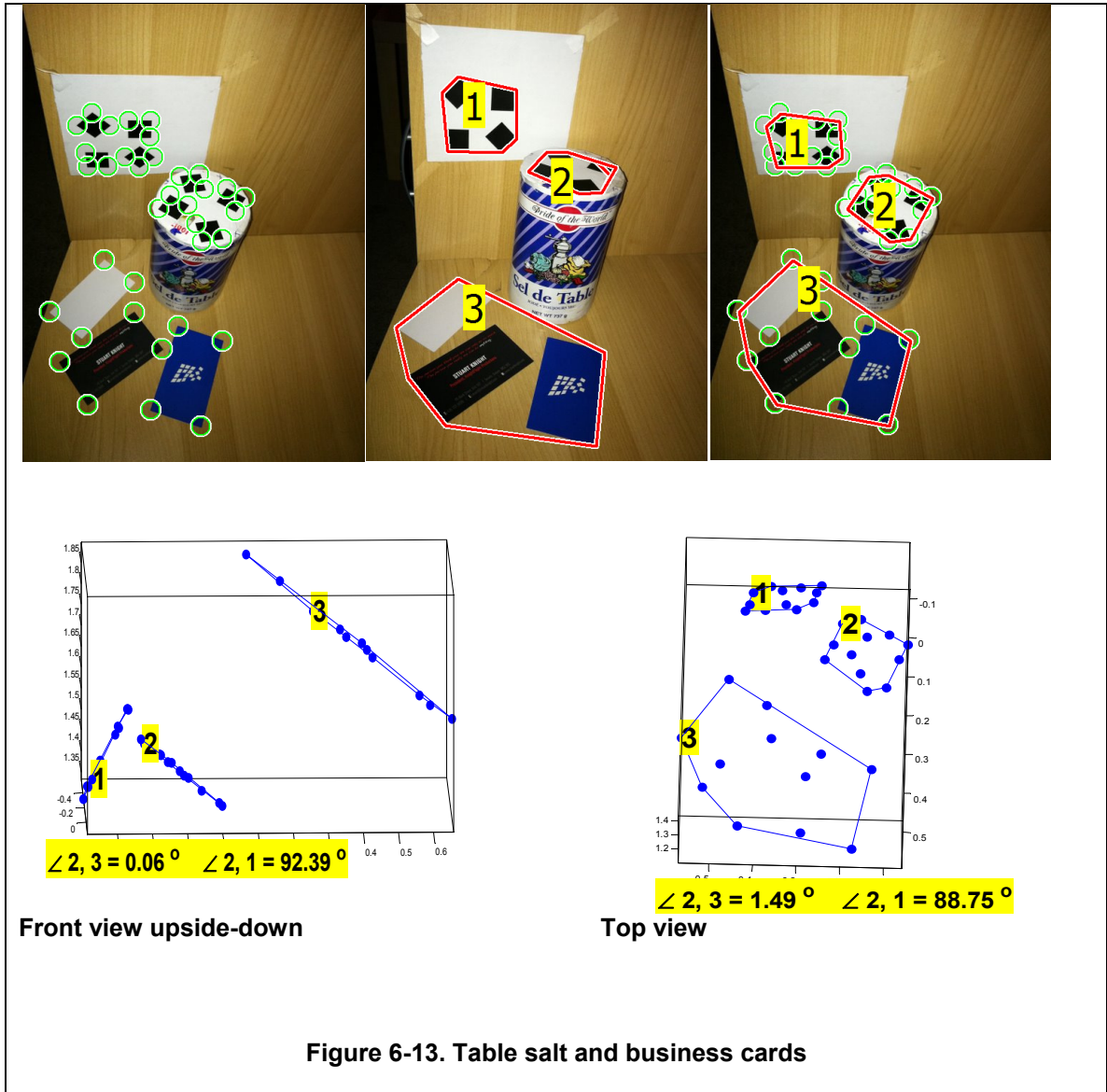


Table 6-4. Experiments with real images.

Scene	Boxes		Office		College		Table salt	
# planes	5		4		3		3	
# points	12		8		11		12	
Planes Method	$\angle\Pi_1, \Pi_2$ parallel	$\angle\Pi_1, \Pi_3$	$\angle\Pi_1, \Pi_2$ parallel	$\angle\Pi_1, \Pi_3$	$\angle\Pi_1, \Pi_2$ parallel	$\angle\Pi_1, \Pi_3$	$\angle\Pi_1, \Pi_2$ parallel	$\angle\Pi_1, \Pi_3$
Pollefeys	3.91°	97.34°	4.47°	77.56°	3.93°	87.03°	1.49°	88.75°
Unconstrained	0.13°	54°	failed	failed	failed	failed	4.21°	88.19°
Constrained	0.70°(a)	93.34°(a)	0.93°	89.58°	2°	89.64°	0.06°	92.39°
Actual angles	2°	92°	1.3°	90°	NA	NA	1.5°	91°

6.5 Conclusion

In this chapter, we have proposed autocalibration methods for both affine and metric calibration of cameras. Two methods are proposed for the affine calibration of cameras from parallel planes. The first method works with a minimum of one pair of parallel planes and a moving camera with constant parameters. Our second method is linear and works even with images captured using different cameras. This method works with a minimum of two images and two pairs of planes (in different directions) and no other constraints. The results of our extensive experiments have demonstrated that the affine 3D reconstruction is very good.

We have also propose a metric calibration method that combine geometric constraints with an existing autocalibration method. It uses perpendicularity and parallelism of planes in the scene to impose constraints on the values of camera parameters. The proposed method particularly combines the two constraints into a single

nonlinear framework. The method does not require accurate starting points and convergences even with a very inaccurate ones from random guesses. Thus, it does not need any assumption about the principal points being in the center of images, which is usually used for estimating a linear starting point for nonlinear camera calibrations. The simulation results showed that our method with minimal constraints (a pair of parallel planes and a pair of perpendicular ones) has very good convergence properties.

In all experiments with real scene, low end cameras were used to take the pictures. As well, the images were resized down to 640×480 or smaller then the matching phase was conducted. Shrinking the image before matching usually leads to lower the accuracy of matched feature points. Yet, all proposed methods in this dissertation delivered very good results on real images.

7 Conclusion and Remarks

This dissertation aimed at achieving two main goals: the identification of plane-based geometric properties in a scene from uncalibrated images and, their use for recovering the three-dimensional structure of the scene. The proposed approaches are suitable to urban area scenes which are abundant in planar structures, e.g., buildings, streets and certain kinds of manufactured objects such as furniture. Such scenes provide numerous constraints for camera calibration and 3D reconstruction. In particular, these man-made environments contain plenty of planes that are generally either parallel or perpendicular to each other.

A general solution was proposed in Chapter 4 for the automatic identification of physical planes in a scene using uncalibrated images. We have overcome the problem of existing homography-based approaches where in addition to real planes, undesirable virtual planes are also extracted from images. We have presented a new scheme, in conjunction with the homography-based plane detection method, for the identification of virtual plane so that they can be removed. The proposed approach uses non-coplanar points inside an extracted plane to decide whether the plane is physical or virtual. As a consequence, we were able to compute the homographies for all potential physical planes.

In Chapter 5, we have proposed a novel method for parallel planes identification from point correspondences across three or more images. Our method neither requires the camera to be calibrated nor a 3D reconstruction to be calculated. It relies on prior detection of planes, the outcome from Chapter 4, and point correspondences only. The

results of the numerous experiments we have conducted have shown that the proposed method achieves its goal with a very high success rate, even for high levels of noise.

In Chapter 6, we have proposed affine and metric camera calibration methods using the results from Chapters 4 and 5. Our first affine camera calibration method uses a minimum of one pair of parallel planes and a moving camera with constant parameters. A second affine calibration method was proposed for the case of images taken by different cameras, i.e., non-constant intrinsic parameters. The method uses a minimum of 2 images and 2 pairs of parallel planes (with different orientations). The obtained results have clearly demonstrated that both proposed methods yield very good 3D affine structure of the scene. A metric calibration method was also proposed in this chapter. In this method, we have combined scene geometric constraints, arising from parallel and/or perpendicular planes, with existing autocalibration methods. In particular, the two geometric constraints were translated into a single nonlinear framework. We have obtained very good results for the metric reconstruction on both simulated and real images. Although we have low end cameras for all our real experiments, the quality of the obtained results have exceeded our expectations.

In brief:

- We proposed a 3-point homography-based method to identify major scene planes from images. In that work a merging phase was proposed to obtain robust results. A general scheme was introduced to distinguish between virtual and physical planes.
- A novel reconstruction free method was introduced to identify the scene parallel planes.

- The parallel planes were used to recover the affine structure of the scene for 1) using a camera with constant parameters 2) using various cameras to shoot pictures
- We also proposed a novel nonlinear autocalibration method which combines both planes orthogonality and parallelism in a single optimization framework. The method upgrades the existing projective structure to a metric one directly. Although nonlinear, our method has a satisfactory convergence rate even when the optimization procedure is randomly initialized.

Futurework

Accurate identification of plane scenes is useful for our future work where edge points will be used to extract the actual facets of objects in the scene. Once such facets are identified, they can be reconstructed using the proposed refinement method to allow for an accurate and fast 3D modeling of planar scenes. The obtained results from this dissertation can help advance computer vision research towards its ultimate goal, total image understanding. In particular, the extraction of planes can be combined with edges to obtain physical facets that will lead to the semantic segmentation of the scene. In addition, the automatic detection of parallelism and perpendicularity will help in deriving semantic relationships between facets and objects. Our proposed plane identification method [Amintabar and Boufama, 2008] was referred to by [Dornaika and Elder, 2010]. The authors use the plane identification to ease the process of image registration in registering high-resolution, small field-of-view images with low-resolution panoramic images provided by a panoramic catadioptric video sensor. In another work, our plane identification method was cited by [Wang, Z. and Zhao, 2011]. The authors mention our

approach to detect and discarding virtual planes using convex set theory and a voting strategy.

References

- [Amintabar and Boufama, 2008] Amintabar, A. and Boufama, B. (2008). Homography-based plane identification and matching. In *15th IEEE International Conference on Image Processing (ICIP)*, pages 297–300. IEEE.
- [Amintabar and Boufama, 2009] Amintabar, A. and Boufama, B. (2009). The distinction between virtual and physical planes using homography. *Image Analysis and Recognition*, pages 727–736.
- [Armstrong et al., 1996] Armstrong, M., Zisserman, A., and Hartley, R. (1996). Euclidean reconstruction from image triplets. *European Conference on Computer Vision -ECCV'96*, pages 3–16.
- [Boufama and Habed, 2007] Boufama, B. and Habed, A. (2007). Three-dimensional reconstruction using the perpendicularity constraint. In *Sixth International Conference on 3-D Digital Imaging and Modeling (3DIM'07)*, pages 241–248. IEEE.
- [Boufama et al., 1993] Boufama, B., Mohr, R., and Veillon, F. (1993). Euclidean constraints for uncalibrated reconstruction. In *Fourth International Conference on Computer Vision*, pages 466–470. IEEE.
- [Bouguet and Perona, 1998] Bouguet, J. and Perona, P. (1998). 3D photography on your desk. In *Sixth International Conference on Computer Vision*, pages 43–50. IEEE.
- [Cardon et al., 2005] Cardon, D., Fife, W., Archibald, J., and Lee, D. (2005). Fast 3D reconstruction for small autonomous robots. In *31st Annual Conference of IEEE Industrial Electronics Society, 2005. (IECON)*, pages 6–pp. IEEE.
- [Chan, 1996] Chan, T. (1996). Optimal output-sensitive convex hull algorithms in two and three dimensions. *Discrete and Computational Geometry*, 16(4):361–368.
- [Chandraker, 2009] Chandraker, M. (2009). *From Pictures to 3D: Global Optimization for Scene Reconstruction*. PhD Thesis, University of California, San Diego.
- [Chandraker et al., 2007] Chandraker, M., Agarwal, S., Kahl, F., Nistér, D., and Kriegman, D. (2007). Autocalibration via rank-constrained estimation of the absolute quadric. In *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR'07.*, pages 1–8. IEEE.

- [Chandraker et al., 2010] Chandraker, M., Agarwal, S., Kriegman, D., and Belongie, S. (2010). Globally Optimal Algorithms for Stratified Autocalibration. *International Journal of Computer Vision*, pages 1–19.
- [Choi et al., 2007] Choi, O., Kim, H., and Kweon, I. (2007). Simultaneous plane extraction and 2D homography estimation using local feature transformations. *Asian Conference on Computer Vision (ACCV)*, pages 269–278.
- [Chuan et al., 2003] Chuan, Z., Da Long, T., Feng, Z., and Li, D. (2003). A planar homography estimation method for camera calibration. In *Computational Intelligence in Robotics and Automation, 2003. Proceedings. 2003 IEEE International Symposium on*, volume 1, pages 424–429. IEEE.
- [Criminisi et al., 2000] Criminisi, A., Reid, I. D., and Zisserman, A. (2000). Single view metrology. *International Journal of Computer Vision*, 40(2):123–148.
- [Cui and Ngan, 2010] Cui, C. and Ngan, K. (2010). Plane-based external camera calibration with accuracy measured by relative deflection angle. *Signal Processing: Image Communication*, 25(3):224–234.
- [Cui and Zhu, 2009] Cui, S. and Zhu, X. (2009). A generalized reference-plane-based calibration method in optical triangular profilometry. *Optics Express*, 17(23):20735–20746.
- [Cyganek et al., 2009] Cyganek, B., Siebert, J., and Corporation, E. (2009). *An Introduction to 3D Computer Vision Techniques and Algorithms*. Wiley Online Library.
- [De Agapito et al., 1999] De Agapito, L., Hayman, E., and Hartley, R. (1999). Linear self-calibration of a rotating and zooming camera. In *IEEE Conference on Computer Vision and Pattern Recognition (cvpr)*, pages 1015–1022. Published by the IEEE Computer Society.
- [Ducke et al., 2011] Ducke, B., Score, D., and Reeves, J. (2011). Multiview 3D reconstruction of the archaeological site at Weymouth from image series. *Computers & Graphics*.
- [Espuny et al., 2011] Espuny, F., Aranda, J., and Burgos Gil, J. (2011). Camera self-calibration with parallel screw axis motion by intersecting imaged horopters. *Image Analysis*, pages 1–12.

- [Faugeras et al., 1992] Faugeras, O., Luong, Q., and Maybank, S. (1992). Camera self-calibration: Theory and experiments. In *European Conference on Computer Vision—ECCV’92*, pages 321–334. Springer.
- [Faugeras and Maybank, 1990] Faugeras, O. and Maybank, S. (1990). Motion from point matches: multiplicity of solutions. *International Journal of Computer Vision*, 4(3):225–246.
- [Fusiello, 2001] Fusiello, A. (2001). A new autocalibration algorithm: Experimental evaluation. In *Computer Analysis of Images and Patterns*, pages 717–724. Springer.
- [Gasparini and Sturm, 2009] Gasparini, S. and Sturm, P. (2009). Multi-view matching tensors from lines for general camera models. *Tensors in Image Processing and Computer Vision Springer*, pages 198–214.
- [Gong and Xu, 2004] Gong, R. and Xu, G. (2004). Quadratic surface reconstruction from multiple views using sqp. *Integrated image and graphics technologies*, pages 197–217.
- [Groeber et al., 2006] Groeber, M., Haley, B., Uchic, M., Dimiduk, D., and Ghosh, S. (2006). 3D reconstruction and characterization of polycrystalline microstructures using a FIB-SEM system. *Materials characterization*, 57(4-5):259–273.
- [Gurdjos et al., 2010] Gurdjos, P., Bartoli, A., and Sturm, P. (2010). Is dual linear self-calibration artificially ambiguous. In *IEEE 12th International Conference on Computer Vision, 2009*, pages 88–95. IEEE.
- [Habed et al., 2010a] Habed, A., Amintabar, A., and Boufama, B. (2010a). AFFINE CAMERA CALIBRATION FROM HOMOGRAPHIES OF PARALLEL PLANES. *IEEE International Conference on Image Processing ICIP*.
- [Habed et al., 2010b] Habed, A., Amintabar, A., and Boufama, B. (2010b). Reconstruction-free parallel planes identification from uncalibrated images. *20th International Conference on Pattern Recognition ICPR*.
- [Habed and Boufama, 2006] Habed, A. and Boufama, B. (2006). Camera self-calibration from bivariate polynomial equations and the coplanarity constraint. *Image and Vision Computing*, 24(5):498–514.
- [Hammarstedt et al., 2006] Hammarstedt, P., Kahl, F., and Heyden, A. (2006). Affine reconstruction from translational motion under various autocalibration constraints. *Journal of Mathematical Imaging and Vision*, 24(2):245–257.

- [Hartley, 1994a] Hartley, R. (1994a). Euclidean reconstruction from uncalibrated views. *Applications of invariance in computer vision*, pages 235–256.
- [Hartley, 1994b] Hartley, R. (1994b). Self-calibration from multiple views with a rotating camera. *European Conference on Computer Vision—ECCV’94*, pages 471–478.
- [Hartley, 1997] Hartley, R. (1997). In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593.
- [Hartley, 1998] Hartley, R. (1998). The chirality. *Int. Journal of Computer Vision*, 26(1):41–61.
- [Hartley et al., 2002] Hartley, R., Hayman, E., de Agapito, L., and Reid, I. (2002). Camera calibration and the search for infinity. In *The Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999.*, volume 1, pages 510–517. IEEE.
- [Hartley and Zisserman, 2004] Hartley, R. and Zisserman, A. (2004). *Multiple view geometry in computer vision*. Cambridge Univ Pr.
- [Hassanpour and Atalay, 2004] Hassanpour, R. and Atalay, V. (2004). Camera auto-calibration using a sequence of 2D images with small rotations. *Pattern recognition letters*, 25(9):989–997.
- [He and Chu, 2006] He, Q. and Chu, C. (2006). Planar surface detection in image pairs using homographic constraints. *Advances in Visual Computing*, pages 19–27.
- [Heyden and Astrom, 1996] Heyden, A. and Astrom, K. (1996). Euclidean reconstruction from constant intrinsic parameters. In *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, volume 1, pages 339–343. IEEE.
- [Heyden and Astrom, 1999] Heyden, A. and Astrom, K. (1999). Flexible calibration: Minimal cases for auto-calibration. In *The Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999.*, volume 1, pages 350–355. IEEE.
- [Heyden and Huynh, 2002] Heyden, A. and Huynh, D. (2002). Auto-calibration via the absolute quadric and scene constraints. *Proceedings - International Conference on Pattern Recognition*, 16(2):631–634.
- [Huynh and Heyden, 2005] Huynh, D. and Heyden, A. (2005). Scene point constraints in camera auto-calibration: an implementational perspective. *Image and Vision Computing*, 23(8):747–760.

- [Jennings et al., 1977] Jennings, A., McKeown, J., and Jennings, A. (1977). *Matrix computation for engineers and scientists*. John Wiley & Sons.
- [Kahl et al., 2000] Kahl, F., Triggs, B., and Åström, K. (2000). Critical motions for auto-calibration when some intrinsic parameters can vary. *Journal of Mathematical Imaging and Vision*, 13(2):131–146.
- [Kanatani, 2008] Kanatani, K. (2008). Latest progress of 3-D reconstruction from multiple camera images. *Robotics Research Trends*, pages 33–75.
- [Kato et al., 1994] Kato, K., Nakanishi, T., Shio, A., and Ishii, K. (1994). Structure from image sequences captured through a monocular extra-wide angle lens. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994*, pages 919–924. IEEE.
- [Kawakam et al., 2006] Kawakam, H., Ito, Y., and Kanazawa, Y. (2006). A robust method for detecting planar regions based on random sampling. *Systems and Computers in Japan*, 37(4):11–22.
- [Kawasaki and Furukawa, 2009] Kawasaki, H. and Furukawa, R. (2009). Shape reconstruction and camera self-calibration using cast shadows and scene geometries. *International Journal of Computer Vision*, 83(2):135–148.
- [Khan and Shah, 2009] Khan, S. M. and Shah, M. (2009). Tracking multiple occluding people by localizing on multiple scene planes source. *IEEE transactions on pattern analysis and machine intelligence*, 31:505–519.
- [Kirchhof, 2008] Kirchhof, M. (2008). Linear constraints in two-view multiple homography estimation of uncalibrated Scenes. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. ISPRS*, pages 13–20.
- [Klein, 2004] Klein, F. (2004). *Elementary mathematics from an advanced standpoint: Arithmetic, algebra, analysis*. Dover Pubns.
- [Kruppa, 1913] Kruppa, E. (1913). *Zur Ermittlung eines Objektes aus zwei Perspektiven mit innerer Orientierung*. Holder.
- [Lhuillier, 2008] Lhuillier, M. (2008). Automatic scene structure and camera motion using a catadioptric system. *Computer Vision and Image Understanding*, 109(2):186–203.

- [Liebowitz and Zisserman, 1999] Liebowitz, D. and Zisserman, A. (1999). Combining scene and auto-calibration constraints. In *International Conference on Computer Vision (iccv)*, pages 293–300. Published by the IEEE Computer Society.
- [Lourakis and Deriche, 2000] Lourakis, M. and Deriche, R. (2000). Camera self-calibration using the singular value decomposition of the fundamental matrix: From point correspondences to 3d measurements. *Proceedings of the Asian Conference on Computer Vision, Taipei, Taiwan*, 1:403–408.
- [Luong and Faugeras, 1997] Luong, Q. and Faugeras, O. (1997). Self-calibration of a moving camera from point correspondences and fundamental matrices. *International Journal of Computer Vision*, 22(3):261–289.
- [Manning and Dyer, 2001] Manning, R. and Dyer, C. (2001). Affine calibration from moving objects. In *IEEE International Conference on Computer Vision, (ICCV)*, volume 1, pages 494–500. IEEE.
- [Marcus and Minc, 1988] Marcus, M. and Minc, H. (1988). *Introduction to Linear Algebra*. New York: Dover.
- [Maybank and Faugeras, 1992] Maybank, S. and Faugeras, O. (1992). A theory of self-calibration of a moving camera. *International Journal of Computer Vision*, 8(2):123–151.
- [Menudet et al., 2008] Menudet, J., Becker, J., Fournel, T., and Mennessier, C. (2008). Plane-based camera self-calibration by metric rectification of images. *Image and Vision Computing*, 26(7):913–934.
- [Moons et al., 1996] Moons, T., Van Gool, L., Proesmans, M., and Pauwels, E. (1996). Affine reconstruction from perspective image pairs with a relative object-camera translation in between. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(1):77–83.
- [Nicolas et al., 2005] Nicolas, G., Guerrero, J., Pellejero, O., and Sagues, C. (2005). Computing homographies from three lines or points in an image pair. *International Conference on Image Analysis and Processing (CIAP)'05*, pages 446–453.
- [Nistér, 2004] Nistér, D. (2004). Untwisting a projective reconstruction. *International Journal of Computer Vision*, 60(2):165–183.

- [Piazzzi and Prattichizzo, 2006] Piazzzi, J. and Prattichizzo, D. (2006). Plane detection with stereo images. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 922–927. IEEE.
- [Pollefeys, 2004] Pollefeys, M. (2004). Visual 3d modeling from images. *Tutorial Notes. Chapel Hill, NC: University of North Carolina.*
- [Pollefeys and Gool, 1999] Pollefeys, M. and Gool, L. V. (1999). Stratified self-calibration with the modulus constraint. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):707–724.
- [Pollefeys et al., 2008] Pollefeys, M., Nistér, D., Frahm, J., Akbarzadeh, A., Mordohai, P., Clipp, B., Engels, C., Gallup, D., Kim, S., Merrell, P., et al. (2008). Detailed real-time urban 3d reconstruction from video. *International Journal of Computer Vision*, 78(2):143–167.
- [Pollefeys and Van Gool, 1997] Pollefeys, M. and Van Gool, L. (1997). Self-calibration from the absolute conic on the plane at infinity. In *Computer Analysis of Images and Patterns*, pages 175–182. Springer.
- [Pollefeys et al., 1996] Pollefeys, M., Van Gool, L., and Oosterlinck, A. (1996). The Modulus Constraint: A New Constraint for Self-Calibration. In *Proceedings of the 1996 International Conference on Pattern Recognition (ICPR'96) Volume I-Volume 7270*, pages 349–353. IEEE Computer Society.
- [Pollefeys et al., 2004] Pollefeys, M., Van Gool, L., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J., and Koch, R. (2004). Visual modeling with a hand-held camera. *International Journal of Computer Vision*, 59(3):207–232.
- [Proesmans and Van Gool, 1997] Proesmans, M. and Van Gool, L. (1997). Reading between the lines—a method for extracting dynamic 3D with texture. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 95–102. ACM.
- [Rothwell et al., 1995] Rothwell, C., Faugeras, O., and Csurka, G. (1995). Different paths towards projective reconstruction. *Workshop on Geometric Modeling & Invariants for Computer Vision, Xian, China*, pages 245–252.
- [Ruf et al., 1998] Ruf, A., Csurka, G., and Horaud, R. (1998). Projective translations and affine stereo calibration. In *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pages 475–481. IEEE.

- [Schaffalitzky, 2000] Schaffalitzky, F. (2000). Direct solution of modulus constraints. In *Proc. Indian Conf. on Computer Vision, Graphics and Image Processing*, pages 314–321.
- [Seo et al., 2006] Seo, K., Lee, J., and Choi, H. (2006). An efficient detection of vanishing points using inverted coordinates image space. *Pattern Recognition Letters*, 27(2):102–108.
- [Seo and Heyden, 2004] Seo, Y. and Heyden, A. (2004). Auto-calibration by linear iteration using the DAC equation. *Image and Vision Computing*, 22(11):919–926.
- [Simond and Rives, 2008] Simond, N. and Rives, P. (2008). What can be done with an embedded stereo-rig in urban environments? *Robotics and Autonomous Systems*, 56(9):777–789.
- [Snow, 1996] Snow, M. (1996). *Charting Presence in Virtual Environments and Its Effects on Performance*. PhD thesis, Virginia Polytechnic Institute and State University.
- [Sturm, 1997] Sturm, P. (1997). Critical motion sequences for monocular self-calibration and uncalibrated Euclidean reconstruction. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 1100–1105. IEEE.
- [Sturm, 2000] Sturm, P. (2000). A case against Kruppa’s equations for camera self-calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(10):1199–1204.
- [Sturm and Quan, 1995] Sturm, P. and Quan, L. (1995). Affine stereo calibration. In *Computer Analysis of Images and Patterns*, pages 838–843. Springer.
- [Sturm et al., 2011] Sturm, P., Ramalingam, S., Tardif, J., Gasparini, S., and Barreto, J. (2011). Camera Models and Fundamental Concepts Used in Geometric Computer Vision. *Foundations and Trends in Computer Graphics and Vision*, 6(1–2):1–183.
- [Su et al., 2007] Su, J., Chung, R., and Jin, L. (2007). Homography-based partitioning of curved surface for stereo correspondence establishment. *Pattern Recognition Letters*, 28(12):1459–1471.
- [Sugimoto and Okutomi, 2007] Sugimoto, S. and Okutomi, M. (2007). A direct and efficient method for piecewise-planar surface reconstruction from stereo images. In *Computer Vision and Pattern Recognition, 2007. CVPR’07. IEEE Conference on*, pages 1–8. IEEE.

- [Tachakra, 2001] Tachakra, S. (2001). Depth perception in telemedical consultations. *Telemedicine Journal and E-Health*, 7(2):77–85.
- [Tebaldini et al., 2008] Tebaldini, S., Marcon, M., Sarti, A., and Tubaro, S. (2008). Uncalibrated view synthesis from relative affine structure based on planes parallelism. *Proceedings - International Conference on Image Processing ICIP 08*, pages 317–320.
- [Tournaire and Paparoditis, 2009] Tournaire, O. and Paparoditis, N. (2009). A geometric stochastic approach based on marked point processes for road mark. *ISPRS Journal of Photogrammetry and Remote Sensing*, 64:621–631.
- [Trefethen and Bau, 1997] Trefethen, L. and Bau, D. (1997). *Numerical linear algebra*. Society for Industrial Mathematics.
- [Triggs, 1997] Triggs, B. (1997). Autocalibration and the absolute quadric. pages 609–615. Published by the IEEE Computer Society.
- [Triggs, 1998] Triggs, B. (1998). Autocalibration from planar scenes. *Computer Vision—ECCV’98*, pages 89–105.
- [Tsai, 1987] Tsai, R. (1987). A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of robotics and Automation*, 3(4):323–344.
- [van den Heuvel, 1999] van den Heuvel, F. (1999). Estimation of interior orientation parameters from constraints on line measurements in a single image. *International archives of photogrammetry and remote sensing*, 32(5):81–88.
- [Vergauwen, 2006] Vergauwen, M. (2006). Harnessing structure-and-motion. *Ph.D. thesis, Katholieke Universiteit Leuven, Belgium*.
- [Vincent and Laganier, 2001] Vincent, E. and Laganier, R. (2001). Detecting planar homographies in an image pair. In *Image and Signal Processing and Analysis, 2001. ISPA 2001. Proceedings of the 2nd International Symposium on*, pages 182–187. IEEE.
- [Wang et al., 2008] Wang, G., Jonathan Wu, Q., and Zhang, W. (2008). Kruppa equation based camera calibration from homography induced by remote plane. *Pattern Recognition Letters*, 29(16):2137–2144.
- [Wang et al., 2005] Wang, G., Tsui, H., and Hu, Z. (2005). Reconstruction of structured scenes from two uncalibrated images. *Pattern Recognition Letters*, 26(2):207–220.

- [Welch et al., 2011] Welch, G., Sonnenwald, D., Fuchs, H., Cairns, B., Mayer-Patel, K., Yang, R., State, A., Towles, H., Ilie, A., Krishnan, S., et al. (2011). Remote 3D medical consultation. *Virtual Realities*, pages 139–159.
- [Xu et al., 2006] Xu, D., Li, Y. F., Shen, Y., and Tan, M. (2006). New pose-detection method for self-calibrated cameras based on parallel lines and its application in visual control system. *IEEE Transactions On Systems, Man, And Cybernetics-Part B: Cybernetics*, 36(5):1104–17.
- [Zeller et al., 1996] Zeller, C., Faugeras, O., and national de recherche en informatique et en automatique, I. (1996). Camera self-calibration from video sequences: the kruppa equations revisited. *Rapport De Recherche-Institut National De Recherche En Informatique Et En Automatique*.
- [Zeng et al., 2008] Zeng, H., Deng, X., and Hu, Z. (2008). A new normalized method on line-based homography estimation. *Pattern Recognition Letters*, 29(9):1236–1244.
- [Zhu et al., 2009] Zhu, Q., Wang, H., and Tian, W. (2009). A practical new approach to 3D scene recovery. *Signal Processing*, 89(11):2152–2158.
- [Zisserman et al., 1998] Zisserman, A., Liebowitz, D., and Armstrong, M. (1998). Resolving ambiguities in auto-calibration. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 356(1740):1193.

VITA AUCTORIS

Amirhasan received his bachelor degree in Computer Engineering, where he completed the design and implementation of both hardware and software of a 12Mhz digital PC oscilloscope, with 2D graphical user interface, as his graduation project. He later attended the Eastern Mediterranean University where he studied distributed systems and parallel computing. He received a degree in master's of Computer Engineering from the Department of Computer Engineering. He later joined the University of Windsor to pursue his PhD in Computer Science, where he developed and implemented 2D and 3D computer vision algorithms at the computer vision lab. He received his PhD in 2011.