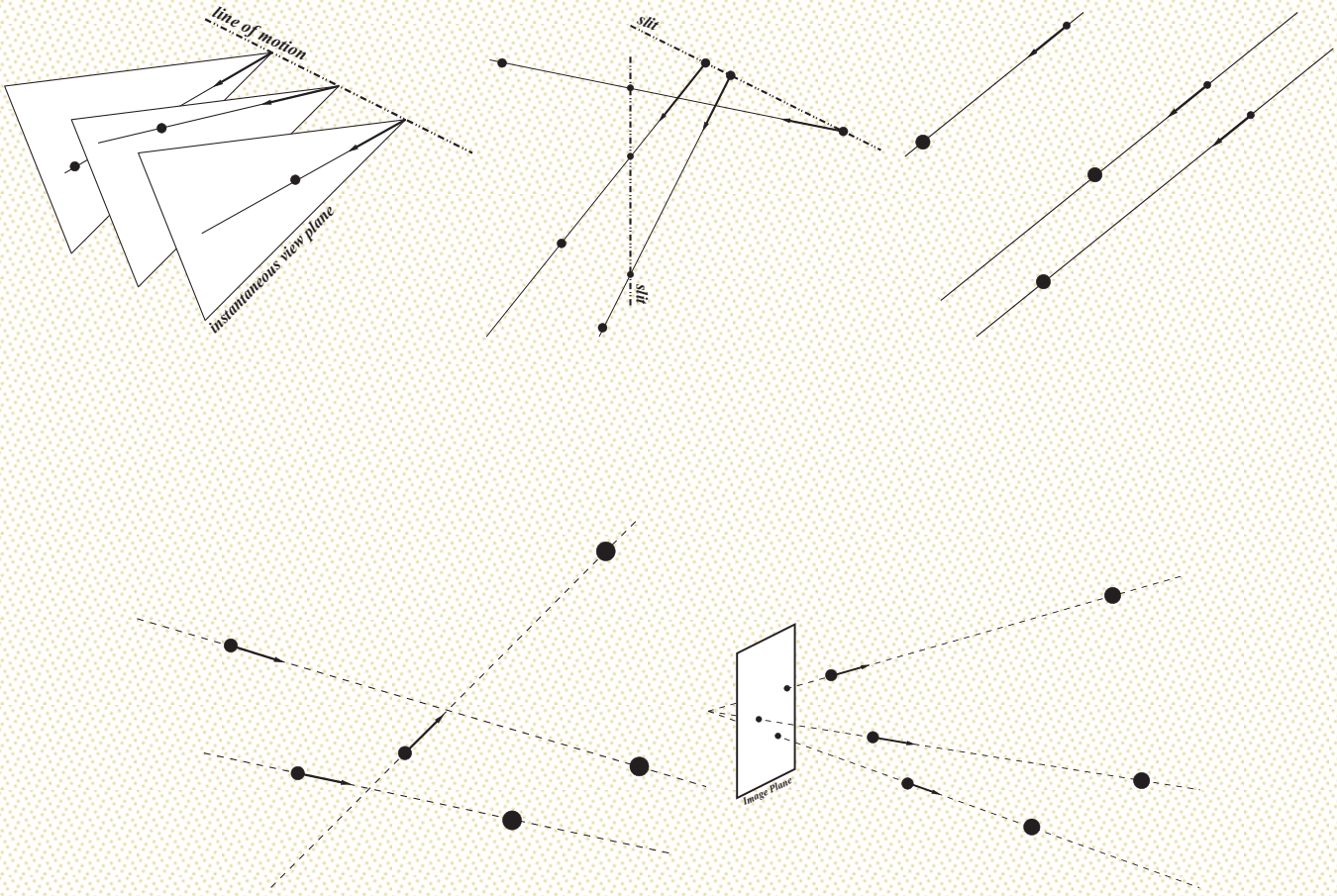


Pedro Daniel dos Santos Miraldo

GENERAL CAMERA MODELS:
Calibration and Pose

Doutoramento em Engenharia Electrotécnica
Especialidade de Informática
Abril 2013

· U C ·

Universidade de Coimbra

Dissertação submetida à Faculdade de Ciências e Tecnologia da Universidade de Coimbra para a satisfação parcial dos requisitos para obtenção do grau de Doutor.
Curso: Doutoramento em Engenharia Electrotécnica, Especialidade de Informática.

Tese efectuada sob a orientação do Professor Doutor Helder de Jesus Araújo, Professor Catedrático da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Júri da prova de doutoramento:

Doutor Jorge Manuel Moreira de Campos Pereira Batista – **Presidente**
Professor Associado da Faculdade de Ciências e Tecnologia da Universidade de Coimbra

Doutor Peter Sturm
Directeur de Recherche, Institut National de Recherche en Informatique et Automatique – France

Doutor José Alberto Rosado Santos Victor
Professor Catedrático do Instituto Superior Técnico da Universidade de Lisboa

Doutor Aurélio Joaquim de Castro Campilho
Professor Catedrático da Faculdade de Engenharia da Universidade do Porto

Doutor Pedro Manuel Urbano de Almeida Lima
Professor Associado com Agregação do Instituto Superior Técnico da Universidade de Lisboa

Doutor Helder de Jesus Araújo
Professor Catedrático da Faculdade de Ciências e Tecnologia da Universidade de Coimbra

Doutor João Pedro de Almeida Barreto
Professor Auxiliar da Faculdade de Ciências e Tecnologia da Universidade de Coimbra

---

FCT
Fundação para a Ciência e a Tecnologia
MINISTÉRIO DA EDUCAÇÃO E CIÊNCIA

Esta dissertação foi suportada financeiramente pela Fundação para a Ciência e a Tecnologia com referências SFRH/BD/49054/2008 (bolsa individual de doutoramento), PTDC/EIA-CCO/109120/2008, PTDC/EIA-EIA/122454/2010 e PHC/PESSOA 2011/2012. Gostaria ainda de agradecer ao programa QREN-MaisCentro com referência CENTRO-07-ST24-FEDER-002027.

Quero agradecer a orientação do Prof. Doutor Helder Araújo e as contribuições do Prof. Doutor João Filipe Queiró e Prof. Doutor Nuno Gonçalves. Agradeço também aos colegas Fernando Paulino, Rui Melo e Diogo Ferreira pela sua contribuição na aquisição de dados.

Quero agradecer à minha família e a todas as pessoas que me apoiaram.

Universidade de Coimbra
Faculdade de Ciências e Tecnologia
Departamento de Engenharia Electrotécnica e de Computadores
3030-290 Coimbra, Portugal

General Camera Models:

Calibration and Pose

Pedro Daniel dos Santos Miraldo
Coimbra, Abril 2013

Resumo

Nesta dissertação estudamos a calibração e pose de modelos genéricos de câmaras. Os modelos genéricos de câmaras podem ser usados para representar qualquer tipo de câmara. Os métodos anteriores baseavam-se em modelos discretos. A cada pixel na imagem é associada uma recta genérica no mundo.

Nesta dissertação, alteramos o modelo de câmaras genéricas. A versão proposta simplifica substancialmente o modelo de câmara e a sua calibração. O modelo proposto é obtido modificando o modelo geral de câmaras, possibilitando a interpolação entre píxeis vizinhos (admitindo com isso um aumento na resolução – devido à sua natureza continua) e uma representação mais compacta. Usando o modelo de câmara apresentado, propomos um processo de calibração que só requer um ponto 3D para cada pixel (os métodos anteriores necessitam de pelo menos dois pontos 3D para cada pixel na imagem). O método proposto também não necessita da calibração de todos os píxeis pertencentes à imagem. Por isso e comparando com o estado-da-arte, o processo de calibração proposto é significativamente mais simples. Com base nos resultados experimentais, concluímos que o método dá bons resultados e é facilmente implementado, especialmente quando comparamos com os métodos anteriores.

Para a estimação da pose consideramos duas aproximações: soluções mínimas e não mínimas. Os métodos anteriores para a estimação de soluções mínimas eram baseados em propriedades geométricas, por exemplo na preservação da distância entre pontos. Nesta dissertação, propomos uma solução alternativa baseada em propriedades algébricas. Para isso, representamos a pose por uma matriz 3×3*. A pose é estimada usando a relação de incidência entre rectas e pontos 3D e as propriedades da matriz. Comparado com o estado-da-arte, a maior vantagem do método proposto é a sua robustez a configurações críticas.*

As soluções mínimas são importantes especialmente devido à sua rapidez e à forma intuitiva como representam o problema. No entanto, estas soluções sofrem muito com o ruído. Com o objectivo de obter uma solução robusta, propomos também uma solução não mínima para o problema da pose usando

modelos de câmaras genéricos. Em vez de usarmos a aproximação clássica que requer coordenadas 3D de pontos e as suas imagens como dados, nesta dissertação desenvolvemos um método que usa coordenadas 3D de rectas e as suas respectivas imagens. A maior vantagem do método proposto é que não necessita da determinação do mapeamento de pontos no mundo para pontos da imagem, o que simplifica bastante a aquisição de dados. Para isso, propomos uma solução não iterativa (solução fechada), e uma solução iterativa (não linear). Com base nos resultados experimentais, concluímos que o método dá bons resultados (especialmente o método iterativo).

Abstract

In this dissertation we address the calibration and pose of general camera models. Generic imaging models can be used to represent any camera. Current generic models are discrete and define a mapping between each pixel in the image and an unconstrained straight line in 3D space.

In this thesis, we change the generic camera model allowing the simplification of the calibration procedure. The proposed model is obtained by modifying the general imaging model using radial basis functions to interpolate image coordinates and 3D lines, thereby allowing both an increase in resolution (due to their continuous nature) and a more compact representation. Using the proposed model, we also develop a calibration procedure that only requires that a 3D point be matched to each pixel (previous methods required at least two 3D points for each image pixel). In addition not all the pixels need to be calibrated. As a result the complexity of the procedure is significantly decreased. In terms of experimental results, we concluded that our method gives good results and that it can be easily used in practice, specially when compared with the state-of-the-art method.

For the estimation of the pose for general imaging systems, we consider two approaches: minimal and non-minimal solutions. Previous solutions for the minimal absolute pose problem were based on geometric properties, such as the preservation of distance between points. In this dissertation, we propose a new parameterization of the problem using an algebraic-based approach. We represent the pose by a 3×3 *matrix. Using both the algebraic relationship between three incident 3D points and straight lines and the underlying constraints of the pose matrix, the pose can be computed. The main contribution of the proposed method is the robustness to critical configurations.*

Minimal solutions are important because of their speed, because they allow insights and a better understanding of the problem. However, they are not robust to noise. Thus and in addition, we address the non-minimal problem of pose estimation for general camera models. Instead of the classic approach where the data-set is made up by associations between known coordinates of

3D points and their images, we propose a solution based on the knowledge of the coordinates of 3D straight lines (expressed in the world coordinate system) and their respective images. The use of lines simplifies the correspondence problem when compared to the use of world and image points – this is the main advantage of the proposed approach. In this thesis we propose a solution using both a non-iterative analytical solution and an iterative non-linear solution. Based on the experiments, we see that the proposed method gives good results.

Contents

I Presentation **1**

1 Introduction **2**
1.1 Motivation 2
1.2 Contributions 4

2 State–of–the–Art Methods **5**
2.1 General Camera Model and its Calibration 5
2.2 Minimal Solutions to the Pose Problem using General Camera Models 6
2.3 Pose Estimation for General Camera Models Using Lines . . . 9

3 Mathematical Properties **10**
3.1 Notations 10
3.2 Background 12
3.2.1 Plücker Coordinates 12
3.2.2 Distance Between Lines 13
3.2.3 Interpolation Using Radial Basis Functions 13
3.3 Mathematical Derivations 15
3.3.1 Affine Transformation to the Line Space 15
3.3.2 Geometric Distance Between a 3D Line and a 3D Point 17

II Smooth Camera Model and its Calibration **19**

4 Introduction **20**
4.1 Our Approach 20

5 Proposed Model and Calibration **23**
5.1 Smooth Camera Model 23
5.2 Formalization 24
5.2.1 Vector–Valued Function 24

5.2.2 Linear Point–based Calibration 26
5.2.3 Data Normalization 30

6 Experiments **33**
6.1 Results with Synthetic Data Sets 33
6.1.1 Evaluation Results Using Smooth Camera Models . . . 33
6.1.2 Results Using Non–Smooth Camera Models 37
6.1.3 Experimental Results Using 3D Data from a Single Surface 38
6.2 Results with Data sets of Real Images 40
6.3 Using a Calibrated Perspective Camera to Acquire a Data-set with Real Data 43
6.3.1 Removing Distortion: Results for a Spherical Catadioptric System 46

7 Discussion **49**

III Minimal Pose Problem for General Camera Models **51**

8 Introduction **52**
8.1 Our Approaches 53

9 Formalization – Method 1 **55**
9.1 Decomposition of the *Homography* Matrix 56
9.2 Proposed Approaches 58
9.2.1 Minimal Absolute Pose Problem for General Cameras . 59
9.2.2 Minimal Absolute Pose Problem for Central Cameras . 62
9.3 Algorithm Outline 64

10 Formalization – Method 2 **67**
10.1 Proposed Approach 68
10.1.1 Minimal Absolute Pose for General Camera Models . . 69
10.1.2 Minimal Absolute Pose for Central Camera 70
10.1.3 Recovery of the Pose 73
10.2 Algorithms 74

11 Experiments **76**
11.1 Numerical Errors 78
11.2 Number of Solutions 80
11.3 Computation Time 80

11.4 Experiments with Noise 82
11.5 Critical Configurations 82

12 Discussion **85**
12.1 Analysis for the General Case 85
12.2 Analysis for the Central Case 88

IV Pose Estimation Using Lines **89**

13 Introduction **90**
13.1 Our Approach 91
13.2 Notations 92
13.2.1 Intersection of Lines 92
13.2.2 Rigid Transformation Applied to Lines 92

14 Formalization **93**
14.1 Analytical Solution for the Pose 93
14.2 Degenerate Cases 96
14.3 Non-Linear Optimization 97
14.4 Algorithm 99

15 Experiments **100**
15.1 Synthetic Data 100
15.2 Convergence of the Non-Linear Method 104
15.3 Experiments with Real Images 105

16 Discussion **107**

V Conclusions **109**

17 Conclusions **110**
17.1 Model and Calibration of Smooth Camera Models 110
17.2 Minimal Absolute Pose Problem 111
17.3 Pose Using Lines 111

VI Appendices **113**

A Unique Solution for the Estimation of the Plücker Coordinates Using Radial Basis Functions **114**

A.1 More on Radial Basis Functions . . . 114
A.2 Introduction . . . 117
A.3 *Rank* of matrix $\mathbf{M}$ . . . 117
A.3.1 Proof that matrix M can have rank $6P + 17$ . . . 119
A.3.2 The set $\lambda\left(\mathbf{K}^{-1}\mathbf{F}\left(\mathbf{D}_{\mathbf{1,2}}\right)^{-1}\mathbf{F}\left(\mathbf{T}_{\mathbf{1,2}}\right)\right)$ . . . 126
A.4 Conclusions . . . 127
A.5 Some Matrix Results . . . 128
A.5.1 Rank of $\mathbf{D}_1 - \mathbf{L}\mathbf{D}_2\mathbf{L}^{-1}$ . . . 128
A.5.2 Inverse of Matrices . . . 129
A.5.3 Eigenvector Matrices . . . 130
A.5.4 Intersection Subspace . . . 131

B Analytical Solution for the Null-Space of Both N and M Matrices 133

C Minimal Solvers 135
C.1 Solver for the General Case . . . 135
C.2 Solver for Central Camera Models . . . 136

D Proof of Theorems and Propositions 138
D.1 *proof* of the Theorem 9.1 . . . 138
D.2 *Proof* of the Theorem 9.2 . . . 140
D.3 *Proof* of the Proposition 9.2 . . . 142

List of Figures

2.1 A unified camera model only assumes that an association between image pixels and rays in the world space exists. 6

2.2 Representation of the pose estimation problem, considering that we have a calibrated generalized camera. Three 3D points are imaged through three rays. Since the generic camera model is calibrated, the 3D ray coordinates are known in the camera coordinate system. Pose estimation consists therefore on the estimation of the rotation and translation that define the transformation from the 3D world coordinate system into the camera coordinate system such that the incidence relationship between points and lines is verified. 8

3.1 In this figure we display the representation of a line $\mathbf{l}^{(1)}$ in the coordinate system centered at point $\mathbf{p}$. The plane $\boldsymbol{\pi}$ is defined by $\mathbf{l}^{(2)}$ and the origin of the coordinate system. 18

4.1 In this figure we display the result of the application of the calibration method to the minimal case according to the theoretical constraints (see Section 5.2.2). In the minimal case there are six points in the the image and their correspondent points in the world. 22

6.1 Evaluation of the average line distance error, Equation (3.12) – Section 3.2.2, as a function of the number of *control points*, for non–central catadioptric systems with a quadric mirror, for a crossed–slits camera model and for a model including refraction (a camera looking through a volume of water contained between two parallel planes) (a), (d) and (g) respectively. For each camera model we evaluate the distance error defined in Equation (3.12), for *gaussian* and *multi–quadrics radial basis functions* and the errors are shown in Figures (b)-(c), (e)-(f) and (h)-(i) respectively. 34

6.2 In this figure we show results of tests performed with the three previously mentioned camera models, Figure 6.1, with noise added. We set the number of control points to 60 and vary the variance of the noise added to the coordinates of the 3D points used in the calibration procedure. The standard deviation of the noise is proportional to the smallest distance among all the world points of the calibration data set. (a) is the distance error in the line space for *gaussian* RBF and (b) for *multi–quadrics*. To eliminate some errors, we can use more points than the ones needed for the minimal solution. We consider the noise such that $\alpha = 0.10$ – Equation (6.2); and we vary the β, where $N = \beta P$. The results are shown in Figures (c) and (d) for *gaussian* and *multi–quadrics* respectively. 36

6.3 This figure shows results for a non–smooth camera model synthetically generated, and made up by combining four images obtained from four different perspective cameras as shown in (a). In figure (b) errors corresponding to the distances between the 3D lines are shown. 37

6.4 For 3D calibration points on a spherical surface, Figure (a), displays the image points (black squares) and control points (red dots) used in the calibration process. Their correspondent world points are shown in Figure (b) (black dots). Figure (c) shows the results, where the ground–truth lines are shown in black and the estimated lines in red. The same experiments were performed with the calibration points belonging to a "step" surface and the results are presented in Figures (d) (e) and (f). In both cases we use 40 *control points* and 360 data points. 39

6.5 This figure shows the calibration of a catadioptric camera model, where the 3D points used in the calibration belong to a cylindrical surface. Figure (a), displays the image points (black dots) and control points (red dots) used in the calibration process. Their correspondent world points are shown in Figures (b) and (c) (black dots). Figure (d) shows the results, where the ground–truth lines are shown in black and the estimated lines in red. We use 40 *control points* and 360 data points. 40

6.6 Results with real images, obtained using the method described for three types of camera models: perspective camera (a), catadioptric system with spherical mirror (d) and catadioptric system with two planar mirrors (g). Yellow points in (b), (e) and (h) are the set of image coordinates $\{\mathbf{x}_i\}$ and yellow points in (c), (f) and (i) are the corresponding set of world points $\{\mathbf{p}_i\}$, used in the calibration method. Red squares in the $2D$ plot and the red rays in the 3D plot are the a subset of $\left\{\mathbf{y}_i \mapsto \breve{\mathbf{l}}_i\right\}$ obtained with *multi–quadrics* RBF. Green points in (b), (e) and (h), are the corresponding set of *control points* used in the calibration. 41

6.7 In this figure we show five examples of images taken from the chess board used to get the calibration data sets. The first row of images corresponds to the perpective camera; the second row corresponds to the non-central catadioptric camera that uses a spherical mirror; and the last row corresponds to the non-central camera formed by two planar mirrors. 42

6.8 In Figure (a), we display the set-up used to acquire the images of the data set for the case where the goal is to calibrate the camera model that views a scene through a water-filled glass tank. In Figure (d), we show the perspective camera and the spherical mirror that form the non-central catadioptric camera. Examples of images acquired by both non-central imaging system are shown in Figures (b) and (e). A planar chess board was used. The data set is made up by the coordinates of the images of the points corresponding to the intersections of the chess board edges (seen through the tank). Examples of the calibration results are shown in Figures (c) and (f). In yellow we show all the 3D points used in the calibration procedure. In blue we show the ground truth 3D points on which the estimated lines must be incident. The blue 3D points correspond to the chess board edges of the images Figures (b) and (e) respectively. The estimated lines (according to the method proposed in the thesis) are shown in red. The units in both Figures (c) and (c) are millimeters. 44

6.9 In this figure we show a set of examples of images that we used to generate the data set. In Figure (a) we show the case where the goal is to calibrate a refraction camera. In Figure (b) we show images used for the catadioptric case. 45

6.10 In this experiment we show two examples of results obtaining an undistorted image from a non–central catadioptric camera, using the knowledge of the scene structure (in this case a plane). Figures (a) and (d) show the original images. Figures (b) and (e) show the images of the same plane after removal of the distortion. In addition, we show the disparities between estimated and ground–truth image pixels of the corners of the checkerboard (Figures (c)) and (f)) 47

11.1 In this figure we show the numerical distribution of the errors and the distribution of the number of solutions for general (non-central) camera models. The proposed method is compared against the state–of–the–art algorithms proposed by Nistér and Stewénius at [Nis04a, NS07] and by Chen and Chang at [CC02, CC04a]. For our method, we consider both the case where the null-space of $\mathbf{N}$ is computed using *singular value decomposition*, denoted as `Our - NSVD`; and the case where we use the analytical solution developed in Appendix B, denoted as `Our - NCF`. 78

11.2 In this figure we show the numerical distribution of the errors and the distribution of the number of solutions for central camera models. The proposed method is compared against the state–of–the–art algorithm proposed by Kneip *et al.* [KSS11] and the variation Nistér and Stewénius at [Nis04a, NS07] for central cameras (can be solved in close–form). For the algorithm proposed in this article: we consider the case when the null-space of the matrix $\mathbf{M}$ is computed using the closed-form solution – Apendix B, denoted as `Our - Closed-Form`; and the case when the null-space is computed using the *singular value decomposition*, denoted as `Our - MSVD`. In addition, we consider our general case – used in the experiments of Figure 11.1, denoted as `Our - General`. 79

11.3 In this figure we show the analysis of the error for data with noise. We consider the `Noise` variable as the standard deviation of three random noise vector $\mathbf{n}_i^{(\mathcal{W})}$ and compute the pose using the data-set $\left\{\mathbf{l}_i^{(\mathcal{C})}, \breve{\mathbf{p}}_i^{(\mathcal{W})}\right\}$, where $\breve{\mathbf{p}}_i^{(\mathcal{W})} = \mathbf{p}_i^{(\mathcal{W})} + \mathbf{n}_i^{(\mathcal{W})}$. For each value of the `Noise`, we use 10^3 trials and the errors are computed as the median of all the errors. As it can be seen from the figures, the four plots overlap, which means that the errors are modeled only by the noise. 82

11.4 In this figure we evaluate the stability and robustness of the proposed algorithms. We consider the algorithms that we used in previous tests, for general camera models. For that, we consider three critical cases: *orthogonal* configuration (a) and (d); *pushbroom* cameras (b) and (e); and *X-Slit* cameras (c) and (f). To analyze the stability and robustness by computing the errors in rotation and translation parameters that define the pose as a function of a `Distance from Critical Case` – see the Section 11.5. Note that the errors are shown in a 10-base representation. The errors are computed as a median of 10^3 valid trials. Close to critical cases it is possible that some algorithms fail to compute a solution. As a result, and for each algorithm, we show the number of failures that we get until 10^3 valid trials are obtained. 84

14.1 In this figure we show an example of the intersection between a world line $\mathcal{G}_i$ and four "projection lines" $\mathcal{H}_{j,i}$, for $j = 1, \ldots, 4$. 94

15.1 In this figure we show two examples of the generation of the 3D lines and "projection lines" in the camera coordinate system $\mathcal{H}_{j,i}^{(\mathcal{C})}$ – blue lines; and $\mathcal{G}_i^{(\mathcal{C})}$ – red lines. In green we show the intersection points. In Figure (a), we show an example of the general case. In the case of Figure (b) the directions of the "projection lines" ensure that they pass close to each other – close to central. 101
15.2 In this figure we show the evaluation of the proposed method using synthetic data. We consider the error for the six parameters that define the pose: three rotation angles – radians; and three for the translation. In Figure (a) we show the results as a function of the `Noise` variable. In Figure (b) we show the results for different number of lines. 102
15.3 In Figure (a) we show the evaluation of the algorithm as a function of the noise, for `Deviation from Central Case` equals to twenty. In Figure (b) we study the variation of the error as a function of the `Deviation from Central Case`. For each value of the evaluation variable, we consider 10^3 trials as described in the text. 103
15.4 In this figure we evaluate the convergence ratio of the non-linear method proposed in Section 14.3. For this purpose, we vary the number of known 3D straight lines. In addition, we also consider different values for `Deviation from Central Case` variable. Note that we consider the case of central camera – `Deviation from Central Case` equal to zero. 104
15.5 In Figure (a) we show five examples of images taken from the non-central catadioptric camera. The image curves marked in the images correspond to the data set used for the computation of the pose. In Figure (b) we test the proposed method using an application of augmented reality. The created objects are shown in Figure 15.6. 105
15.6 In this figure we show the reconstruction of the motion obtained from a sequence of images taken from the non-central catadioptric camera – formed with a perspective camera and a spherical mirror. In the graphic, we show the recovered position of the perspective camera. 106

List of Tables

2.1 This table represents a raxel array of parameters, according to the paper [GN01, GN05]. For each image pixel ($\mathbf{u}$), we have an array of parameters that represents: a 3D line ($\mathbf{p}, \mathbf{d}$); major and minor standard deviation ($\sigma_M(r), \sigma_m(r)$); elliptical orientation ($\Psi(r)$); radiometric response ($g(e)$); and fall-off factor (h). 7

6.1 Results for the experiments, using the IR tracker [Nor09]. . . . 43
6.2 Results for the experiments, using the Perspective Camera to Acquire a Data-set with Real Data. 46

9.1 In this table we show the coefficient parameters of the polynomial equations derived in Equation (9.28) 61
9.2 In this table we show the coefficient parameters of the quadratic equations of Equation (9.34). 63

10.1 In this table, we show the coefficient parameters $\kappa_n^{(i,j)}$ of the polynomial equations $g_{i,j}(a, b, c)$ – Equation (10.22). 71
10.2 In this table, we show the coefficient parameters $\mu_n^{(i,j)}$ of the polynomial equations $g_i(a, b, c)$ – Equation (10.30). 72

12.1 In this table we summarize the analysis of the main computational effort required in the main computation steps of the proposed algorithms. K – represents the number of valid solutions given by the algorithms. The * symbol means that it corresponds to the computational effort of the minimal solver that uses the *companion matrix*. We note that for our method, other minimal solvers can be used. 86

Part I: Presentation

Contents

1 Introduction **2**
- 1.1 Motivation . . . 2
- 1.2 Contributions . . . 4

2 State–of–the–Art Methods **5**
- 2.1 General Camera Model and its Calibration . . . 5
- 2.2 Minimal Solutions to the Pose Problem using General Camera Models . . . 6
- 2.3 Pose Estimation for General Camera Models Using Lines . . . 9

3 Mathematical Properties **10**
- 3.1 Notations . . . 10
- 3.2 Background . . . 12
  - 3.2.1 Plücker Coordinates . . . 12
  - 3.2.2 Distance Between Lines . . . 13
  - 3.2.3 Interpolation Using Radial Basis Functions . . . 13
- 3.3 Mathematical Derivations . . . 15
  - 3.3.1 Affine Transformation to the Line Space . . . 15
  - 3.3.2 Geometric Distance Between a 3D Line and a 3D Point 17

Chapter 1

Introduction

In this chapter, we introduce the framework of generalized camera model and the motivations for the dissertation. We also describe the contributions of the dissertation.

1.1 Motivation

In terms of geometry, a camera is a mapping between the 3D world space and a 2D image plane. In computer vision, we consider that the 2D image space is discrete and we call its elements pixels.

Most cameras can be modeled by a perspective projection – the pinhole camera model [HZ00, MSKS04]. The set of pixels are mapped from 3D straight lines in the world, that pass through a single point – usually called the single view point.

This ideal pinhole camera model is well studied in the literature. The mapping of 3D world points to pixels can be described by an eleven degrees of freedom matrix – 3×4 matrix. Due to its geometric constraints, the calibration procedure for this models is well known and easy to implement [Tsa86, Zha00, HZ00].

Other camera models that verify the single view point constraint – central camera models, but do not satisfy the pinhole model have been defined, such as the special cases of catadioptric camera models [BN99, GSB09] – which are formed by the combination of a perspective camera and a quadric mirror. Moreover, central camera models with non–parametric association of image pixels to 3D straight lines (general central camera model) and their respective calibration have also been studied [HK05, NSG05, RSLSK10].

In the last few years, cameras whose projecting rays do not satisfy the constraint of intersecting at a single effective viewpoint started to be used,

due essentially to the large fields of view that can be obtained. These cameras are called non–central and in many cases are obtained by a generic combination of a perspective camera and a curved mirror–the non-central catadioptric cameras [Gon08, ATR10, SGN06, MP04b]. They are used in several applications ranging from robotics to visualization.

Many other non–central camera models have also been developed such as linear cameras [Pon09, GH97, YM04, ZFPW03], fisheye cameras [HF09, KB06] or camera models that include refractive elements [TSS08, KS08]. All of these systems have, in general, models that are characterized by a small number of parameters. Since the number of parameters is small the calibration procedures are, in general, simple. However, since they are derived using geometric models of the cameras [SRT+11] they lack generality. To deal with these drawbacks, some techniques were developed to calibrate special classes of cameras such as [MP04a, TP12].

The aim of the general camera models is to represent any type of camera – central and non-central cameras. The goal is to represent any imaging device, without enforcing any geometric constraints to the 3D straight lines mapped from the image pixels. As any other camera calibration, the calibration within the framework of the generalized camera models consists on the estimation the parameters that define the mapping between image pixels and the corresponding 3D straight lines.

In most of the cameras, the pixel–line mapping varies smoothly. This means that pixels close to each other will correspond to similar 3D straight lines in the world. In this dissertation we investigate these constraints and the results on the general camera models and their calibration. We define the *Smooth Camera Model* and using the proposed camera model, we developed a new and significantly easier calibration technique.

Usually image space does not change and, as a result, we can define a 3D coordinate system for the image coordinates. On the other hand, a camera is a mobile device and as a consequence we can not define a fixed global coordinate system to represent the lines mapped into the image points. Therefore we define a 3D reference coordinate system associated to the camera to represent the 3D lines mapped into the image pixels. As a consequence, to estimate the coordinates of 3D entities represented in a different coordinate system, we need to estimate a rigid transformation mapping the camera coordinate system into the other reference coordinate system, which we call the absolute pose problem. This problem can be divided into two subproblems: minimal and non–minimal cases.

Absolute pose is important for many applications that require using the 3D entities imaged by a camera. The analysis of solutions using minimal data is important since they allow insights and a better understanding of the

problem. On the other hand, non–minimal solutions are used to estimate the pose for more point correspondences than the minimal, when we need a robust solution. The advantages of minimal solutions are in terms of computation speed and their applications in the hypothesis–and–test estimation methods such as RANSAC [FB81, Nis03].

In this dissertation, we studied both the minimal and non-minimal pose estimation for general imaging devices. The minimal pose consists in the estimation of the transformation between the world and camera coordinate system, using a single image of three 3D points in the scene. For non-minimal cases, we propose a method based on known coordinates of 3D lines in the world coordinate system.

1.2 Contributions

The main contributions of the dissertation are described in Parts II, III and IV:

- In Part II, we propose the *smooth camera model* and its calibration. This part of the dissertation was published in the following articles [MA10, MAQ11, MA13];
- In Part III, we study the minimal generalized pose problem. We propose a novel and robust solution for this problem;
- In Part IV, we propose a solution for the absolute pose (under the framework of generalized camera models), using known coordinates of 3D straight lines, in the world coordinate system.

Chapter 2

State–of–the–Art Methods

In this chapter we describe the state–of–the–art methods that are related to the subjects studied in the dissertation. In Section 2.1, we describe the previous methods for the general camera models and its calibration. In Section 2.2, we preset the state–of–the–art solutions for the minimal absolute pose problem, under the framework of generalized camera models. To conclude, in Section 2.3, we describe the state–of–the–art method for the non-minimal pose problem for general camera models.

2.1 General Camera Model and its Calibration

The concept of generalized camera model was proposed by Grossberg and Nayar [GN01, GN05]. They defined a non-parametric discrete imaging model that can represent any type of camera, central or non-central – the general camera model. Differently from the usual parametric camera models, this camera model consists in the individual mapping between pixels and rays in 3D space, Figure 2.1. To each pixel, a set of parameters called *raxel* is associated – Table 2.1. The set of all *raxels* (representing all pixels) constitutes the complete general camera model. A *raxel* is a set of parameters including image pixel coordinates, the coordinates of the associated ray in the world (position and direction) and radiometric parameters.

Grossberg and Nayar also propose a method for estimating the parameters of the *general camera model*. Their approach requires the acquisition of, at least, two images of a calibration object with known structure and also requires the knowledge of the object motion between the images. Basically, it requires two or more world points for the same image pixel.

Sturm and Ramalingam at [SR04] and Ramalingam *et al.* at [RSL05]

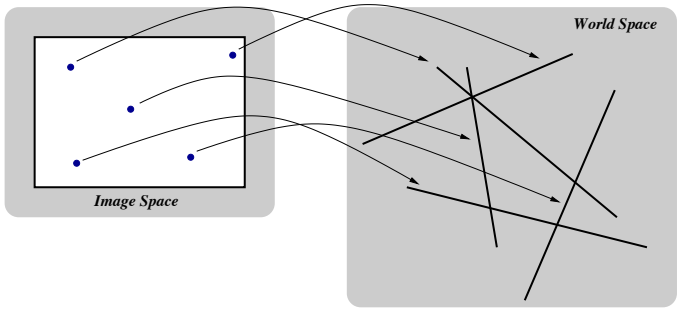


Figure 2.1: *A unified camera model only assumes that an association between image pixels and rays in the world space exists.*

proposed a calibration method based on the discrete imaging model, suggested by Grossberg and Nayar. However, they excluded from their model the radiometric entities of the *raxel*. Their method assumes that the camera is fully described by the coordinates of rays and the mapping between rays and pixels.

Instead of using two images, Sturm and Ramalingam methods require three images of the calibration object, acquired from arbitrary and unknown viewing positions. If three points of the calibration object are seen for the same pixel, the collinearity constraint allows the computation of the motion between the images of the calibration object and, as a result, it allows the estimation of the direction of the ray in 3D space.

2.2 Minimal Solutions to the Pose Problem using General Camera Models

As described in the motivation Section 1.1, the minimal absolute pose problem consists in the estimation of the transformation between the world and camera coordinates system, using the image of three 3D points. We consider that the camera is calibrated using general camera models. As a result, the image of the three 3D point will be mapped into straight lines in the world. The pose is then given by estimating of the rotation and translation parameters that define the aim transformation, using the incident relation between three world points and three 3D lines – Figure 2.2.

We do not address the relative pose problem where two or more images of the scene are used. In general estimates based on minimal data provide multiple solutions. In this dissertation we develop a new method for the pose estimation using the minimal number of three 3D points.

Pixel	*3D Line*		*Point Spread*			*Response*	*Fall-off*
$\mathbf{u}$	$\mathbf{p}$	$\mathbf{d}$	$\sigma_M(r)$	$\sigma_m(r)$	$\Psi(r)$	$g(e)$	h

Table 2.1: *This table represents a raxel array of parameters, according to the paper [GN01, GN05]. For each image pixel ($\mathbf{u}$), we have an array of parameters that represents: a 3D line ($\mathbf{p}, \mathbf{d}$); major and minor standard deviation ($\sigma_M(r), \sigma_m(r)$); elliptical orientation ($\Psi(r)$); radiometric response ($g(e)$); and fall-off factor (h).*

This problem was studied by Nistér & Stewénius at [Nis04a, NS07], by Chen & Chang at [CC02, CC04a] and by Ramalingam *et al.* at [RLS06].

The algorithm proposed by Chen and Chang at [CC02, CC04a] was the first method to estimate the minimal three–point pose problem for general imaging devices. The algorithm is based on the estimation of the roots of an eighth degree polynomial equation. To get this polynomial equation, the data-set is transformed so that lines and points verify a set of constraints. On the new coordinate system an algorithm is used, relating the triangle defined by distances between the three known world points and the three projection lines.

This algorithm estimates the coordinates of the points in the camera coordinate system. Thus and considering the pose described by a rotation and a translation, additional steps are required. The estimation of the rigid transformation based on the matching between the coordinates of the points in the camera and world coordinate system is required – for instance [AHB87, Ume91].

The approach of Ramalingam *et al.* at [RLS06] is similar to the one proposed by Chen and Chang, which is based on the fact that the distances between the world points must be preserved. Both algorithms estimate the world points in the camera coordinate system. They claim that the three distances constraints generate an eighth degree polynomial equation. However, unlike the algorithm proposed by Chen and Chang, they do not give a closed-form solution for the polynomial coefficients. The main steps of both [RLS06] and [CC02, CC04a] methods are the intersection between three axis-aligned cylinders.

Nistér and Stewénius [Nis04a, NS07] proposed an algorithm that is also based on the estimation of the roots of a single variable eighth degree polynomial equation. This algorithm also requires the transformation of the points and lines into world coordinates. The coefficients of the polynomial equation are obtained from the intersection between a ruled quartic surface and a circle. One of the advantages of this algorithm is that it estimates the rotation and translation parameters directly, which will be reflected on the

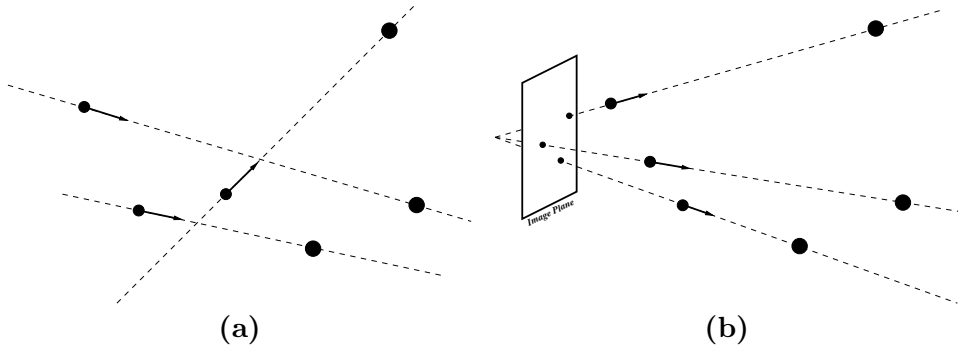


Figure 2.2: *Representation of the pose estimation problem, considering that we have a calibrated generalized camera. Three 3D points are imaged through three rays. Since the generic camera model is calibrated, the 3D ray coordinates are known in the camera coordinate system. Pose estimation consists therefore on the estimation of the rotation and translation that define the transformation from the 3D world coordinate system into the camera coordinate system such that the incidence relationship between points and lines is verified.*

computational effort.

To conclude, we note that the all the state-of-the-art algorithms are based on the determination of an eighth degree polynomial equation, as a function of a single variable. Despite the differences between the approaches, these algorithms require that a set of rigid transformations of the data-set be performed.

As a comparison between the methods, we note that both the method proposed by Ramalingam *et al.* and Chen & Chang are very intuitive and easy to implement. On the other hand, the algorithm proposed by Nistér and Stewénius is quite complex and very difficult to implement. However, the last method gives the pose parameters directly, while Ramalingam *et al.* and Chen & Chang algorithms estimates the world points in the camera coordinate system. These aspects are analysed in Part III.

We also note that, so far, there was no comparison between any state-of-the-art methods.

2.3 Pose Estimation for General Camera Models Using Lines

In addition to the minimal solution for the pose, in this thesis we also address the non-minimal case – more than three 3D points and their images are known.

The analysis of solutions using minimal data is important especially because they allow insights and a better understanding of the problem. Other advantage is its computational speed. However, non-minimal solutions are less sensible to noise.

A common approach for the estimation of the pose is to use a set of correspondences between 3D points in the world coordinate system and pixels in the image. Using the general imaging models – Section 2.1, the set of image pixels are associated to unconstrained 3D lines in the camera coordinate system. The pose is thus computed using the incidence relation between 3D points and correspondent lines.

For general camera models, there are not many research effort. As far as we know, just Chen & Chang at [CC02, CC04a] and Schweighofer & Pinz at [SP08] addressed this problem. Chen and Chang method first estimates an initial solution based on nominal data and then refine the solution using an iterative method. On the other hand, the method proposed by Schweighofer and Pinz computes the pose globally – iterative solutions.

State–of–the–art methods , estimate the pose using 3D points whose coordinates in world reference system are known, and their corresponding pixels. The determination of point correspondences (in this case between 3D points and their images) is still a difficult problem and current solutions are error-prone. In this thesis we avoid this procedure by using coordinates of 3D lines defined in the world coordinate system, instead of 3D points.

Chapter 3

Mathematical Properties

This chapter is divided into three sections. In the first section we define the notations used in the dissertation. In Section 3.2, we describe the mathematical background and, in Section 3.3, some mathematical results are derived.

3.1 Notations

Through the thesis, we use the following mathematical notations:

- $\mathbb{R}$ represents the set of real numbers. $\mathbb{R}^n$ and $\mathbb{R}^{n\times m}$ represent the n–dimensional vector space of real numbers and the $n \times m$–dimensional matrix of real numbers respectively;

- $\mathbb{P}^n$ represents the n–dimensional real projective space and $\mathbb{L}^3$ the space of lines in 3D;

- Matrices are represented as bold capital letters (*eg.* $\mathbf{A} \in \mathbb{R}^{n\times m}$, n rows and m columns). Vectors are represented as bold small letters (*eg.* $\mathbf{a} \in \mathbb{R}^n$, n elements). By default, a vector is considered a column. Small letters (*eg.* a) represent one dimensional elements. By default, the jth column vector of $\mathbf{A}$ is specified as $\mathbf{a}_j$. The jth element of a vector $\mathbf{a}$ is written as a_j. The element of $\mathbf{A}$ in the line i and column j is represented as $a_{i,j}$. Regular capital letters (*eg.* A) indicate one dimensional constants;

- $\mathcal{SO}(3)$ represents the special orthogonal group on $\mathbb{R}^3$. For a matrix $\mathbf{R} \in \mathcal{SO}(3)$, we have $\mathbf{R}^T\mathbf{R} = \mathbf{I}$ where $\mathbf{I} \in \mathbb{R}^{3\times 3}$ is the identity matrix;

- We use $\mathbb{R}$ after a vector or matrix to denote that it is represented up to a scale factor;

- For two vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$, the operator $(\cdot)$ represents the inner product of two vectors such that $\mathbf{a} \cdot \mathbf{b} = \mathbf{a}^T\mathbf{b}$. In $\mathbf{a} \times \mathbf{b}$, the operator $(\times)$ represents the cross product of two vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^3$;

- For a vector $\mathbf{a} \in \mathbb{R}^3$, the hat operator $\widehat{\mathbf{a}}$ represents a 3×3 matrix that linearizes the exterior product such that $\widehat{\mathbf{a}}\mathbf{b} \doteq \mathbf{a} \times \mathbf{b}$ and

$$\widehat{\mathbf{a}} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}; \tag{3.1}$$

- $||\mathbf{a}||$ represents the standard 2–norm of a vector $\mathbf{a} \in \mathbb{R}^n$, $||\mathbf{a}|| = \sqrt{\mathbf{a} \cdot \mathbf{a}}$;

- $\text{null}(\mathbf{A})$ denotes the *null-space* of the matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$. $\text{rank}(\mathbf{A})$ represents the dimension of the *column-space* and $\text{nullity}(\mathbf{A})$ represents the dimension of the *null-space*. It is well known from linear algebra properties that for a matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$

  $$\text{rank}(\mathbf{A}) + \text{nullity}(\mathbf{A}) = m. \tag{3.2}$$

  $\det(\mathbf{A})$ denotes the determinant of the matrix;

- Let $\mathbf{U} \in \mathbb{R}^{m \times n}$, $\mathbf{V} \in \mathbb{R}^{k \times l}$ and $\mathbf{C} \in \mathbb{R}^{m \times k}$ be known and $\mathbf{X} \in \mathbb{R}^{n \times l}$ unknown. Using the *Kronecker* product, we can write

  $$\mathbf{U}\mathbf{X}\mathbf{V}^T = \mathbf{C} \Rightarrow (\mathbf{V} \otimes \mathbf{U})\,\text{vec}(\mathbf{X}) = \text{vec}(\mathbf{C}) \tag{3.3}$$

  where $\otimes$ is the *Kronecker* product of $\mathbf{U}$ and $\mathbf{V}$, with $(\mathbf{V} \otimes \mathbf{U}) \in \mathbb{R}^{mk \times nl}$ such that

  $$\mathbf{V} \otimes \mathbf{U} \doteq \begin{bmatrix} v_{1,1}\mathbf{U} & v_{1,2}\mathbf{U} & \cdots & v_{1,l}\mathbf{U} \\ v_{2,1}\mathbf{U} & v_{2,2}\mathbf{U} & \cdots & v_{2,l}\mathbf{U} \\ \vdots & \vdots & \ddots & \vdots \\ v_{k,1}\mathbf{U} & v_{k,2}\mathbf{U} & \cdots & v_{k,l}\mathbf{U} \end{bmatrix}, \tag{3.4}$$

  and $\text{vec}(\mathbf{X})$ is a nl–vector formed by stacking the columns of $\mathbf{X}$;

- A vector in the n–dimensional projective space $\mathbf{a} \in \mathbb{P}^n$ can be represented in homogenous coordinates as $\widetilde{\mathbf{a}} \in \mathbb{R}^{n+1}$, such that $\widetilde{\mathbf{a}}\mathbb{R} = (a_0, a_1, \ldots, a_n)$. The regular coordinates coordinates $\mathbf{a} \in \mathbb{R}^n$ are given by $\mathbf{a} = (a_1/a_0, \ldots, a_n/a_0)$.

3.2 Background

In this section we provide a short description of the mathematical basis necessary for the thesis, namely: *Plucker* coordinates to represent lines – Section 3.2.1; a metric to compute distance between lines – Section 3.2.2; and interpolation using *Radial Basis Functions* – Section 3.2.3.

3.2.1 Plücker Coordinates

Plücker coordinates are a special case of *Grassmann* coordinates [PW01, HZ00]. A *Grassmann* manifold is the set of k dimensional subspaces, in a n dimensional vector space, and it is denoted as $\Lambda^k\mathbb{R}^n$ [PW01]. *Plücker* coordinates can be obtained as a representation of the exterior product to four dimensional vectors $\mathbf{x} \wedge \mathbf{w}$. The result of this operation lies in a six dimensional vector space $\mathbb{R}^6$, that can represent lines in $3D$ space.

Let us consider two points in the world ($\mathbf{x}, \mathbf{w} \in \mathbb{P}^3$ – represented in homogeneous coordinates $\widetilde{\mathbf{x}}, \widetilde{\mathbf{w}} \in \mathbb{R}^4$. Using *Plücker* coordinates, we can represent a $3D$ line, incident on both points, as

$$\mathbf{l}\mathbb{R} \doteq \widetilde{\mathbf{x}} \wedge \widetilde{\mathbf{w}} = (\underbrace{l_{01}, l_{02}, l_{03}}_{\mathbf{d}}, \underbrace{l_{23}, l_{31}, l_{12}}_{\mathbf{m}}) \in \Lambda^2\mathbb{R}^4 \subset \mathbb{R}^6 \tag{3.5}$$

with $l_{ij} = \widetilde{x}_i\widetilde{w}_j - \widetilde{x}_j\widetilde{w}_i$, basis $\boldsymbol{e}_{ij} = \boldsymbol{e}_i \wedge \boldsymbol{e}_j$ ($\boldsymbol{e}_i$ are $\mathbb{R}^4$ basis) and $\mathbf{d}$ and $\mathbf{m}$ are, respectively, the direction and the moment of the line.

Although all elements of the four dimensional exterior product, $\Lambda^2\mathbb{R}^4$, belong to $\mathbb{R}^6$, not all elements of $\mathbb{R}^6$ represent lines in $3D$ space. It can be shown that, Equation (3.5) is the result of a four dimensional space exterior product (and therefore it is a line in $3D$ space), if and only if it belongs to the *Klein* quadric which is the same as to write $\mathbf{d} \cdot \mathbf{m} = 0$.

Plücker coordinates enable the computation of the incidence condition of lines and points in the world. Using the direction and moment vectors, a point $\mathbf{p} \in \mathbb{P}^3$ is incident on a line $\mathbf{l} \in \Lambda^2\mathbb{R}^4$ if

$$\underbrace{\begin{bmatrix} \widehat{\mathbf{p}} & -\mathbf{I} \\ \mathbf{0} & \mathbf{p}^T \end{bmatrix}}_{\mathbf{Q}(\mathbf{p})} \mathbf{l}\mathbb{R} = \mathbf{0} \tag{3.6}$$

where $\mathbf{I}$ is a 3×3 identity matrix. Note that, in this case, the point $\mathbf{p}$ is represented in regular coordinates.

3.2.2 Distance Between Lines

In this section we describe a metric to compute the distance between two lines. We use an approach suggested in [PW01], which uses the stereographic projection of the *Klein* Quadric. In the remaining of this section we briefly describe the method.

Let us consider two non-horizontal lines represented by *Plücker* coordinates – Section 3.2.1, $\mathbf{g}, \mathbf{h} \subset \mathbb{R}^6$, represented as

$$\mathbf{g} = (g_1, g_2, 1, g_4, g_5, g_6) \quad \text{and} \quad \mathbf{h} = (h_1, h_2, 1, h_4, h_5, h_6)\,. \tag{3.7}$$

Note that, from the definition of *Plücker* Equation (3.5), lines can be represented up to a scale factor. For this purpose, we consider $g_3 = h_3 = 1$.

Using the stereographic projection of the *Klein* quadric – [PW01], the lines of Equation (3.7) can be represented in Euclidean 4-space $\breve{\mathbf{g}} = (\breve{g}_1, \breve{g}_2, \breve{g}_3, \breve{g}_4)$ and $\breve{\mathbf{h}} = \left(\breve{h}_1, \breve{h}_2, \breve{h}_3, \breve{h}_4\right)$ such that

$$\breve{g}_1 = -\,g_5, \;\; \breve{g}_2 = g_4, \;\; \breve{g}_3 = g_1 + g_5 \;\text{ and }\; \breve{g}_4 = g_2 - g_4; \tag{3.8}$$

$$\breve{h}_1 = -\,h_5, \;\; \breve{h}_2 = h_4, \;\; \breve{h}_3 = h_1 + h_5 \;\text{ and }\; \breve{h}_4 = h_2 - h_4. \tag{3.9}$$

The inverse of this transformation (transformation from the Euclidean 4-space representation to *Plücker* coordinates) is given by

$$\mathbf{g} = (\breve{g}_3 - \breve{g}_1, \breve{g}_4 - \breve{g}_2, 1, \breve{g}_2, -\breve{g}_1, \breve{g}_1\breve{g}_4 - \breve{g}_2\breve{g}_3)\,; \tag{3.10}$$

$$\mathbf{h} = \left(\breve{h}_3 - \breve{h}_1, \breve{h}_4 - \breve{h}_2, 1, \breve{h}_2, -\breve{h}_1, \breve{h}_1\breve{h}_4 - \breve{h}_2\breve{h}_3\right). \tag{3.11}$$

Using this formulation, the distance between lines $\mathbf{g}$ and $\mathbf{h}$ can be given by

$$\delta\left(\breve{\mathbf{g}}, \breve{\mathbf{h}}\right)^2 = \sum_{i=1}^{4}\left(\breve{g}_i - \breve{h}_i\right)^2 + \left(\breve{g}_1 - \breve{h}_1\right)\left(\breve{g}_3 - \breve{h}_3\right) + \left(\breve{g}_2 - \breve{h}_2\right)\left(\breve{g}_4 - \breve{h}_4\right). \tag{3.12}$$

3.2.3 Interpolation Using Radial Basis Functions

Suppose that we want to estimate an unknown function, $f : \mathbb{R}^D \mapsto \mathbb{R}$, from a set of scattered data points $\mathcal{X} \doteq \{\mathbf{x}_i\} \subset \mathbb{R}^D$ (with D a natural number) and $\mathcal{Y} \doteq \{y_i\}$, where the set $\{\mathbf{x}_i, y_i\}$ forms a training data set $\{y_i = f(\mathbf{x}_i)\}$.

Interpolation requires the computation of an interpolating function, s :

$\mathbb{R}^D \mapsto \mathbb{R}$, that satisfies

$$s\left(\mathbf{x}_i\right) = f\left(\mathbf{x}_i\right), \quad \forall i. \tag{3.13}$$

Radial basis functions (RBF) [Buh03, Wen05, QSW93, SW93] can be used to solve this problem. For a set P of training points $\{\mathbf{x}_1, \ldots, \mathbf{x}_P\}$, the RBF interpolant function has the form

$$s\left(\mathbf{x}\right) = a_0 + \mathbf{a}_{\mathrm{x}}^T \mathbf{x} + \sum_{i=1}^{P} w_i \phi\left(||\mathbf{x} - \mathbf{x}_i||\right) \tag{3.14}$$

where $\phi : \mathbb{R}_+ \mapsto \mathbb{R}$ is the *radial basis function* and $\mathbf{a}_{\mathrm{x}} \in \mathbb{R}^D$. a_0, $\mathbf{a}_{\mathrm{x}}$ and w_i are the interpolant unknowns.

There are two types of kernel functions that can be used as the RBF interpolant. One type of kernel functions does not have shape parameters, like *thin–plate splines*, $\phi\left(r\right) = r^2 \log\left(r\right)$, or $\phi\left(r\right) = r^2$. The other type of kernel functions do have shape parameters, such as *Gaussian* functions $\phi\left(r\right) = \exp\left(-\gamma^2 r^2\right)$, and *multi–quadrics* with $\phi\left(r\right) = \left(\gamma^2 + r^2\right)^{1/2}$ where γ is the shape parameter.

The interpolation is obtained by means of the estimation of the unknown parameters of the interpolant ($\mathbf{a} = (a_0, \mathbf{a}_{\mathrm{x}})$ and $\mathbf{w} = (w_1, \ldots, w_P)$ of Equation (3.14)). The interpolating function $s\left(\mathbf{x}\right)$ has $P+D+1$ degrees of freedom and the data sets $\mathcal{X}$ and $\mathcal{Y}$ only yield P equations. For the estimation of the unknowns in Equation (3.14), additional constraints have to be used. Considering the function ϕ conditionally positive definite [Wen05], the following equations are verified,

$$\sum_{i=1}^{P} w_i = 0 \quad \& \quad \sum_{i=1}^{P} w_i x_1^{(i)} = \cdots = \sum_{i=1}^{P} w_i x_D^{(i)} = 0 \tag{3.15}$$

[Wen05] where $x_i^{(j)}$ is the i^{th} element of the j^{th} observation. The use of these constraints allows the estimation of all unknowns.

The estimation of the unknown parameters can be obtained using the following relation

$$\underbrace{\begin{bmatrix} \mathbf{\Phi} & \mathbf{P}^T \\ \mathbf{P} & \mathbf{0} \end{bmatrix}}_{\mathbf{\Gamma}} \begin{bmatrix} \mathbf{w} \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix} \tag{3.16}$$

where $\mathbf{P} \in \mathbb{R}^{3\times P}$ is such that

$$\mathbf{P} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_P \end{bmatrix}, \tag{3.17}$$

$\phi_{i,j} = \phi\left(||\mathbf{x}_i - \mathbf{x}_j||\right)$ and $\mathbf{\Gamma} \in \mathbb{R}^{(P+3)\times(P+3)}$. Furthermore, considering $D = 2$, we can see that $\text{rank}\left(\mathbf{\Gamma}\right) = P + 3$ implies $P \geqslant 3$.

3.3 Mathematical Derivations

In this section, we derive a set of mathematical properties that will be used-later in the dissertation. We derived the affine transformation for the space of line coordinates, represented in *Plücker* coordinates – Section 3.3.1. To conclude, we derive a very simple and elegant solution for the geometric distance between a 3D line and a non-incident point – Section 3.3.2.

3.3.1 Affine Transformation to the Line Space

In this section we derive the affine transformation for the line space of coordinates. Let us consider lines represented in *Plücker* coordinates – Section 3.2.1. The line coordinates before the application of the affine transformation is denoted as $\mathbf{l}^{(1)}\mathbb{R} = \left(\mathbf{d}^{(1)}, \mathbf{m}^{(1)}\right)$ and after the application of the transformation as $\mathbf{l}^{(2)}\mathbb{R} = \left(\mathbf{d}^{(2)}, \mathbf{m}^{(2)}\right)$. From the definition of *Pücker* coordinates [PW01], one has

$$\mathbf{d}^{(1)} = \mathbf{p}_2^{(1)} - \mathbf{p}_1^{(1)} \quad \text{and} \quad \mathbf{m}^{(1)} = -\mathbf{p}_2^{(1)} \times \mathbf{p}_1^{(1)} \tag{3.18}$$

for any two points $\mathbf{p}_1^{(1)}, \mathbf{p}_2^{(1)} \in \mathbb{R}^3$ (regular coordinates) that belong to the line, before the application of the affine transformation.

Using the affine parameters $\{\mathbf{B}, \mathbf{b}, b\}$, a point $\mathbf{p}_i^{(2)}$ after the application of the affine transformation is given by

$$\mathbf{p}_i^{(2)} = b^{-1}\left(\mathbf{B}\mathbf{p}_i^{(1)} + \mathbf{b}\right) \tag{3.19}$$

where $\mathbf{B} \in \mathbb{R}^{3\times3}$ and $\mathbf{b} \in \mathbb{R}^3$. From the definition of *Pücker* coordinates – Equation (3.18), we derive the following equations

$$\mathbf{d}^{(2)} = b^{-1}\left(\mathbf{B}\mathbf{p}_2^{(1)} + \mathbf{b}\right) - b^{-1}\left(\mathbf{B}\mathbf{p}_1^{(1)} + \mathbf{b}\right) = b^{-1}\mathbf{B}\mathbf{d}^{(1)} \tag{3.20}$$

and

$$\begin{aligned}\mathbf{m}^{(2)} &= -b^{-1}\left(\mathbf{B}\mathbf{p}_2^{(1)}+\mathbf{b}\right)\times b^{-1}\left(\mathbf{B}\mathbf{p}_1^{(1)}+\mathbf{b}\right)\\ &= -b^{-2}\left(\mathbf{B}\mathbf{p}_2^{(1)}\times\mathbf{B}\mathbf{p}_1^{(1)}+\mathbf{b}\times\left(\mathbf{B}\left(\mathbf{p}_1^{(1)}-\mathbf{p}_2^{(1)}\right)\right)\right).\end{aligned} \tag{3.21}$$

From the properties of the cross product, one has $\left(\mathbf{B}\mathbf{p}_2^{(1)}\right)\times\left(\mathbf{B}\mathbf{p}_1^{(1)}\right) = \det(\mathbf{B})\,\mathbf{B}^{-T}\left(\mathbf{p}_2^{(1)}\times\mathbf{p}_1^{(1)}\right)$. Using this result and from Equation (3.18), one obtains

$$\mathbf{m}^{(2)} = -b^{-2}\left(-\det(\mathbf{B})\,\mathbf{B}^{-T}\mathbf{m}^{(1)}-\mathbf{b}\times\mathbf{B}\mathbf{d}^{(1)}\right). \tag{3.22}$$

Using the results derived in Equations (3.20) and (3.22), we formalize the following Proposition.

Proposition 3.1. *Considering a line* $\mathbf{l}^{(1)}\mathbb{R} = \left(\mathbf{d}^{(1)},\mathbf{m}^{(1)}\right)\subset\mathbb{R}^6$*, represented in Plücker coordinates, and an affine transformation* $\{\mathbf{B},\mathbf{b},b\}$ *that verifies Equation (3.19). The line coordinates* $\mathbf{l}^{(2)}\mathbb{R} = \left(\mathbf{d}^{(2)},\mathbf{m}^{(2)}\right)\subset\mathbb{R}^6$ *after the application of the affine transformation are given by*

$$\mathbf{l}^{(2)}\mathbb{R} = \underbrace{\begin{bmatrix}\mathbf{B} & \mathbf{0}\\ b^{-1}\widehat{\mathbf{b}}\mathbf{B} & b^{-1}det(\mathbf{B})\,\mathbf{B}^{-T}\end{bmatrix}}_{\mathbf{E}}\mathbf{l}^{(1)}. \tag{3.23}$$

Note that $b\neq 0$ and matrix $\mathbf{B}$ is a non–singular matrix. As a result, matrix $\mathbf{E}$ is invertible and $\mathbf{l}^{(1)}$ can be estimated such that $\mathbf{l}^{(1)}\mathbb{R} = \mathbf{E}^{-1}\mathbf{l}^{(2)}$.

In the case of an Euclidean transformation, one has $\mathbf{B}=\mathbf{R}\in\mathcal{SO}(3)$, $\mathbf{b}=\mathbf{t}\in\mathbb{R}^3$ and $b=1$. Thus, we write the following Result.

Result 3.1. *Let us consider a line* $\mathbf{l}^{(1)}\mathbb{R} = \left(\mathbf{d}^{(1)},\mathbf{m}^{(1)}\right)\subset\mathbb{R}^6$*, represented in Plücker coordinates, and a rigid transformation* $\{\mathbf{R},\mathbf{t}\}$ *where* $\mathbf{R}\in\mathcal{SO}(3)$ *and* $\mathbf{t}\in\mathbb{R}^3$*. The line coordinates* $\mathbf{l}^{(2)}\mathbb{R} = \left(\mathbf{d}^{(2)},\mathbf{m}^{(2)}\right)\subset\mathbb{R}^6$ *after the application of the transformation are given by*

$$\mathbf{l}^{(2)}\mathbb{R} = \begin{bmatrix}\mathbf{R} & \mathbf{0}\\ \widehat{\mathbf{t}}\mathbf{R} & \mathbf{R}\end{bmatrix}\mathbf{l}^{(1)}. \tag{3.24}$$

We note that this particular case was derived in [Ple03, Stu05].

3.3.2 Geometric Distance Between a 3D Line and a 3D Point

In this subsection, we derive a geometric distance between a line in the world $\mathbf{l} \in \mathbb{L}^3$ and a non-incident 3D point $\mathbf{p} \in \mathbb{P}^3$. Let us consider that the line is represented in *Plücker* coordinates $\mathbf{l}\mathbb{R} = (\mathbf{d}, \mathbf{m}) \subset \mathbb{R}^6$, Equation (3.5) – Section 3.2.1, where $\mathbf{d} \in \mathbb{R}^3$ and $\mathbf{m} \in \mathbb{R}^3$ represents the moment and direction of the line respectively.

The distance between a line and a point does not change after a rigid transformation. As a result, instead of estimating the distance in the world coordinate system, the distance can be computed in any other coordinate system.

Let us consider a rigid transformation defined by the translation $\mathbf{t} \in \mathbb{R}^3$ and rotation $\mathbf{R} \in \mathcal{SO}(3)$. From Result 3.1 – Section 3.3.1, the line $\mathbf{l}^{(1)}\mathbb{R} = \left(\mathbf{d}^{(1)}, \mathbf{m}^{(1)}\right)$ can be represented in the new coordinate system $\mathbf{l}^{(2)}\mathbb{R} = \left(\mathbf{d}^{(2)}, \mathbf{m}^{(2)}\right)$, using

$$\mathbf{l}^{(2)}\mathbb{R} = \left(\mathbf{R}\mathbf{d}^{(1)}, \mathbf{R}\mathbf{m}^{(1)} + \widehat{\mathbf{t}}\mathbf{R}\mathbf{d}^{(1)}\right). \tag{3.25}$$

Let us consider a generic line in the world coordinate system $\mathbf{l}^{(1)} \in \mathbb{L}^3$ and a non–incident point $\mathbf{p} \in \mathbb{P}^3$. The coordinates of the same line can be represented in a new coordinate system centered at $\mathbf{p}$, setting $\mathbf{t} = -\mathbf{p}$ and $\mathbf{R} = \mathbf{I}$, as

$$\mathbf{l}^{(2)}\mathbb{R} = \left(\mathbf{d}^{(2)}, \mathbf{m}^{(2)}\right) = \left(\mathbf{d}^{(1)}, \mathbf{m}^{(1)} - \widehat{\mathbf{p}}\mathbf{d}^{(1)}\right). \tag{3.26}$$

In this coordinate system, a plane $\mathbf{\Pi} \doteq \mathbf{l}^{(2)} \cup \mathbf{0}$, spanned by the line and the origin of the coordinate system, can be defined. In this coordinate system, any point $\mathbf{q}$ incident on the line $\mathbf{l}^{(2)}$ verifies

$$\mathbf{q} = \underbrace{\left(\mathbf{q} \cdot \boldsymbol{e}_{\mathbf{d}^{(2)}}\right)}_{\mathbf{q}_-} \boldsymbol{e}_{\mathbf{d}^{(2)}} + \underbrace{\left(\mathbf{q} \cdot \boldsymbol{e}_{\boldsymbol{\xi}}\right)}_{\mathbf{q}_+} \boldsymbol{e}_{\boldsymbol{\xi}} \tag{3.27}$$

where $\boldsymbol{e}_{\mathbf{d}^{(2)}}$ and $\boldsymbol{e}_{\boldsymbol{\xi}}$ are the orthogonal basis for the subspace $\mathbf{\Pi}$. The representation of this basis is show in Figure 3.1.

Since $\mathbf{q}_+$ is constant, vector $\mathbf{q}$ has the minimum norm when $\mathbf{q}_- = 0$ or $\mathbf{q} \cdot \boldsymbol{e}_{\mathbf{d}^{(2)}} = 0$ and, therefore, the distance between the line and the point can be defined as $\delta(\mathbf{l}, \mathbf{p}) = ||\mathbf{q}||$, such that $\mathbf{q} \cdot \boldsymbol{e}_{\mathbf{d}^{(2)}} = 0$.

From the definition of moment of the line, we have $\mathbf{m}^{(2)} = \mathbf{q} \times \mathbf{d}^{(2)}$ for

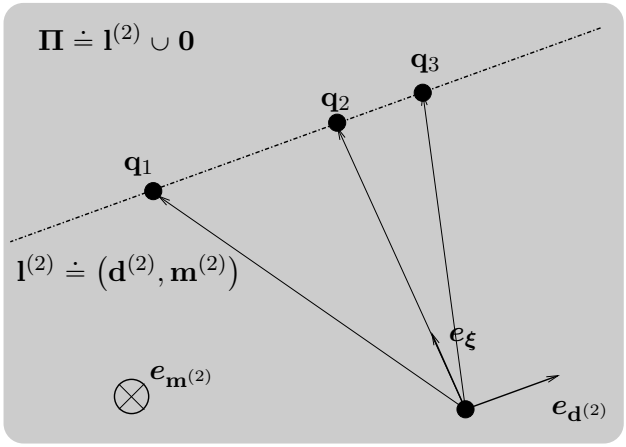


Figure 3.1: *In this figure we display the representation of a line* $\mathbf{l}^{(1)}$ *in the coordinate system centered at point* $\mathbf{p}$*. The plane* $\boldsymbol{\pi}$ *is defined by* $\mathbf{l}^{(2)}$ *and the origin of the coordinate system.*

any $\mathbf{q}$ that belong to the line, which means that

$$\mathbf{m}^{(2)} = ||\mathbf{q}|| \left|\left|\mathbf{d}^{(2)}\right|\right| \sin\left(\Theta\left(\mathbf{q}, \mathbf{d}^{(2)}\right)\right) \boldsymbol{e}_{\boldsymbol{\xi}} \times \boldsymbol{e}_{\mathbf{d}^{(2)}} \tag{3.28}$$

where $\Theta(\mathbf{q}, \mathbf{d})$ is the angle between the vector $\mathbf{q}$ and $\mathbf{d}$. Note that the aim is to find $\mathbf{q}$ that verifies the constraint $\mathbf{q} \cdot \boldsymbol{e}_{\mathbf{d}^{(2)}} = 0$, which implies $\sin(\Theta(\mathbf{q}, \mathbf{d})) = 1$ and, as a result,

$$\left|\left|\mathbf{m}^{(2)}\right|\right| = ||\mathbf{q}|| \left|\left|\mathbf{d}^{(2)}\right|\right|. \tag{3.29}$$

Without loss of generality, let us consider that $\mathbf{l}^{(1)} = \left(\mathbf{d}^{(1)}, \mathbf{m}^{(1)}\right)$ where $\left|\left|\mathbf{d}^{(1)}\right|\right| = 1$ which, from Equation (3.26), implies that $\left|\left|\mathbf{d}^{(2)}\right|\right| = 1$. The distance between a line and a point can be defined as

$$\delta(\mathbf{l}, \mathbf{p}) = ||\mathbf{q}|| = \left|\left|\mathbf{m}^{(2)}\right|\right|. \tag{3.30}$$

To conclude, using the Equations (3.30) and (3.26), we derive the following Proposition

Proposition 3.2. *For a line* $\mathbf{l}\mathbb{R} = (\mathbf{d}, \mathbf{m}) \subset \mathbb{R}^6$*, represented in Plücker coordinates, and a non-incident 3D point* $\mathbf{p} \in \mathbb{R}^3$*, the geometric distance between* $\mathbf{l}$ *and* $\mathbf{p}$ *are given by* $\delta(\mathbf{l}, \mathbf{p})$*, such that*

$$\delta(\mathbf{l}, \mathbf{p}) = \frac{\left|\left|\begin{bmatrix} -\widehat{\mathbf{p}} & \mathbf{I} \end{bmatrix} \mathbf{l}^{(1)}\right|\right|}{||\mathbf{d}^{(1)}||}. \tag{3.31}$$

Part II: Smooth Camera Model and its Calibration

Contents

4 Introduction **20**
4.1 Our Approach 20

5 Proposed Model and Calibration **23**
5.1 Smooth Camera Model 23
5.2 Formalization 24
5.2.1 Vector–Valued Function 24
5.2.2 Linear Point–based Calibration 26
5.2.3 Data Normalization 30

6 Experiments **33**
6.1 Results with Synthetic Data Sets 33
6.1.1 Evaluation Results Using Smooth Camera Models . . . 33
6.1.2 Results Using Non–Smooth Camera Models 37
6.1.3 Experimental Results Using 3D Data from a Single Surface 38
6.2 Results with Data sets of Real Images 40
6.3 Using a Calibrated Perspective Camera to Acquire a Data-set with Real Data 43
6.3.1 Removing Distortion: Results for a Spherical Catadioptric System 46

7 Discussion **49**

Chapter 4

Introduction

Let us consider the state–of–the–art methods of Grossberg & Nayar [GN01, GN05] and Sturm & Ramalingam at [SR04, RSL05] – described in Section 2.1. Both the approaches mentioned are discrete and non–parametric, using mapping arrays (*raxels*, Table 2.1) to calibrate the imaging model. Image pixels have associated a set of parameters that are independent from their neighbours.

Most of the useful camera models have, in general, a pixel–ray relationships that varies smoothly along the image. That is the case of the central perspective camera, non-central catadioptric systems with quadric mirrors, linear cameras, fisheye cameras and of camera models that include refracting elements.

In this thesis we change the discrete model defined by Grossberg and Nayar (the *raxel camera model*) by assuming that the 3D rays associated to the pixels vary smoothly throughout the image. We call this model the *smooth camera model*. The assumption of smoothness is used as a constraint in the definition of the camera model and therefore in the calibration method.

4.1 Our Approach

Since the estimation of the *raxels* for all the image pixels requires a lengthy calibration procedure, previous authors ([GN05] and [SR04, RSL05]) have also assumed that the relationship between neighboring pixels and neighboring 3D rays in the world varies smoothly. However, these constraints are applied after the estimation of at least a subset of all possible image pixels. Sturm and Ramalingam used this constraint to interpolate between a subset of calibrated *raxels* corresponding to neighboring pixels. Grossberg and Nayar at [GN05] defined a *caustic raxel model* that aimed at estimating a

caustic surface from a set of *raxels*.

In both previous cases, it is required the estimation of the at least a subset of all *raxels*. Such estimation needs that more than one 3D point be used for each image pixel. Moreover, the use of constraints enforcing smoothness between neighboring pixels is used post calibration to interpolate between pixels that were not calibrated.

In the proposed approach, the assumption that the relationship between pixels and rays varies smoothly is used directly in the model. For that purpose a vector-valued function that can represent any *smooth camera model* is defined. A parametric representation for the *general camera model* is defined which results in:

- A decrease in the number of unknown parameters – instead of the seven geometric parameters of the *raxel* for all image pixels, our method requires only the parameters that define the vector-valued function;

- The number of parameters and image resolution becomes independent – since we are using a function that operates in the image space, our approach can be seen as independent from the image resolution.

Using this camera model, a new calibration procedure can be developed, where:

- Only one world point is needed for each image point – unlike previous methods, [GN01] and [SR04] that require at least two points in the world, for each image point;

- The calibration procedure has lower complexity – unlike previous methods, our approach only requires to match a 3D point with an image point used in the calibration. Moreover, all the calibration parameters are estimated in a single step.

As in [SR04], only geometric entities of *general camera model* are considered.

In Figure 4.1, an example of the advantages of the use of the proposed method in the calibration of *smooth camera models* is shown. The minimal case (defined according to the theoretical conditions) is considered, where six matchings between world and image points are known (see the proof in Section 5.2.2). In this example all the coordinates of the image points are different. As we can see, the estimated 3D lines pass through the corresponding 3D points (the 3D points used in the calibration) using one-to-one mappings, which occurs as a result of applying the constraint that neighboring pixels must correspond to neighboring rays in the world (smoothness constraint).

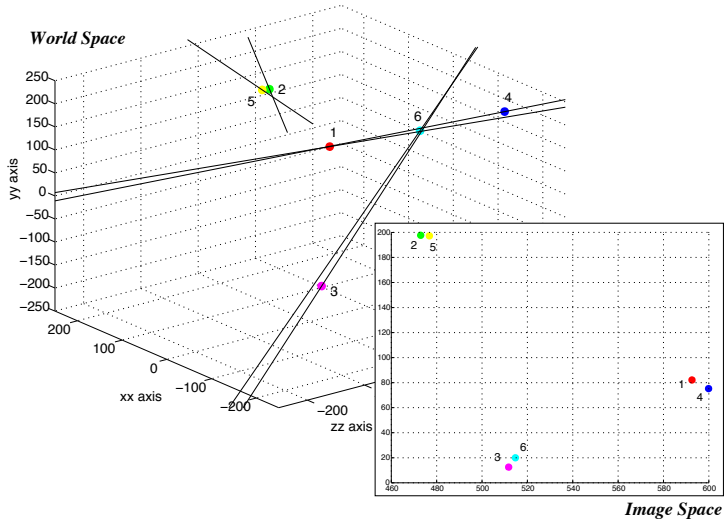


Figure 4.1: *In this figure we display the result of the application of the calibration method to the minimal case according to the theoretical constraints (see Section 5.2.2). In the minimal case there are six points in the the image and their correspondent points in the world.*

The proposed calibration method can be considered a black box general *smooth camera model* which is linear and leads to a simpler calibration method.

Chapter 5

Proposed Model and Calibration

In this chapter, we describe the proposed *smooth camera model* – Section 5.1 and, in Section 5.2, we formalize it. A calibration procedure is also described.

5.1 Smooth Camera Model

From the definition of *General Camera Model* Figure 2.1, introduced by Grossberg and Nayar [GN01], each pixel in the image $\mathbf{x} \in \mathbb{P}^2$ is mapped to a ray in 3D space $\mathbf{l} \in \mathbb{L}^3$. As mentioned before, the model is based on an array of parameters called *raxel* and it can be used to represent central or non–central camera model as well as smooth and non–smooth camera models.

In that formulation, a complete *general camera model* is represented by a non–parametric discrete array of *raxels*, that contains all possible pixels in an image. This means that the pixel–line mappings are required for all pixels, independently of the image resolution or of the smoothness on the variation of the parameters (corresponding to the 3D lines associated to neighboring pixels). If we consider only the geometric entities in Grossberg and Nayar's model, each *raxel* contains at least seven parameters. Thus, for each pixel, there are seven unknown parameters to be computed. For an image with size $N \times M$, there are $7NM$ unknown parameters to be computed. We aim at deriving a model that requires a smaller number of parameters.

Since most of the imaging devices are characterized by having a smooth relationship between image pixels and lines in the world, we use the assumption that the pixel–line mappings can be represented by a smoothly varying vector–valued function $\mathbf{f} : \mathbb{P}^2 \mapsto \mathbb{L}^3$, that maps a point in the image plane to a line in 3D space. We use this approach to represent the *smooth camera model*.

This assumption significantly decreases the number of model unknowns and also allows to filter out some error due to noise.

In most cases, a general direct projection model does not exist, since one 3D line can be mapped into more than one point in the image plane. As a result, a *general camera model* can only be defined when considering the mapping from image coordinates to 3D lines. A recent result, however, derived an analytical model for the forward projection of a non-central system made up of a perspective camera and a rotationally symmetric conic mirror [ATR10].

As previously described, a single mapping between points in the image to points in the world is insufficient to define a *raxel* and as a consequence it can not define a *general camera model*. The information of the direction of the line can not be recovered. However, if we assume that the relationship between image points and world lines varies smoothly, we are imposing new constraints to lines. Those constraints will implicitly allow the complete calibration of the *smooth camera model* using only a single point in the world for a point in the image, as proved in the following sections. An example can be seen in Figure 4.1.

5.2 Formalization

In this section we describe the formulation of the proposed approach to represent the *smooth camera model* – Section 5.2.1, and its calibration procedure – Section 5.2.2. In Section 5.2.3, we describe the data-set normalization.

5.2.1 Vector–Valued Function

A 3D line representation has to be chosen for the output of the vector–valued function. Lines in 3D have four degrees of freedom. However, none of the compact four variable representations for 3D lines is complete.

Plücker coordinates (Section 3.2.1) are a complete, elegant and easy to understand line representation. On the other hand, it has six elements to represent four degrees of freedom. It is defined up to a scale factor and has an orthogonal constraint associated to its elements.

Instead of using *Plücker* coordinates to represent 3D lines, we use a vector made up by stacking the vectors of the direction and moment $\breve{\mathbf{l}}\mathbb{R} = \left(\breve{\mathbf{d}}, \breve{\mathbf{m}}\right)$. In other words, two independent vectors are estimated, up to the same scale factor.

Good estimates of $\breve{\mathbf{d}}$ and $\breve{\mathbf{m}}$ yield small deviations from the orthogonal constraint. However, it is possible to find orthogonal vectors $\mathbf{d}$ and $\mathbf{m}$ from

their estimates using *Schmidt* orthogonalization [GVL96], by finding the closest rotation matrix to $\left[\begin{array}{ccc} \breve{\mathbf{d}}' & \breve{\mathbf{m}}' & \breve{\mathbf{d}}' \times \breve{\mathbf{m}}' \end{array}\right]$, where $\mathbf{x}' = \mathbf{x}/||\mathbf{x}||$, or by using the algorithm proposed by Bartoli and Sturm [BS05].

There are several ways to estimate a non–injective function from a set of scattered data. We use the RBF interpolant described in Section 3.2.3

$$s(\mathbf{x}) = a_0 + \mathbf{a}_{\mathrm{x}}^T \mathbf{x} + \sum_{i=1}^{P} w_i \phi(||\mathbf{x} - \mathbf{c}_i||) \tag{5.1}$$

where $\{\mathbf{c}_i\}, \mathbf{x} \in \mathbb{R}^2$. We can rewrite the previous equation, in matrix form as

$$s(\mathbf{x}) = \left[\begin{array}{cc} \boldsymbol{\phi}(\mathbf{x}) & \mathbf{p}(\mathbf{x}) \end{array}\right] \underbrace{\left[\begin{array}{c} \mathbf{w} \\ \mathbf{a} \end{array}\right]}_{\mathbf{h_{wa}}} \tag{5.2}$$

where $\boldsymbol{\phi}(\mathbf{x})$ and $\mathbf{p}(\mathbf{x})$ are row vectors, with $\phi_i(\mathbf{x}) = \phi(||\mathbf{x} - \mathbf{c}_i||)$ and

$$\mathbf{p}(\mathbf{x}) = \left[\begin{array}{ccc} 1 & x_1 & x_2 \end{array}\right]. \tag{5.3}$$

The vector–valued function output is $\breve{\mathbf{l}} \in \mathbb{R}^6$ where each $\breve{l}_i$ is independent. Thus, we can use six independent RBF interpolants to form the vector–valued function $\mathbf{s}(\mathbf{x})$ such that

$$\mathbf{s}(\mathbf{x}) = \left[\begin{array}{cccc} s_1(\mathbf{x}) & s_2(\mathbf{x}) & \ldots & s_6(\mathbf{x}) \end{array}\right], \tag{5.4}$$

which, using Equation (5.2), can be rewritten as

$$\mathbf{s}(\mathbf{x}) = \left[\begin{array}{cc} \boldsymbol{\phi}(\mathbf{x}) & \mathbf{p}(\mathbf{x}) \end{array}\right] \underbrace{\left[\begin{array}{cccc} \mathbf{h}_{\mathbf{wa}}^{(1)} & \mathbf{h}_{\mathbf{wa}}^{(2)} & \ldots & \mathbf{h}_{\mathbf{wa}}^{(6)} \end{array}\right]}_{\mathbf{H_{wa}}}. \tag{5.5}$$

The vector–valued function is a row vector, $\mathbf{s}(\mathbf{x}) \in \mathbb{R}^{1\times 6}$, which implies $\breve{\mathbf{l}}(\mathbf{x})\,\mathbb{R} = \mathbf{s}^T(\mathbf{x})$.

For a set $\{\mathbf{c}_i\}$, a matrix $\mathbf{H_{wa}}$ and a certain RBF, the estimates of the direction and moment vectors as a function of the coordinates of the image points are given by

$$\breve{\mathbf{l}}(\mathbf{x})^T\,\mathbb{R} = \left[\begin{array}{cc} \breve{\mathbf{d}}(\mathbf{x})^T & \breve{\mathbf{m}}(\mathbf{x})^T \end{array}\right]\mathbb{R} = \left[\begin{array}{cc} \boldsymbol{\phi}(\mathbf{x}) & \mathbf{p}(\mathbf{x}) \end{array}\right]\mathbf{H_{wa}}. \tag{5.6}$$

To conclude this section, we give the following definitions:

- From the previous equation, it can be observed that for two different imaging systems using the same set of points $\{\mathbf{c}_i\}$, the estimation of $\breve{\mathbf{l}}$ for an image point $\mathbf{x}$ only depends on matrix $\mathbf{H_{wa}}$. Thus, we call $\mathbf{H_{wa}}$ the *camera matrix*. On the other hand, for the same imaging system, the values of the parameters of the *camera matrix* depend on the set $\{\mathbf{c}_i\}$, and that is why we call them *control points*. Usually, in statistics, the set $\{\mathbf{c}_i\}$ is called *centers*. In computer vision, the word center in an imaging system typically designates the center of the projection. As a result, we chose to name to the set $\{\mathbf{c}_i\}$ as *control points*;

- For a set P of *control points* defined *a priori* $\{\mathbf{c}_i\}$ and a *camera matrix* $\mathbf{H_{wa}} \in \mathbb{R}^{(P+3)\times 6}$, we define the general *smooth camera model* model by the vector–valued function $\mathbf{s} : \mathbb{R}^2 \mapsto \mathbb{R}^6$;

- The *smooth camera model* only depends on the unknown matrix $\mathbf{H_{wa}}$, for a set of previously defined *control points*. Therefore, the complete calibration of the *smooth camera model* can be obtained by estimating $6P + 18$ unknown parameters, that sets up the *camera matrix*.

Using the proposed *smooth camera model*, in the next section we propose a simple appropriate calibration procedure.

5.2.2 Linear Point–based Calibration

One disadvantage of the use of the *general camera model* is the difficulty of its calibration. Grossberg & Nayar [GN01] and Sturm & Ramalingam [SR04] define two different methods for the calibration. However in both cases, the complete calibration is achieved when we have a *raxel* for all pixels of the image. Note that the estimation of a single raxel requires at least two points in the world for the each image pixel. Since the estimation of the *raxels* for every pixel is a difficult task, interpolation methods were used to estimate all the *raxels* from a subset of calibrated *raxels* – see Section 2.1, Table 2.1. The goal in the calibration of a *smooth camera model* is to allow the calibration of the respective model without estimating each and all the *raxels* while, at same time, allowing the same resolution.

In the method described in this thesis, the calibration procedure only requires a set of point correspondences $\{\mathbf{x}_i \mapsto \mathbf{p}_i\}$, (where $\{\mathbf{p}_i\}$ is the set of world points and $\{\mathbf{x}_i\}$ are their corresponding points in the image plane). This means that point correspondences that satisfy

$$\mathbf{x}_i \neq \mathbf{x}_j, \quad \forall_{i \neq j}, \tag{5.7}$$

can be used.

There are two sets of unknowns in the calibration procedure: the set of *control points* $\{\mathbf{c}_i\}$ and the elements of the *camera matrix* $\mathbf{H_{wa}}$. *Control points* can be defined *a priori*, by selecting a set P of scattered image points, that can or cannot be a subset of data points $\{\mathbf{c}_i\} \subset \{\mathbf{x}_i\}$.

In the rest of this section, we describe a linear method to estimate the $6P + 18$ parameters of the *camera matrix*, for a set of *control points* defined *a priori*.

Calibration Matrix M

World points incident on lines must verify Equation (3.6). From $\breve{\mathbf{l}}(\mathbf{x})\,\mathbb{R} = \mathbf{s}(\mathbf{x})^T$ and Equation (3.6)

$$\breve{\mathbf{l}}(\mathbf{x})^T\,\mathbf{Q}(\mathbf{p})^T = \mathbf{s}(\mathbf{x})\,\mathbf{Q}(\mathbf{p})^T = \mathbf{0}. \tag{5.8}$$

Replacing $\mathbf{s}(\mathbf{x})$ from the previous equation, using Equation (5.5), we get

$$\underbrace{\begin{bmatrix} \boldsymbol{\phi}(\mathbf{x})^T & \mathbf{p}(\mathbf{x})^T \end{bmatrix}}_{\mathbf{r}(\mathbf{x})} \mathbf{H_{wa}}\mathbf{Q}(\mathbf{p})^T = \mathbf{0} \tag{5.9}$$

where $\mathbf{r}(\mathbf{x}) \in \mathbb{R}^{1\times(P+3)}$.

Since we are considering that we know the *control points* $\{\mathbf{c}_i\}$, the unknowns are the elements of *camera matrix* $\mathbf{H_{wa}}$. Thus and using the *Kronecker* product – Section 3.1, we rewrite Equation (5.9) in order to isolate the unknown *camera matrix* as

$$\Big(\mathbf{Q}(\mathbf{p}) \otimes \mathbf{r}(\mathbf{x})\Big)\mathrm{vec}(\mathbf{H_{wa}}) = \mathbf{0}. \tag{5.10}$$

The matrix $\mathbf{Q}(\mathbf{p}) \otimes \mathbf{r}(\mathbf{x}) \in \mathbb{R}^{4\times(6P+18)}$ and $\mathrm{vec}(\mathbf{H_{wa}}) \in \mathbb{R}^{(6P+18)}$.

For a set of point correspondences $\{\mathbf{x}_i \mapsto \mathbf{p}_i\}$, for $i = 1, \ldots, N$, of the same imaging system, the following relation must be verified

$$\underbrace{\begin{bmatrix} \mathbf{Q}(\mathbf{p}_1) \otimes \mathbf{r}(\mathbf{x}_1) \\ \mathbf{Q}(\mathbf{p}_2) \otimes \mathbf{r}(\mathbf{x}_2) \\ \vdots \\ \mathbf{Q}(\mathbf{p}_N) \otimes \mathbf{r}(\mathbf{x}_N) \\ \mathbf{D} \end{bmatrix}}_{\mathbf{M}} \mathrm{vec}(\mathbf{H_{wa}}) = \mathbf{0}. \tag{5.11}$$

We call $\mathbf{M} \in \mathbb{R}^{(4N+18)\times(6P+18)}$ the calibration matrix. The matrix $\mathbf{D} \in \mathbb{R}^{18\times(6P+18)}$ is as

$$\mathbf{D} = \begin{bmatrix} \mathbf{0} & -\boldsymbol{\Delta}_1\mathbf{R}_1 & \boldsymbol{\Delta}_2\mathbf{R}_1 & \mathbf{R}_1 & \mathbf{0} & \mathbf{0} \\ \boldsymbol{\Delta}_1\mathbf{R}_1 & \mathbf{0} & -\boldsymbol{\Delta}_3\mathbf{R}_1 & \mathbf{0} & \mathbf{R}_1 & \mathbf{0} \\ -\boldsymbol{\Delta}_2\mathbf{R}_1 & \boldsymbol{\Delta}_3\mathbf{R}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{R}_1 \\ \mathbf{0} & -\boldsymbol{\Delta}_4\mathbf{R}_2 & \boldsymbol{\Delta}_5\mathbf{R}_2 & \mathbf{R}_2 & \mathbf{0} & \mathbf{0} \\ \boldsymbol{\Delta}_4\mathbf{R}_2 & \mathbf{0} & -\boldsymbol{\Delta}_6\mathbf{R}_2 & \mathbf{0} & \mathbf{R}_2 & \mathbf{0} \\ -\boldsymbol{\Delta}_5\mathbf{R}_2 & \boldsymbol{\Delta}_6\mathbf{R}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{R}_2 \end{bmatrix} \tag{5.12}$$

where

$$\mathbf{R}_1 = \begin{bmatrix} \mathbf{P}_1 & \mathbf{0}_{P\times 3} \end{bmatrix}, \quad \mathbf{R}_2 = \begin{bmatrix} \mathbf{P}_2 & \mathbf{0}_{P\times 3} \end{bmatrix} \tag{5.13}$$

$\mathbf{R}_i \in \mathbb{R}^{3\times P+3}$ and $\mathbf{P}_1, \mathbf{P}_2 \in \mathbb{R}^{3\times P}$ are the stacking of the set $\{\mathbf{p}(\mathbf{x}_i)\}$ for $i = 1, \ldots, P$, and $\{\mathbf{p}(\mathbf{x}_i)\}$ for $i = P+1, \ldots, 2P$ respectively. Matrices $\boldsymbol{\Delta}_i \in \mathbb{R}^{3\times 3}$ are random diagonal matrices. The constraints that result from

$$\mathbf{D}\text{vec}(\mathbf{H}_{\mathbf{wa}}) = \mathbf{0} \tag{5.14}$$

were derived from the additional constraints given from the use of conditionally positive definite *radial basis functions* – See Section 3.2.3, Equation (3.15). For more information see Appendix A. These constraints are specialy important for the case where the minimal number of points is given, described later in this section.

To conclude, the calibration procedure is reduced to the estimation of the unknown *camera matrix* $\mathbf{H}_{\mathbf{wa}}$ such that

$$\mathbf{M}\text{vec}(\mathbf{H}_{\mathbf{wa}}) = \mathbf{0}, \quad \text{which means} \quad \text{vec}(\mathbf{H}_{\mathbf{wa}}) \in \text{null}(\mathbf{M}) \tag{5.15}$$

Computation of the *Camera Matrix* $\mathbf{H}_{\mathbf{wa}}$

From Equation (5.15), the estimate of the *camera matrix* must belong the the *null–space* of the calibration matrix $\mathbf{M}$. To ensure an unique solution, and since $\text{vec}(\mathbf{H}_{\mathbf{wa}}) \in \mathbb{R}^{6P+18}$, the dimension of the *null–space* of $\mathbf{M}$ must be $\text{nullity}(\mathbf{M}) = 1$. As a result, and since $\text{rank}(\mathbf{M}) + \text{nullity}(\mathbf{M}) = 6P + 18$ [GVL96], the dimension of the *column space* of $\mathbf{M}$ must be $\text{rank}(\mathbf{M}) = 6P + 17$.

From Equation (5.6), the *camera matrix* can be defined up to a scale factor. Thus, assuming that $\text{nullity}(\mathbf{M}) = 1$, any element of the one dimensional *null–space* of $\mathbf{M}$ is a solution, except for the *trivial solution* $\text{vec}(\mathbf{H}_{\mathbf{wa}}) = \mathbf{0}$.

To prove that the dimension of the *column space* of $\mathbf{M}$ is $\text{rank}(\mathbf{M}) = 6P+17$, we decompose Equation (5.11) into rows, by means of the *Kronecker* product – Section 3.1, Equation (3.4). Since the permutation of the rows in any matrix does not change the dimension of the *column space*, considering $N = 2P$, we define the following Theorem.

Theorem 5.1. *For a set of constraints of the data set (described below) and $N = 2P$, it is possible to rearrange matrix $\mathbf{M}$ as a block of matrices similar to $\mathbf{\Gamma}$ (Equation (3.16)). It can be proved that: matrices $\mathbf{\Gamma}$ are full-rank; and $\text{rank}(\mathbf{D}) = 17$. To conclude, we can prove that, **in general**, the dimension of the column space of $\mathbf{M}$ is $\text{rank}(\mathbf{M}) = 6P + 17$.*

The *proof* to this theorem is derived in the Appendix A.

From this result, and since $\text{rank}(\mathbf{\Gamma}) = P + 3$ implies $P \geqslant 3$ – see Section 3.2.3, we can conclude that the minimal configuration to ensure a unique solution for the computation of the camera matrix corresponds to $N = 6$. An example of a calibration corresponding to the minimal case is shown in Figure 4.1.

Relationship between *Control Points*, point correspondences and *radial basis function* used in the calibration

In this section, we describe the constraints that must be met by $\{\mathbf{x}_i\}$ and $\{\mathbf{c}_i\}$ to obtain $\text{rank}(\mathbf{M}) = 6P+17$. Two types of *radial basis functions*, that can be applied to get that *rank* are also discussed.

Let us assume that a set of point correspondences, $\{\mathbf{x}_i \mapsto \mathbf{p}_i\}$ for $i = 1, \ldots, 2P$ and $\mathbf{x}_i \neq \mathbf{x}_j, \ \forall i \neq j$ is available, and that it can be split into two sub-sets such that

$$\begin{array}{rcll} \mathcal{K}^{(1)} & \doteq & \{\mathbf{x}_i\} & \text{for} \quad i = 1, \ldots, P \\ \mathcal{K}^{(2)} & \doteq & \{\mathbf{x}_i\} & \text{for} \quad i = P+1, \ldots, 2P. \end{array} \tag{5.16}$$

In [QSW93] it is shown that only *control points* ($\{\mathbf{c}_i\}$, for $i = 1, \ldots, P$) and *data points* ($\{\mathbf{x}_i\}$, for $i = 1, \ldots, 2P$) that meet the condition

$$d < \epsilon q \tag{5.17}$$

can be considered. In Equation (5.17), $d = \min\{d_1, d_2\}$, $0 < \epsilon \leqslant 1$, $d_1 = \max\{||\mathbf{x}_i - \mathbf{c}_i||\}$, where the set $\{\mathbf{x}_i\}$ belongs to the set $\mathcal{K}^{(1)}$, $d_2 = \max\{||\mathbf{x}_i - \mathbf{c}_i||\}$, where the set $\{\mathbf{x}_i\}$ belongs to $\mathcal{K}^{(2)}$ and $2q = \min_{j \neq i}\{||\mathbf{c}_i - \mathbf{c}_j||\}$.

Moreover, if we consider $\{\mathbf{c}_i\} \doteq \mathcal{K}^{(1)}$ and if $d_2 < \epsilon q$ then we can also obtain $\text{rank}(\mathbf{M}) = 6P+17$. This solution also meets the constraint $\mathbf{x}_i \neq \mathbf{x}_j, \ \forall i \neq j$.

Quak *et al.* [QSW93] proved that $\phi_1(r) = (\gamma_1^2 + r^2)^{1/2}$ and $\phi_2(r) = \exp(-\gamma_2 r^2)$ are good choices for *radial basis functions*, since an adequate choice for the shape parameters γ_1 or γ_2 can decrease the negative effects of q and ϵ – in Equation (5.17).

5.2.3 Data Normalization

One of the issues in the method proposed in the previous section is related to the shape parameters of the RBF. The choice of their values depends on the distribution of the set of image points $\{\mathbf{x}_i\}$. To avoid this problem, a data normalization procedure for both image and world coordinates was developed. We consider non–isotropic normalization [Har97] for both image and world points, such that

- the centroid is at the origin;
- the principal moments are equal to unity.

If on one hand, normalization of the image space coordinates is easy to implement, on the other hand, normalization of the coordinates of the set of world points $\{\mathbf{p}_i\}$ will have additional implications. The normalization of the coordinates of the world points, $\{\mathbf{p}_i\}$, in the calibration procedure will imply a change on the parameters of the *smooth camera model*. To account for this change, the inverse of this transformation has to be applied to the parameters of the *smooth camera model* that were estimated.

In the rest of the section describe how we normalize the set of image and world points. To conclude the section, we describe the complete calibration algorithm for the proposed *smooth camera model*.

Scaling the Coordinates of the Set of Image Points

For the normalization of the image space, an affine transformation

$$\mathbf{u}_i = a^{-1}(\mathbf{A}\mathbf{x}_i + \mathbf{a}) \tag{5.18}$$

is considered, where $\mathbf{A} \in \mathbb{R}^{2\times 2}$ is upper triangular, and $\mathbf{a} \in \mathbb{R}^2$.

To obtain the normalization parameters $\{\mathbf{A}, \mathbf{a}, a\}$, an affine transformation based on *Choleski* factorization is used [Har97, GVL96]. Let us consider the *homogeneous* representation of the image coordinates $\mathbf{x}$ as $\widetilde{\mathbf{x}}$. Since matrix $\sum_{i=1}^{N} \widetilde{\mathbf{x}}_i \widetilde{\mathbf{x}}_i^T$ is symmetric and positive definite, using the *Choleski* factoriza-

tion one can define

$$\sum_{i=1}^{N} \widetilde{\mathbf{x}}_i \widetilde{\mathbf{x}}_i^T = N\mathbf{K}\mathbf{K}^T, \tag{5.19}$$

here $\mathbf{K} \in \mathbb{R}^{3\times 3}$ is an upper triangular matrix. Developing this equation, we get

$$\sum_{i=1}^{N} \mathbf{K}^{-1}\widetilde{\mathbf{x}}_i \widetilde{\mathbf{x}}_i^T \mathbf{K}^{-T} = N\mathbf{I}. \tag{5.20}$$

As a result and using $\widetilde{\mathbf{u}}_i = \mathbf{K}^{-1}\widetilde{\mathbf{x}}_i$, one obtains $\sum_{i=1}^{N} \widetilde{\mathbf{u}}_i \widetilde{\mathbf{u}}_i^T = N\mathbf{I}$ which means that the set $\{\mathbf{u}_i\}$ has its centroid at the origin of the coordinate systems and the two principal moments of the set of points are equal to one.

Since $\mathbf{K}$ is upper triangular, $\mathbf{K}^{-1}$ is also upper triangular. As a result, the affine parameters $\{\mathbf{A}, \mathbf{a}, a\}$ can be defined such that

$$\mathbf{K}^{-1} = \begin{bmatrix} \mathbf{A} & \mathbf{a} \\ \mathbf{0} & a \end{bmatrix}. \tag{5.21}$$

Scaling the Coordinates of the Set of World Points

To apply the normalization to the coordinates of the points in world space, a similar method to the one applyed to the image space is considered. The coordinates of the set of world points $\{\mathbf{p}_i\}$ are changed into $\{\mathbf{q}_i\}$ such that

$$\mathbf{q}_i = b^{-1}\left(\mathbf{B}\mathbf{p}_i + \mathbf{b}\right) \quad \text{and} \quad \mathbf{K}^{-1} = \begin{bmatrix} \mathbf{B} & \mathbf{b} \\ \mathbf{0} & b \end{bmatrix}. \tag{5.22}$$

However, in this case, $\mathbf{K}^{-1} \in \mathbb{R}^{4\times 4}$ is an upper triangular matrix (for points in 3D space). $\mathbf{B} \in \mathbb{R}^{3\times 3}$ is upper triangular and $\mathbf{b} \in \mathbb{R}^3$.

Note that this normalization has consequences: it will change the coordinates of the line space. The output of function $\mathbf{s}$ are the coordinates of the 3D lines. Therefore one has to account for this change of coordinates. The line coordinates which are the output of function $\mathbf{s}$ have to be changed into the original coordinate frame.

Using the affine transformation defined in Proposition 3.1, we define an affine transformation in line space as

$$\mathbf{l}^{(2)}\mathbb{R} = \mathbf{E}\mathbf{l}^{(1)}, \tag{5.23}$$

Algorithm 5.1 Calibration of Smooth Camera Models Using Normalized Data.

1. ***Normalization:*** Compute the affine transformations parameters $\{\mathbf{A}, \mathbf{a}, a\}$ and $\{\mathbf{B}, \mathbf{b}, b\}$ from Equations (5.21) and (5.22) respectively. Using the respective affine parameters, compute the normalized image coordinates $\{\mathbf{u}_i\}$ and $\{\mathbf{q}_i\}$ from $\{\mathbf{x}_i\}$ and $\{\mathbf{p}_i\}$ respectively. Using the affine transformation, get the matrix $\mathbf{E}$, using Proposition 3.1;

2. ***Calibration Matrix:*** Compute the calibration matrix $\mathbf{M}$ using Equation (5.11). Note that instead of $\{\mathbf{x}_i\}$ and $\{\mathbf{p}_i\}$, $\{\mathbf{u}_i\}$ and $\{\mathbf{q}_i\}$ should be used, respectively;

3. ***Get Camera Matrix:*** Get $\text{vec}\left(\mathbf{H_{wa}}\right)$ which minimizes $\mathbf{M}\text{vec}\left(\mathbf{H_{wa}}\right)$ – using the *Singular Value Decomposition* [HZ00], and un-stack vector $\text{vec}\left(\mathbf{H_{wa}}\right)$ to matrix $\mathbf{H_{wa}}$;

4. ***Camera Model:*** The camera model with normalization is defined by a function $\mathbf{s} : \mathbb{R}^2 \mapsto \mathbb{R}^6$ where $\breve{\mathbf{l}}\mathbb{R} = \mathbf{s}\left(\mathbf{x}\right)$, such that

$$\mathbf{s}\left(\mathbf{x}\right) = \mathbf{r}\Big(a^{-1}\left(\mathbf{A}\mathbf{x} + \mathbf{a}\right)\Big)\mathbf{H_{wa}}\mathbf{E}^{-T} \tag{5.24}$$

   where $\mathbf{r} : \mathbb{R}^2 \mapsto \mathbb{R}^{(p+3)}$ is defined in Equation (5.9). Note that the coordinate system where line coordinates $\breve{\mathbf{l}}\left(\mathbf{x}\right)$ are represented is the same coordinate system where the original set of point in the world $\{\mathbf{p}_i\}$ is represented.

where $\mathbf{E} \subset \mathbb{R}^{6\times 6}$. Note that $\text{rank}\left(\mathbf{E}\right) = 6$, which mens that $\mathbf{E}$ is invertible and as a resul $\mathbf{l}^{(1)}$ can be estimated such that $\mathbf{l}^{(1)}\mathbb{R} = \mathbf{E}^{-1}\mathbf{l}^{(2)}$.

Camera Model and Calibration with Normalization

To conclude the section we decompose the algorithm for the calibration of *smooth camera model* in four steps. Procedure is shown in the Algorithm 5.1.

Chapter 6

Experiments

To evaluate the proposed *smooth camera model* and its calibration procedure, the use of both synthetic – Section 6.1, and real data – Section 6.2.

6.1 Results with Synthetic Data Sets

In this section, synthetic data sets were generated. Using known camera models, 3D lines are generated, which are mapped from image points. The data set made up by pairings between 3D points and image points is obtained from the first data set (containing coordinates of 3D lines) by computing 3D points incident on the lines. The camera models used in the evaluation were: catadioptric system with quadric mirror, Figure 6.1(a); crossed–slits camera model, Figure 6.1(d); and refraction camera model, made up of a camera looking through a volume of water contained between two parallel planes, Figure 6.1(g).

Since the mappings between points in the image and 3D lines are known, the calibration results can be evaluated by the estimates of the distance errors in line space (distances between lines). Note that the camera model is represented by a mapping between points in the image and 3D lines.

6.1.1 Evaluation Results Using Smooth Camera Models

Synthetic data sets were used to evaluate the effect of the number of *control points* on the error defined as a distance between ground truth 3D lines and estimated 3D lines. As *radial basis functions multi–quadrics* and *exponential* were used.

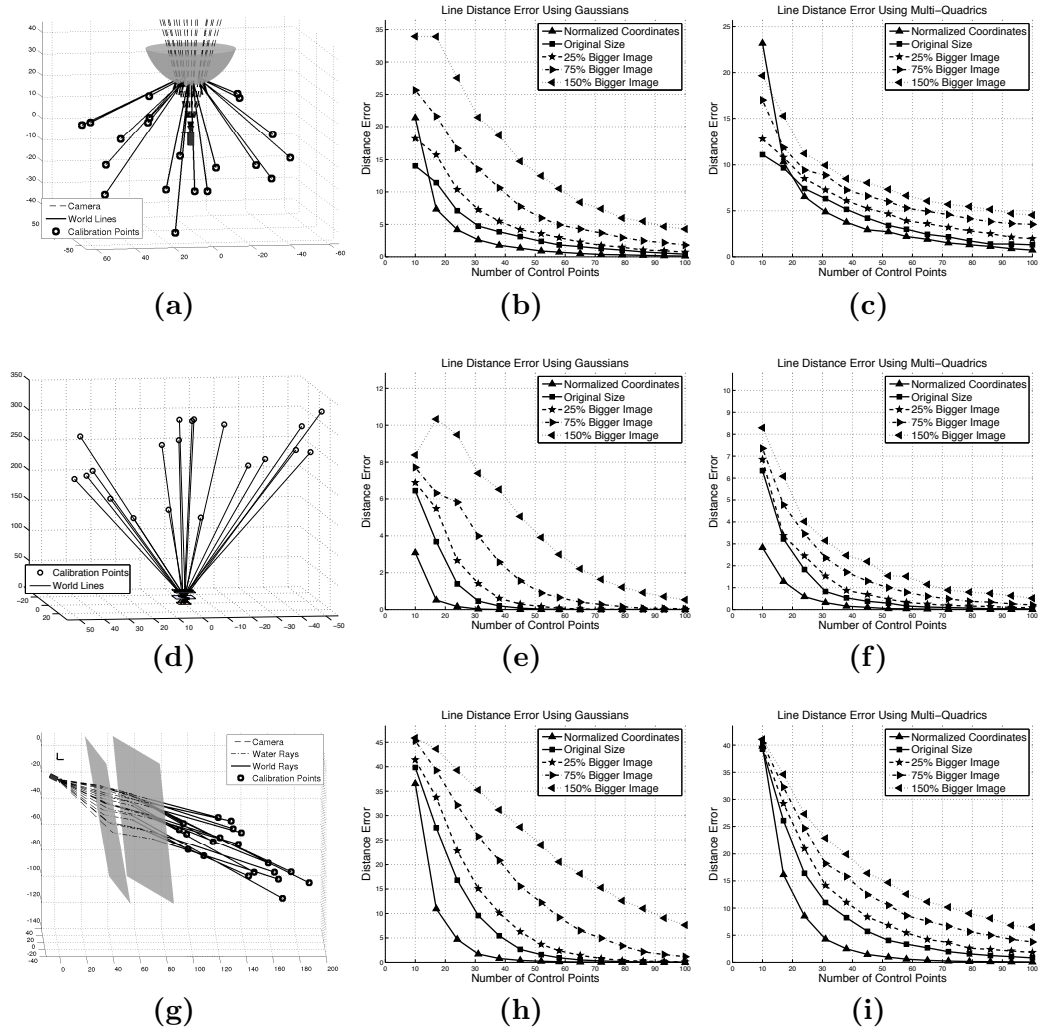


Figure 6.1: *Evaluation of the average line distance error, Equation (3.12) – Section 3.2.2, as a function of the number of control points, for non–central catadioptric systems with a quadric mirror, for a crossed–slits camera model and for a model including refraction (a camera looking through a volume of water contained between two parallel planes) (a), (d) and (g) respectively. For each camera model we evaluate the distance error defined in Equation (3.12), for gaussian and multi–quadrics radial basis functions and the errors are shown in Figures (b)-(c), (e)-(f) and (h)-(i) respectively.*

The synthetic data sets were obtained for the following smooth camera models $\mathbf{f} : \mathbb{P}^2 \mapsto \mathbb{L}^3$: non–central catadioptric systems using quadric mirrors [Gon08], crossed-slits camera model [ZFPW03] and refraction-based

camera model [TSS08, KS08].

The calibration method estimates an interpolant function $\mathbf{s} : \mathbb{R}^2 \mapsto \mathbb{R}^6$ that should fit an imaging model, defined by a function $\mathbf{f} : \mathbb{P}^2 \mapsto \mathbb{L}^3$, where $\mathbf{f}$ is an analytic representation of the corresponding imaging model.

$\mathbf{f}$ is used to generate a data set of $\{\mathbf{x}_i \mapsto \mathbf{p}_i\}$, for $i = 1, \ldots, N$ and P *control points* are selected from the set $\{\mathbf{x}_i\}$, with the conditions described in Section 5.2.2 being met.

Using $\mathbf{f}$, a ground truth data set is generated $\{\mathbf{y}_i \mapsto \mathbf{l}_i\}$, where $\mathbf{y}_i$ are the ground truth image coordinates and $\mathbf{l}_i$ are the ground truth 3D line coordinates.

The sets $\{\mathbf{x}_i \mapsto \mathbf{p}_i\}$ and $\{\mathbf{c}_i\}$ are used to calibrate the camera model, Algorithm 5.1. Using the interpolant functions $\mathbf{s}$, the set $\left\{\mathbf{y}_i \mapsto \breve{\mathbf{l}}_i\right\}$ is estimated, where $\breve{\mathbf{l}}_i = \mathbf{s}(\mathbf{y}_i)$. To evaluate the calibration method the distances between $\breve{\mathbf{l}}_i$ and $\mathbf{l}_i$ are used. To characterize the error the average of the distances is computed. The metric used for the computation of the distance between two lines $\delta\left(\mathbf{l}_i, \breve{\mathbf{l}}_i\right)$ is briefly described in Section 3.2.2 – see Equation (3.12).

The results are evaluated by varying the number of *control points*, P, from 10 to 100. For each number of *control points*, the calibration is evaluated using the ground truth data set $\{\mathbf{y}_i \mapsto \mathbf{l}_i\}$ for $i = 1, \ldots, 120$. For each set of *control points*, the calibration is repeated 150 times for different values of $\{\mathbf{x}_i\}$, $\{\mathbf{y}_i\}$ and $\{\mathbf{p}_i\}$, $\{\mathbf{l}_i\}$, respectively. The average of the distance errors is computed for each number of control points.

The calibration results are dependent on the value of the shape parameter in the case of either *multi–quadric* or *gaussian radial basis functions*. By normalizing the data coordinates as described in Section 5.2.3, the value of the shape parameter is no longer dependent on the image size and also on the spatial distribution of the image points.

For a fixed value of the shape parameter, the error as a function of the spatial distribution of the set $\{\mathbf{x}_i\}$ was evaluated. For that purpose and for each set $\{\mathbf{x}_i\}$ and $\{\mathbf{y}_i\}$, $\left\{\mathbf{x}_i^{(j)}\right\}$ and $\left\{\mathbf{y}_i^{(j)}\right\}$ were computed with

$$\mathbf{x}_i^{(j)} = \mathbf{x}_i \sigma_j \quad \forall i \tag{6.1}$$

for $j = 1, 2, 3$ with $\sigma_1 = 1.25$, $\sigma_2 = 1.75$ and $\sigma_3 = 2.5$. The results are shown in Figure 6.1. Note that by using normalized coordinates the error variation does not depend on the image size. Therefore the distance error can be represented by a single curve.

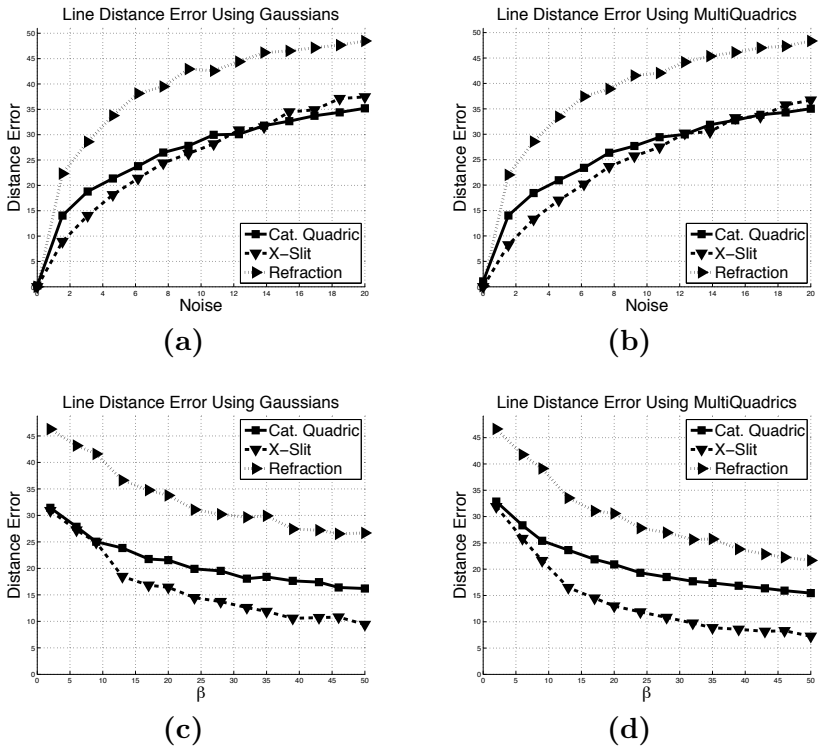
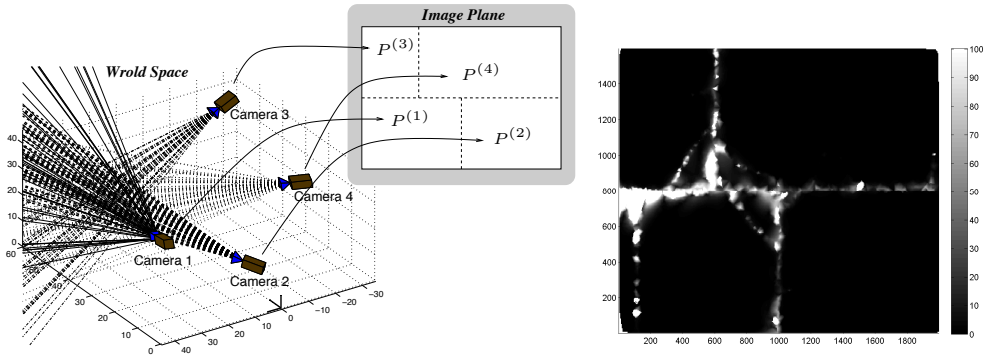


Figure 6.2: *In this figure we show results of tests performed with the three previously mentioned camera models, Figure 6.1, with noise added. We set the number of control points to* 60 *and vary the variance of the noise added to the coordinates of the 3D points used in the calibration procedure. The standard deviation of the noise is proportional to the smallest distance among all the world points of the calibration data set. (a) is the distance error in the line space for gaussian RBF and (b) for multi–quadrics. To eliminate some errors, we can use more points than the ones needed for the minimal solution. We consider the noise such that* $\alpha = 0.10$ *– Equation (6.2); and we vary the* β*, where* $N = \beta P$*. The results are shown in Figures (c) and (d) for gaussian and multi–quadrics respectively.*

Results with Noise Added

The same evaluation procedure was repeated by adding Gaussian noise $\{\boldsymbol{\mu}_i\}$ to the 3D coordinates of the points used in the calibration procedure $\{\mathbf{p}_i\}$. For that purpose the standard deviation of the norm of the noise vector was defined as

$$\text{std}\left(||\boldsymbol{\mu}_i||\right) = \alpha e \tag{6.2}$$



(a) **(b)**

Figure 6.3: *This figure shows results for a non–smooth camera model synthetically generated, and made up by combining four images obtained from four different perspective cameras as shown in (a). In figure (b) errors corresponding to the distances between the 3D lines are shown.*

where $e = \min(||\mathbf{p}_i - \mathbf{p}_j||)$ for all $i \neq j$. Instead of using the set $\{\mathbf{x}_i \mapsto \mathbf{p}_i\}$ to calibrate, the set $\{\mathbf{x}_i \mapsto \breve{\mathbf{p}}_i\}$, where $\breve{\mathbf{p}}_i = \mathbf{p}_i + \boldsymbol{\mu}_i$ for all i, was used.

The errors are evaluated in line space (Equation (3.12)) for the three previously mentioned camera models, with values for α from zero to 0.20. Calibration was performed using normalized coordinates (Algorithm 5.1) with $P = 60$. Results are shown in Figure 6.2(a) and 6.2(b) for *gaussian* and *multi–quadrics* RBF, respectively.

Note that in the previous experiments, $N = 2P$ was used, for P equal to the number of *control points*. On the other hand if $N = \beta P$ for $\beta > 2$, then matrix $\mathbf{M}$ can become *full–column rank* and, as a consequence, there will be no exact solution for $\mathbf{H_{wa}}$. However, an estimate for $\text{vec}(\mathbf{H_{wa}})$ can still be obtained, in the least–squares sense [HZ00].

The errors for β from 2 to 50 were also evaluated (measured as the distances between the 3D lines corresponding to the ground truth and the estimated 3D lines). For that purpose, $P = 60$ was used, and Gaussian noise with $\alpha = 0.10$, Equation (6.2) was added.

The results are shown in Figure 6.2(c) and 6.2(d) for *gaussian* and *multi–quadrics* RBF, respectively.

6.1.2 Results Using Non–Smooth Camera Models

The approach described in this thesis is based on the assumption that the camera model is smooth. Therefore this method can not be applied to non-smooth camera models – where line positions and orientations do not change

smoothly. However, and even for some of those camera models, the approach can still be applied, provided that errors at the discontinuities can be accepted.

As an example the calibration procedure was applied to a non–central camera model obtained by combining images acquired by four different perspective cameras, with different poses (Figure 6.3(a)). The errors in 3D line space (Equation (3.12)), displayed as a function of the image coordinates are shown in Figure 6.3(b). These results were obtained for $P = 250$ and *multi–quadrics* RBF. The errors are small over most of the global image with the exception of the boundaries between the individual images.

One of the of the main issues in the estimation of the lines close to the boundaries is the contribution of the neighboring points belonging to other clusters. As a result, if the clusters are known, contributions from pixels belonging to different clusters can be suppressed. The set of *radial basis functions* that constitutes the interpolant function ($\phi\left(||\mathbf{x}-\mathbf{c}_i||\right)$ for all i at Equation (5.1)) can be assigned to each cluster and a new *radial basis function* can be defined such that

$$\breve{\phi}\left((||\mathbf{x}-\mathbf{c}_i||\right) = \begin{cases} 0, & \text{for } \mathbf{x} \text{ out of the cluster } j \\ \phi_j\left(||\mathbf{x}-\mathbf{c}_i||\right), & \text{for } \mathbf{x} \text{ inside the cluster } j \end{cases} . \tag{6.3}$$

The association of functions $\phi\left(||\mathbf{x}-\mathbf{c}_i||\right)$ to cluster j can be performed by choosing the control points $\mathbf{c}_i$ so that they belong to the same cluster j.

To obtain improved results, the image model we can re-calibrated using the specific *radial basis functions* corresponding to each cluster. $\{\mathbf{x}_i\}$ also have to be associated to a cluster. Note that we still have a single function to represent the all image model; however, this function is not continuous any more.

6.1.3 Experimental Results Using 3D Data from a Single Surface

In this section we present the results of a new calibration experiment where the 3D points data used for the calibration belong to a single surface. Synthetic data was obtained from a non-central catadioptric system made up by a quadric mirror (Figure 6.1(a)). The catadioptric system was used to generate the data for the calibration, where it was assumed that the correspondences between the world points and their images were known.

We consider three cases: world points on a spherical surface – Figures 6.4(a)-(c); world points on a "step", two non-overlapping planes – Fig-

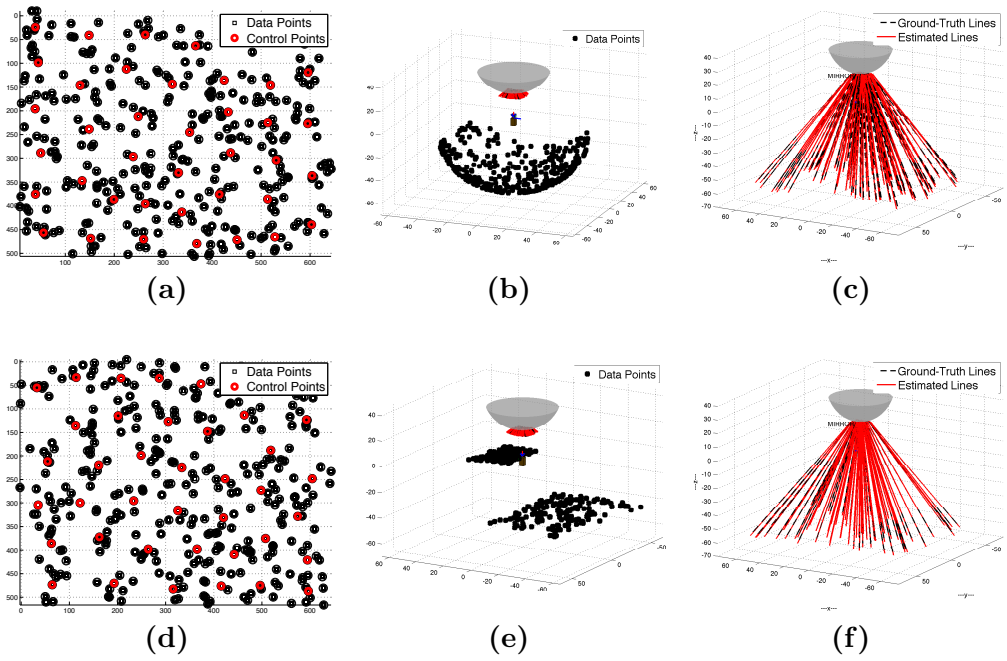
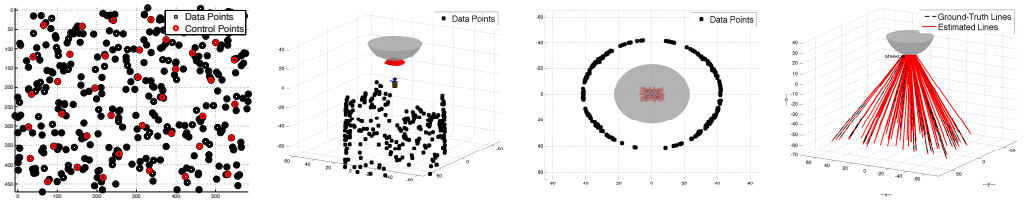


Figure 6.4: *For 3D calibration points on a spherical surface, Figure (a), displays the image points (black squares) and control points (red dots) used in the calibration process. Their correspondent world points are shown in Figure (b) (black dots). Figure (c) shows the results, where the ground–truth lines are shown in black and the estimated lines in red. The same experiments were performed with the calibration points belonging to a "step" surface and the results are presented in Figures (d) (e) and (f). In both cases we use* 40 *control points and* 360 *data points.*

ures 6.4(d)-(f) (note that in all the case the rays of the image model do not pass through both planes i.e., each ray is associated to a single 3D point); and the case where all the world points belong to a cylinder surface – Figures 6.5(a)-(d). Note that, according to the theoretical conditions described in Appendix A, the 3D points can not belong to a single plane or to a single 3D line. In these cases, the null space of the calibration matrix $\mathbf{M}$ has always dimension greater than one.

For all the cases, we use the a non-minimal solution, with 40 control points and 360 data points. The *camera matrix* is given in the least-squares sense [HZ00]. From the experiments, lines are estimated with small error.



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

Figure 6.5: *This figure shows the calibration of a catadioptric camera model, where the 3D points used in the calibration belong to a cylindrical surface. Figure (a), displays the image points (black dots) and control points (red dots) used in the calibration process. Their correspondent world points are shown in Figures (b) and (c) (black dots). Figure (d) shows the results, where the ground–truth lines are shown in black and the estimated lines in red. We use* 40 *control points and* 360 *data points.*

6.2 Results with Data sets of Real Images

For experiments with real data sets, three different types of imaging systems were calibrated, using the model described in this article and point-based calibration – Algorithm 5.1. A perspective camera and two different catadioptric systems were used, which images are shown in Figures 6.6(a), 6.6(d) and 6.6(g) respectively. Note that the catadioptric camera model with two planar mirrors is a non–smooth camera model.

To acquire the point correspondences $\{\mathbf{x}_i \mapsto \mathbf{p}_i\}$, which is the data necessary for the calibration method, a chess board was used. Infrared (IR) LEDs were attached to the chess board and their positions in the world were measured using an IR tracker [Nor09]. This tracker has an accuracy of 0.1[mm] and a resolution of 0.01[mm].

Each corner of the chess board image $\mathbf{x}_i$ is associated to a position in the world $\mathbf{p}_i$, which is given by its corresponding position in the 3D chess board. The set of associations $\{\mathbf{x}_i \mapsto \mathbf{p}_i\}$, for $i = 1, \ldots, 3840$, was used in the calibration procedure. An example of the images that we used to get the calibration data is shown in Figure 6.7.

The set of image points $\{\mathbf{x}_i\}$ are the yellow points in Figures 6.6(b), 6.6(e) and 6.6(h). The set of world points $\{\mathbf{p}_i\}$ are the yellow points in Figures 6.6(c), 6.6(f) and 6.6(i). *Control points* $\{\mathbf{c}_i\}$ are chosen as a subset of $\{\mathbf{x}_i\}$. They are shown as green points, in Figures 6.6(b), 6.6(e) and 6.6(h).

Using $\{\mathbf{x}_i \mapsto \mathbf{p}_i\}$, the interpolant function $\mathbf{s}$ is estimated, using the proposed algorithm – Algorithm 5.1. Since in the real experiments we have

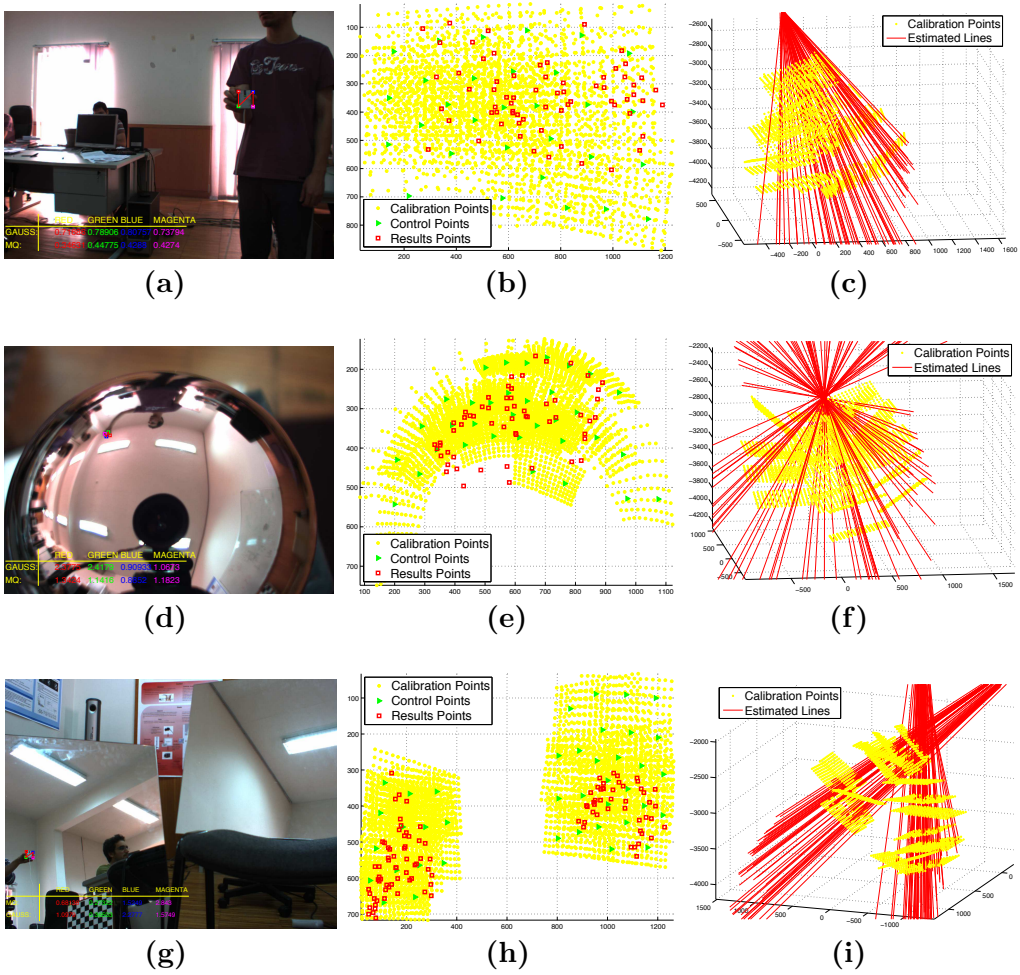


Figure 6.6: *Results with real images, obtained using the method described for three types of camera models: perspective camera (a), catadioptric system with spherical mirror (d) and catadioptric system with two planar mirrors (g). Yellow points in (b), (e) and (h) are the set of image coordinates* $\{\mathbf{x}_i\}$ *and yellow points in (c), (f) and (i) are the corresponding set of world points* $\{\mathbf{p}_i\}$*, used in the calibration method. Red squares in the 2D plot and the red rays in the 3D plot are the a subset of* $\left\{\mathbf{y}_i \mapsto \check{\mathbf{l}}_i\right\}$ *obtained with multi–quadrics RBF. Green points in (b), (e) and (h), are the corresponding set of control points used in the calibration.*

$N > 2P$, the solution will be obtained solving an over–determined system of equations. The *camera matrix* is estimated as least–squares solution for the homogeneous equations [HZ00].

To evaluate the calibration, a different set of 3D coordinates, correspond-

Figure 6.7: *In this figure we show five examples of images taken from the chess board used to get the calibration data sets. The first row of images corresponds to the perpective camera; the second row corresponds to the non-central catadioptric camera that uses a spherical mirror; and the last row corresponds to the non-central camera formed by two planar mirrors.*

ing to a different object were used. The IR tracker has a "test object" (also with LEDs) which is provided to enable the estimation of 3D coordinates of points. This different object was used to generate a new data set $\{\mathbf{y}_i \mapsto \mathbf{w}_i\}$, for $i = 1, \ldots, 700$, where $\mathbf{y}_i$ are image points and $\mathbf{w}_i$ are world points. This new data set was used to evaluate the calibration performed with the former data set.

The distance error is defined by the geometric distance from the world point $\mathbf{w}_i$ to the line estimated $\breve{\mathbf{l}}_i$, where $\breve{\mathbf{l}}_i = \mathbf{s}(\mathbf{y}_i)$. A subset of $\{\mathbf{y}_i\}$ and the corresponding lines $\left\{\breve{\mathbf{l}}_i\right\}$ are shown as red squares, in Figures 6.6(b), 6.6(e) and 6.6(h), and as red lines in Figure 6.6(c), 6.6(f) and 6.6(i), respectively. In Section 3.3.2, we derived the geometric distance between a line (in *Plücker* coordinates) and a point in the world as $\delta\left(\breve{\mathbf{l}}_i, \mathbf{w}_i\right)$ – see Proposition 3.2.

The number of *control points* used in the calibration were 30, 35 and 40, for the imaging systems of Figures 6.6(a), 6.6(d) and 6.6(g) respectively. The results for the means and standad deviations, when using *gaussians* and *multi-quadrics*, are shown in Table 6.1.

RBF	*Gaussian* [cm]	*Multi–quadrics* [cm]
Projective	0.93058 ± 1.9783	0.54711 ± 0.43567
Sphere mirror	2.8743 ± 2.1074	1.8863 ± 1.2379
Plane mirrors	0.91795 ± 1.0608	0.99492 ± 0.78778

Table 6.1: *Results for the experiments, using the IR tracker [Nor09].*

6.3 Using a Calibrated Perspective Camera to Acquire a Data-set with Real Data

To further evaluate the calibration method, additional data-sets of real data were acquired.

Consider an object or scene whose images are acquired by two perspective cameras. One of the cameras is looking directly at the object/scene, whereas the second camera acquires images of the object by viewing it through a glass tank filled with water – see the scheme in Figure 6.8(a). The goal is to calibrate the second imaging system, i.e., including the effect of viewing the object through the glass tank filled with water. An example of the image acquired by this system is shown in Figure 6.8(b).

In addition, we consider the same technique to calibrate a catadioptric camera formed with a perspective camera and a spherical mirror – Figure 6.8(d). An example of a image acquired by the catadioptric camera is shown in Figure 6.8(e).

A chess board is used to calibrate both systems. Several images of the chess board were acquired, with the chess board in different positions and orientations. A set of examples of matched images used for the calibration of both imaging systems is shown in Figure 6.9. The direct images of the chess board, i.e., acquired by the camera that views the scene directly (not through the glass tank nor through the mirror) were used to estimate the coordinates of the planes in the world coordinate system. For that purpose the Bouguet calibration toolbox was used [Bou10].

Eighteen different images of the chess board in different positions and orientations were used. In each chess board image there were of 160 points whose coordinates could be used (these points are located at the intersection of the edges). The 3D points are displayed in Figures 6.8(c) and Figures 6.8(f) as yellow and blue dots. The coordinates of the points on the chess board are used as world points forming the set $\left\{\mathbf{p}_i^{(j)}\right\}$ where $\mathbf{p}_i^{(j)}$ are the coordinates of the i^{th} point of the j^{th} chess position in the world.

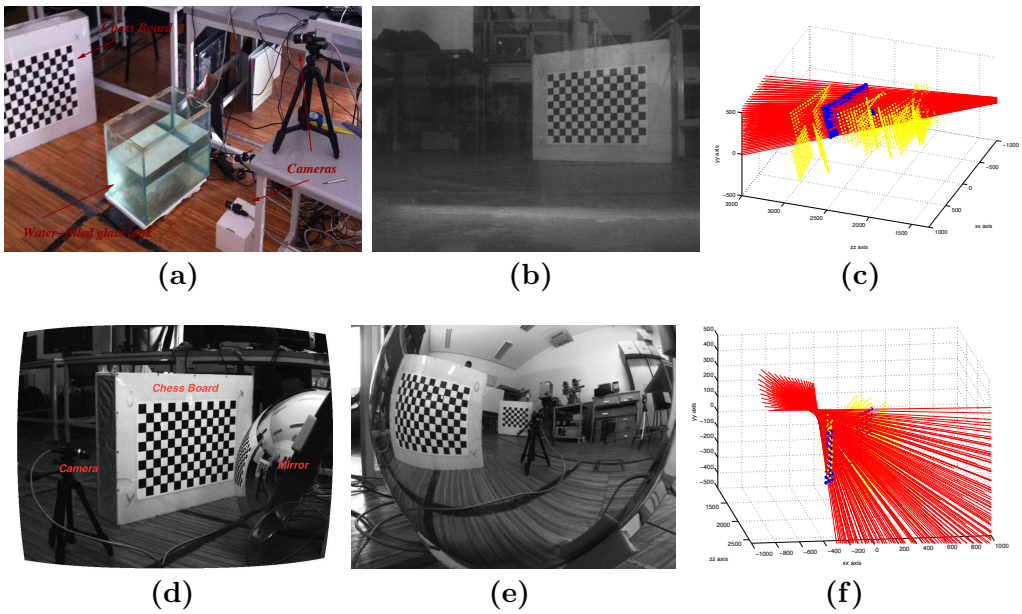
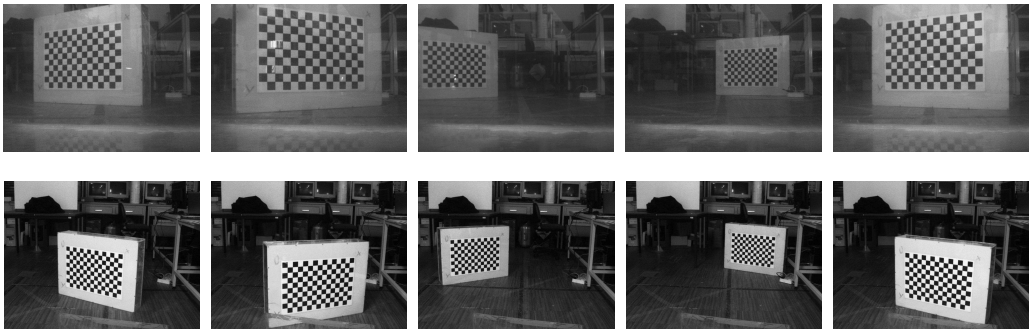


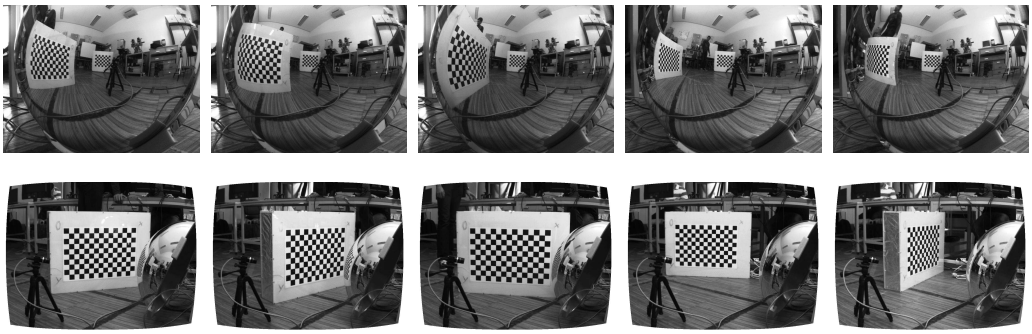
Figure 6.8: *In Figure (a), we display the set-up used to acquire the images of the data set for the case where the goal is to calibrate the camera model that views a scene through a water-filled glass tank. In Figure (d), we show the perspective camera and the spherical mirror that form the non-central catadioptric camera. Examples of images acquired by both non-central imaging system are shown in Figures (b) and (e). A planar chess board was used. The data set is made up by the coordinates of the images of the points corresponding to the intersections of the chess board edges (seen through the tank). Examples of the calibration results are shown in Figures (c) and (f). In yellow we show all the 3D points used in the calibration procedure. In blue we show the ground truth 3D points on which the estimated lines must be incident. The blue 3D points correspond to the chess board edges of the images Figures (b) and (e) respectively. The estimated lines (according to the method proposed in the thesis) are shown in red. The units in both Figures (c) and (c) are millimeters.*

Since these points are also are visible in the image acquired through the water-filled glass tank / mirror, the images of the points on of each chess-board images $\left\{\mathbf{x}_i^{(j)}\right\}$ (where $\mathbf{x}_i^{(j)}$ is the i^{th} point coordinate of the j^{th} image of the j^{th} chess board position) are associated with the corresponding world coordinates $\left\{\mathbf{p}_i^{(j)}\right\}$, forming the set of mappings $\left\{\mathbf{x}_i^{(j)} \mapsto \mathbf{p}_i^{(j)}\right\}$.

To test the calibration method the following procedure was followed: for each specific chess board position and orientation j, all the data from all other



(a) The first row corresponds to the images taken from the rafraction camera. In the second row we shows the correspondent images, acquired from the perspective camera that sees the scene directly.



(b) The first row shows the images acquired from the catadioptric camera. The second row shows the correspondent images taken directly from the scene.

Figure 6.9: *In this figure we show a set of examples of images that we used to generate the data set. In Figure (a) we show the case where the goal is to calibrate a refraction camera. In Figure (b) we show images used for the catadioptric case.*

chess boards at different positions and orientations were used as calibration data set. The coordinates of the points of the j^{th} chess board position were used as as ground truth. Therefore the coordinates of the points of the j^{th} chess board position were used as ground truth data set $\{\mathbf{y}_i \mapsto \mathbf{w}_i\}$ (160 mappings between image and world points) and the coordinates of the other points of the other chess board positions were used as calibration data–set $\{\mathbf{x}_i \mapsto \mathbf{p}_i\}$ (2720 mappings between image and world points). For each $\mathbf{y}_i$, the distances between the estimated lines $\breve{\mathbf{l}}_i$ and the corresponding points in the world $\mathbf{w}_i$ are estimated $\delta\left(\breve{\mathbf{l}}_i, \mathbf{w}_i\right)$, using Proposition 3.2.

This procedure is repeated for all the j^{th} chess board positions and the means and standard deviation of the distance errors are computed, for both *gaussian* and *multi-quadrics* RBFs. The number of *control points* used was 10. The results are shown in the Table 6.2.

RBF	*Gaussian* [cm]	*Multi–quadrics* [cm]
Refraction Water	0.313 ± 0.545	0.111 ± 0.075
Sphere Mirror	0.936 ± 2.477	0.289 ± 0.207

Table 6.2: *Results for the experiments, using the Perspective Camera to Acquire a Data-set with Real Data.*

Examples of the estimated lines, using the proposed method (for a specific plane position) are shown in Figures 6.8(c) and 6.8(f) as red lines, for the refraction based and reflection based imaging systems respectively.

6.3.1 Removing Distortion: Results for a Spherical Catadioptric System

In general, undistorted views cannot be computed from images acquired by a general non-central camera models (without any additional constraint). As described by Swaminathan *et al.* at [SGN03], obtaining undistorted images from general non-central systems requires the knowledge of the scene structure. On the other hand, in the case of central camera models (and of special non-central systems), the undistorted image is given by the camera model. The image model proposed in the thesis can be applied to both central and general non–central camera models. Swaminathan *et al.* proposed two approaches to obtain undistorted images from a general non-central catadioptric system. Also two metrics to measure distortion are presented. Both approaches are based on an estimate of an approximation of the scene structure, which is obtained using a so-called "global view map". As a result views with minimal distortions can be obtained. However these views are always affected by caustic distortions since the removal of these distortions require the knowledge of the scene structure.

We have performed a new experiment using a non-central catadioptric system made up by a spherical mirror. As was pointed out by Swaminathan *et al.* , if we know the scene structure we can get undistorted images. Using the approach proposed in that paper (and the knowledge of the scene structure) we performed a new experiment to estimate undistorted images from the non-central catadioptric camera using a spherical mirror.

We use the same calibration procedure described in Section 6.3. We know position of the seventeen checkerboards and use sixteen of these checkerboard positions to calibrate the image model. We then obtained the undistorted image of the 17^{th} plane. Using the knowledge of the position of the plane not

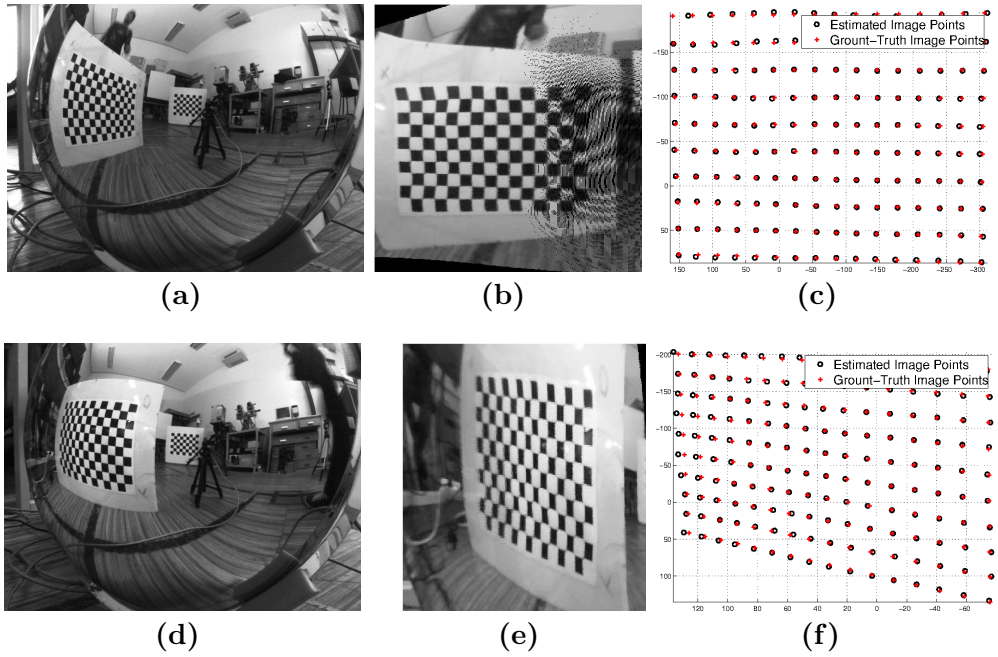


Figure 6.10: *In this experiment we show two examples of results obtaining an undistorted image from a non–central catadioptric camera, using the knowledge of the scene structure (in this case a plane). Figures (a) and (d) show the original images. Figures (b) and (e) show the images of the same plane after removal of the distortion. In addition, we show the disparities between estimated and ground–truth image pixels of the corners of the checkerboard (Figures (c)) and (f))*

used in the calibration, and its image, the undistorted image was computed.

Using a virtual perspective camera (located at $(0, 0, 0)$), we generated the undistorted image. The image is obtained by projectig the points in the world that result from the intersection between the estimated rays and the known plane coordinates. The results for two examples are presented in Figure 6.10. As it can be seen from the pairs of images Figures 6.10(a) and 6.10(d) and Figures 6.10(b) and 6.10(e) the curved lines in the distorted image are mapped into straight lines in the undistorted image.

In addition and since we know the coordinates of the checkerboard in the world, we can project the corners onto the virtual image (which will define the ground truth) and compute the disparity between these image points and the correspondent image points of the estimated undistorted image. The results are shown in Figures 6.10(c) and Figures 6.10(f) for plane positions that correspond to Figures 6.10(a) and 6.10(d) respectively. The means and

standad deviations for the disparities are 1.8 ± 1.2 and 1.5 ± 0.9 pixels for the cases of Figures 6.10(c) and Figures 6.10(f) respectively.

Chapter 7

Discussion

The experiments and results with synthetic data were specially relevant for the goals of this paper. They were particularly important to evaluate the effect of the number of control points as well as to evaluate the sensitivity of the method to the noise, by means of a fully specified data set.

Considering the experimental results obtained without noise Figure 6.1, those obtained using *gaussian* RBF are slightly better than those obtained using *multi–quadrics*. However, *multi–quadrics* RBF tend to be less sensitive to the image scale variation. The experimental results also show that the inclusion of the coordinate normalization procedure tends to consistently yield better results, as expected. When noise is added to the synthetic data, and as expected, the use the least-squares estimates yields better results than the ones obtained applying the theoretical condition $N = 2P$.

In some cases the calibration is affected with errors that go up to 60 units. These are extreme cases and we discuss the reasons why they occur. In one case (Figure 6.1) these errors correspond to results obtained with a very small number of control points 10. Note that this number of control points is small for the specific camera model (for other systems it might be adequate). Therefore in that case the reason for the error value is the small number of control points. In the case of Figure 6.2 these values are obtained for levels of noise with standard deviations which go up to 20 units. These are presented to show the performance of the model in very special (and unrealistic) conditions.

Experiments with synthetic data also show that this method can be applied to the calibration of non–smooth camera models if the discontinuities on the data are limited.

To conclude the synthetic experiments, we perform an experiment where (with the appropriate choices for the interpolant paramenters) we can calibrate a camera model using world points in a single surface, Fig-

ures 6.4 and 6.5. These experiments show that the method can be used without explicitly accounting for the collinearity constraints.

Experimental results with real data–sets were also performed to evaluate the behavior of the method under real conditions. The results shown in Figures 6.6 and 6.8 show the expected behavior of the $3D$ line estimates obtained under the assumption of a *smooth camera model*.

In addition, we also show distortion correction results for a non–central catadioptric camera, using the proposed camera model.

Part III: Minimal Pose Problem for General Camera Models

Contents

8 Introduction **52**
8.1 Our Approaches 53

9 Formalization – Method 1 **55**
9.1 Decomposition of the *Homography* Matrix 56
9.2 Proposed Approaches 58
9.2.1 Minimal Absolute Pose Problem for General Cameras . 59
9.2.2 Minimal Absolute Pose Problem for Central Cameras . 62
9.3 Algorithm Outline 64

10 Formalization – Method 2 **67**
10.1 Proposed Approach 68
10.1.1 Minimal Absolute Pose for General Camera Models . . 69
10.1.2 Minimal Absolute Pose for Central Camera 70
10.1.3 Recovery of the Pose 73
10.2 Algorithms 74

11 Experiments **76**
11.1 Numerical Errors 78
11.2 Number of Solutions 80
11.3 Computation Time 80
11.4 Experiments with Noise 82
11.5 Critical Configurations 82

12 Discussion **85**
12.1 Analysis for the General Case 85
12.2 Analysis for the Central Case 88

Chapter 8

Introduction

In this part we address the minimal absolute pose problem. This problem consists in the estimation of the rotation and translation parameters that define the rigid transformation between three 3D points in the world coordinate system and three incident straight lines in the camera coordinate system. To handle any type of camera – central and non-central cameras, we consider the general camera model, Section 2.1.

To distinguish betweeen features in the world and camera coordinate systems, we use the superscripts $(\mathcal{W})$ and $(\mathcal{C})$ respectively.

The usual approach to solve this problem is to estimate the 3D points, in the camera coordinate system. Let us consider that the line is given by a point $\mathbf{x}_i^{(\mathcal{C})}$ and a direction $\mathbf{d}_i^{(\mathcal{C})}$, in the camera coordinate system. In this representation, any point $\mathbf{p}_i^{(\mathcal{C})}$ that belongs to the line is given by

$$\mathbf{p}^{(\mathcal{C})} = \mathbf{x}_i^{(\mathcal{C})} + \alpha_i \mathbf{d}_i^{(\mathcal{C})}, \quad \text{for some } \alpha_i \in \mathbb{R}. \tag{8.1}$$

Since the distance between 3D points must be preserved, we can define the constraints

$$\left|\left|\mathbf{p}_i^{(\mathcal{C})} - \mathbf{p}_j^{(\mathcal{C})}\right|\right| = \left|\left|\mathbf{p}_i^{(\mathcal{W})} - \mathbf{p}_j^{(\mathcal{W})}\right|\right|. \tag{8.2}$$

The minimal case consists in the use of three points in the world coordinate system – $\mathbf{p}_i^{(\mathcal{W})}$, for $i = 1, 2, 3$. Using these points and from Equation (8.2), we generate three constraints. Since the distances $\left|\left|\mathbf{p}_i^{(\mathcal{W})} - \mathbf{p}_j^{(\mathcal{W})}\right|\right|$ are known and replacing $\mathbf{p}_i^{(\mathcal{C})}$ in Equation (8.2) using the Equation (8.1), we derive three constraints $g_{i,j}\left(\alpha_i, \alpha_j\right) = 0$ as

$$g_{i,j}\left(\alpha_i, \alpha_j\right) = \kappa_1^{(i,j)}\alpha_i^2 + \kappa_2^{(i,j)}\alpha_j^2 + \kappa_3^{(i,j)}\alpha_i\alpha_j + \kappa_4^{(i,j)}\alpha_i + \kappa_5^{(i,j)}\alpha_j + \kappa_6^{(i,j)}. \tag{8.3}$$

The estimation of the dephs α_1, α_2 and α_3 is computed using

$$g_{1,2}(\alpha_1, \alpha_2) = g_{1,3}(\alpha_1, \alpha_3) = g_{2,3}(\alpha_2, \alpha_3) = 0 \tag{8.4}$$

which correspond to finding the points where three axis–aligned cilinders intersect. This formulation was proposed by both Chen & Chang [CC02, CC04a] and Ramalingam *et al.* at [RLS06]. In both articles the authors claim that, from the three constraints, it is possible to derive an eighth degree polynomial equation. Moreover, Chen and Chang proposed a set of transformations of the data set which will make the computation of the eight degree polynomial coefficients easier.

Since pose is given by the rotation and translation parameters that define the transformation between the world and camera coordinates, the computation the rotation and translation from two 3D points sets is required. If we use the convencional methods such as [AHB87, Ume91], a *single value decomposition* is required for each valid solution.

Nistér and Stewénius at [Nis04a, NS07] expresses rotation and translation parameter as a problem of intersection between a ruled quartic surface and a circle. Differently of Chen and Chang method, Nistér and Stewénius's outputs the rotation and translation parameters that represent the pose. However, for a simple problem such as the minimal absolute pose problem their formulation is quite complex and less intuitive.

This part is organized as follows: in the remainder of this section we briefly describe our approaches; in Chapter 9 and Chapter 10, we formulate two approaches to solve the problem; in Chapter 11 we evaluate the proposed approaches which are discussed in Chapter 12.

8.1 Our Approaches

In this thesis we propose a novel alternative solution to the Chen & Chang and Nistér & Stewénius algorithms.

The minimal absolute pose problem consists in the estimation of a transformation between the world and camera coordinatas, using the coordinates of three known points in the world and their correspondent images. These three 3D points define a plane. As a result, the *homography* map can be used to represent the transformation between the world and camera coordinates. The aim of the proposed algorithms are to estimate the *homography* matrix [HZ00, MSKS04].

Using the *homography* to represent the pose, we derived an algebraic relation that estimates a three degrees of freedom solution for the *homography*

matrix. To have a finite number of solutions, three additional constraints are required. We use the fact that *homography* matrix must represent a rotation to define these constraints. After the estimation of the *homography* matrices that define the solutions for the pose, they are decomposed into rotation and translation, concluding the approaches.

In the first approach, we derived a direct solution – no predefined transformation to the data-set is required. The estimation of the *homography* matrix is very simple, direct and intuitive. However, the decomposition of the matrix is complex requiring an *eigen decomposition* of a 3×3 matrix. Despite the fact that there is a closed-form solution for this decomposition, the distribution of the numerical errors becomes significantly worse. As a result, in the literature, authors use iterative methods to compute the decomposition [KSS11].

In the second case, we used a predefined transformation of the 3D points, in the world coordinate system. The goal of the suggested transformation was to express the *homography* matrix such that the first two columns must be orthonormal – they must correspond to the two first columns of the rotation matrix. In this case the difficulty of the decomposition of the *homography* matrix is significantly reduced. The decomposition is complete after two analytical operations.

When considering the proposed approaches compared to the state–of–the–art methods, the main results of the proposed formulation are:

- A simple and intuitive solution that can be easily implemented – specialy when compared to the Nistér and Stewénius algorithm;
- Returns the rotation and translation parameters that define the pose – when compared to Chen and Chang and Ramalingam *et al.* , does not require addicional steps for the computation of the rotation and translation from two 3D points sets;
- When compared to previous methods, our formulation is more robust to both degenerate and critical cases.

Chapter 9

Formalization – Method 1

Pose estimation requires the estimation of a rotation matrix $\mathbf{R} \in \mathcal{SO}(3)$ (special orthogonal group) and a translation vector $\mathbf{t} \in \mathbb{R}^3$ that define the rigid transformation between the world and camera coordinate systems. Since we consider that the imaging device is calibrated, pose is specified by the rigid transformation that satisfies the relationship of incidence between points in the world coordinate system and 3D straight lines represented in the camera coordinate system, Figure 2.2.

The rigid transformation between a point in world coordinates $\mathbf{p}^{(\mathcal{W})}$ and the same point in camera coordinates $\mathbf{p}^{(\mathcal{C})}$ is given by

$$\mathbf{p}^{(\mathcal{C})} = \mathbf{R}\mathbf{p}^{(\mathcal{W})} + \mathbf{t}. \tag{9.1}$$

Since three points define a plane, we can define a plane $\mathbf{\Pi}^{(\mathcal{W})} \in \mathbb{R}^4$ such that

$$\mathbf{\Pi}^{(\mathcal{W})} \doteq \mathbf{p}_1^{(\mathcal{W})} \cup \mathbf{p}_2^{(\mathcal{W})} \cup \mathbf{p}_3^{(\mathcal{W})}, \tag{9.2}$$

and from the planar *homography* [MSKS04], we can rewrite Equation (9.1) as

$$\mathbf{p}^{(\mathcal{C})} = \underbrace{\left(\mathbf{R} + \frac{1}{\zeta^{(\mathcal{W})}}\mathbf{t}\,\boldsymbol{\pi}^{(\mathcal{W})\,T}\right)}_{\mathbf{H}} \mathbf{p}^{(\mathcal{W})} \tag{9.3}$$

where $\mathbf{H} \in \mathbb{R}^{3\times3}$ is called the *homography* matrix, $\zeta^{(\mathcal{W})}$ and $\boldsymbol{\pi}^{(\mathcal{W})}$ are the distance from the plane to the origin and the unit normal vector to the plane $\mathbf{\Pi}^{(\mathcal{W})}$ respectively.

The *homography* matrix can be defined up to a scale factor $\breve{\mathbf{H}}\mathbb{R} = \mathbf{H}$. However, $\mathbf{H}$ only defines a rigid transformation for points $\mathbf{p}^{(\mathcal{W})} \in \mathbf{\Pi}^{(\mathcal{W})}$ if it

verifies the following condition:

Proposition 9.1. *A matrix* $\mathbf{H}$ *belongs to the space of the homography matrices* $\mathcal{H}$*, that preserve rigidity for points* $\mathbf{p}^{(\mathcal{W})} \in \mathbf{\Pi}^{(\mathcal{W})}$*, if and only if it verifies following condition*

$$\mathbf{H} \in \mathcal{H} \doteq \left\{ \mathbf{H} \in \mathbb{R}^{3\times 3} : \sigma_2 \left(\mathbf{H}\right)^2 = 1 \right\} \tag{9.4}$$

where $\sigma_2 \left(\mathbf{H}\right)^2$ *is the second biggest singular value of* $\mathbf{H}$*.*

The *proof* of this proposition is a simple extension of the proof found in [MSKS04, Lemma 5.18].

9.1 Decomposition of the *Homography* Matrix

For the decomposition of the *homography* matrix, we use the method described in Section 5.3.3 of [MSKS04]. We note that there are other approaches for the decomposition of this matrix as is shown in [MV07]. In this section we briefly describe this method and analyze the decomposition applied to our approach, in which the coordinates of $\mathbf{\Pi}^{(\mathcal{W})}$ are known.

Let us consider the *eigen decomposition* of $\mathbf{H}^T\mathbf{H}$ as

$$\mathbf{H}^T\mathbf{H} = \mathbf{V}\mathbf{\Sigma}\mathbf{V}^T \tag{9.5}$$

with $\mathbf{V} \in \mathcal{SO}(3)$ that can be decomposed into columns as $\mathbf{V} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \end{bmatrix}$ and, from Proposition 9.1, $\mathbf{\Sigma} = \operatorname{diag}\{\sigma_1^2, 1, \sigma_3^2\}$, where $\sigma_1^2 \geqslant 1 \geqslant \sigma_3^2$. We can define $\mathbf{u}_1$ and $\mathbf{u}_2$ as

$$\mathbf{u}_1 = \frac{\sqrt{1-\sigma_3^2}\mathbf{v}_1 + \sqrt{\sigma_1^2 - 1}\mathbf{v}_3}{\sqrt{\sigma_1^2 - \sigma_3^2}} \tag{9.6}$$

$$\mathbf{u}_2 = \frac{\sqrt{1-\sigma_3^2}\mathbf{v}_1 - \sqrt{\sigma_1^2 - 1}\mathbf{v}_3}{\sqrt{\sigma_1^2 - \sigma_3^2}} \tag{9.7}$$

where $|\mathbf{u}_i| = 1$ for $i = 1, 2$.

$$
\begin{array}{ll}
\textbf{Solution 1:} & \textbf{Solution 3:} \\
\mathbf{R}^{(1)} = \mathbf{W}^{(1)} \left(\mathbf{U}^{(1)}\right)^T & \mathbf{R}^{(3)} = \mathbf{R}^{(1)} \\
\breve{\boldsymbol{\pi}}_1^{(\mathcal{W})} = \mathbf{v}_2 \times \mathbf{u}_1 & \breve{\boldsymbol{\pi}}_3^{(\mathcal{W})} = -\breve{\boldsymbol{\pi}}_1^{(\mathcal{W})} \\
\frac{1}{\zeta^{(\mathcal{W})}}\mathbf{t}^{(1)} = \left(\mathbf{H} - \mathbf{R}^{(1)}\right)\mathbf{R}^{(1)} & \frac{1}{\zeta^{(\mathcal{W})}}\mathbf{t}^{(3)} = -\frac{1}{\zeta^{(\mathcal{W})}}\mathbf{t}^{(1)}
\end{array}
\tag{9.12}
$$

$$
\begin{array}{ll}
\textbf{Solution 2:} & \textbf{Solution 4:} \\
\mathbf{R}^{(2)} = \mathbf{W}^{(2)} \left(\mathbf{U}^{(2)}\right)^T & \mathbf{R}^{(4)} = \mathbf{R}^{(2)} \\
\breve{\boldsymbol{\pi}}_2^{(\mathcal{W})} = \mathbf{v}_2 \times \mathbf{u}_2 & \breve{\boldsymbol{\pi}}_4^{(\mathcal{W})} = -\breve{\boldsymbol{\pi}}_2^{(\mathcal{W})} \\
\frac{1}{\zeta^{(\mathcal{W})}}\mathbf{t}^{(2)} = \left(\mathbf{H} - \mathbf{R}^{(2)}\right)\mathbf{R}^{(2)} & \frac{1}{\zeta^{(\mathcal{W})}}\mathbf{t}^{(4)} = -\frac{1}{\zeta^{(\mathcal{W})}}\mathbf{t}^{(2)}
\end{array}
\tag{9.13}
$$

The vectors $\breve{\boldsymbol{\pi}}_i^{(\mathcal{W})}$ correspond to the estimates of the vector $\boldsymbol{\pi}^{(\mathcal{W})}$

Defining matrices $\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \mathbf{W}^{(1)}, \mathbf{W}^{(2)} \in \mathbb{R}^{3\times 3}$ as

$$\mathbf{U}^{(1)} = \left[\begin{array}{ccc} \mathbf{v}_2 & \mathbf{u}_1 & \mathbf{v}_2 \times \mathbf{u}_1 \end{array}\right] \tag{9.8}$$

$$\mathbf{U}^{(2)} = \left[\begin{array}{ccc} \mathbf{v}_2 & \mathbf{u}_2 & \mathbf{v}_2 \times \mathbf{u}_2 \end{array}\right] \tag{9.9}$$

$$\mathbf{W}^{(1)} = \left[\begin{array}{ccc} \mathbf{H}\mathbf{v}_2 & \mathbf{H}\mathbf{u}_1 & \mathbf{H}\mathbf{v}_2 \times \mathbf{H}\mathbf{u}_1 \end{array}\right] \tag{9.10}$$

$$\mathbf{W}^{(2)} = \left[\begin{array}{ccc} \mathbf{H}\mathbf{v}_2 & \mathbf{H}\mathbf{u}_2 & \mathbf{H}\mathbf{v}_2 \times \mathbf{H}\mathbf{u}_2 \end{array}\right], \tag{9.11}$$

the four possible solutions for the decomposition of $\mathbf{H}$ are presented in Equations (9.12) and (9.13).

Usually, the *homography* matrix is used to estimate the relative pose between two perspective images when a planar pattern is viewed by both cameras. In that case, the decomposition of $\mathbf{H}$ into a rotation matrix and translation vector can yield up to four solutions – Equations (9.12) and (9.13). However, in the problem addressed in this thesis we assume that the 3D coordinates of the points are known in one of the coordinate systems, which means that the plane coordinates are known in that coordinate system.

For this problem, it is possible to define the following theorem.

Theorem 9.1. *Given a homography matrix,* $\mathbf{H} \in \mathcal{H}$*, and a plane of known coordinates* $\boldsymbol{\Pi}^{(\mathcal{W})}\mathbb{R} \doteq \left(-\zeta^{(\mathcal{W})}, \boldsymbol{\pi}^{(\mathcal{W})}\right) \in \mathbb{R}^4$ *where* $\left|\boldsymbol{\pi}^{(\mathcal{W})}\right| = 1$*, the decomposition of* $\mathbf{H}$ *into rotation and translation has a unique solution.*

The *proof* of this theorem is derived in Section D.1.

9.2 Proposed Approaches

We are considering a camera previously calibrated according to the general camera model – Section 2.1. Therefore for each pixel the corresponding 3D line (in the camera coordinate system) is known. A 3D line can be represented in *Plücker* coordinates $\mathbf{l}^{(\mathcal{C})}\mathbb{R} \doteq \left(\mathbf{d}^{(\mathcal{C})}, \mathbf{m}^{(\mathcal{C})}\right)$ – Section 3.2.1.

Using this representation and from [PW01], a point that is incident on a line verifies the following relationship

$$\mathbf{d}^{(\mathcal{C})} \times \mathbf{p}^{(\mathcal{C})} = \widehat{\mathbf{d}}^{(\mathcal{C})}\mathbf{p}^{(\mathcal{C})} = \mathbf{m}^{(\mathcal{C})}. \tag{9.14}$$

We wish to determine the relationship between the points in the world coordinate system and the lines in the camera coordinate system. Thus, without loss of generality considering the plane $\mathbf{\Pi}^{(\mathcal{W})}$ defined by the three world points (Equation (9.2)), we can write

$$\widehat{\mathbf{d}}^{(\mathcal{C})}\mathbf{H}\mathbf{p}^{(\mathcal{W})} = \mathbf{m}^{(\mathcal{C})}. \tag{9.15}$$

The aim of this approach is to estimate the pose based on the *homography* matrix. Thus, we consider the linearization of the unknown matrix $\mathbf{H}$ in Equation (9.15), using the *Kronecker* product, Section 3.1,

$$\left(\mathbf{p}^{(\mathcal{W})\,T} \otimes \widehat{\mathbf{d}}^{(\mathcal{C})}\right) \mathrm{vec}\,(\mathbf{H}) = \mathbf{m}^{(\mathcal{C})}. \tag{9.16}$$

It is well known from linear algebra [HJ91] that

$$\mathrm{rank}\left(\mathbf{p}^{(\mathcal{W})\,T} \otimes \widehat{\mathbf{d}}^{(\mathcal{C})}\right) = \mathrm{rank}\left(\mathbf{p}^{(\mathcal{W})\,T}\right) \mathrm{rank}\left(\widehat{\mathbf{d}}^{(\mathcal{C})}\right). \tag{9.17}$$

As a result, and since

$$\mathrm{rank}\left(\widehat{\mathbf{d}}^{(\mathcal{C})}\right) = 2 \quad \text{and} \quad \mathrm{rank}\left(\mathbf{p}^{(\mathcal{W})\,T}\right) = 1, \tag{9.18}$$

we have

$$\mathrm{rank}\left(\mathbf{p}^{(\mathcal{W})\,T} \otimes \widehat{\mathbf{d}}^{(\mathcal{C})}\right) = 2, \tag{9.19}$$

which means that from the three rows generated by a single 3D point in the world coordinate system and its correspondent line in the camera coordinate system, only two are linearly independent.

The minimal problem in the case of the camera absolute pose corresponds to the determination of the mapping between three world points – repre-

sented in the world coordinate system, and their corresponding 3D lines – represented in the camera coordinate system. Therefore and using the representation of Equation (9.16), we can define the following algebraic relation

$$\underbrace{\begin{bmatrix} \mathbf{p}_1^{(\mathcal{W})\,T} \otimes \widehat{\mathbf{d}}_1^{(\mathcal{C})} \\ \mathbf{p}_2^{(\mathcal{W})\,T} \otimes \widehat{\mathbf{d}}_2^{(\mathcal{C})} \\ \mathbf{p}_3^{(\mathcal{W})\,T} \otimes \widehat{\mathbf{d}}_3^{(\mathcal{C})} \end{bmatrix}}_{\mathbf{M}} \operatorname{vec}(\mathbf{H}) = \underbrace{\begin{bmatrix} \mathbf{m}_1^{(\mathcal{C})} \\ \mathbf{m}_2^{(\mathcal{C})} \\ \mathbf{m}_3^{(\mathcal{C})} \end{bmatrix}}_{\mathbf{w}} \tag{9.20}$$

where $\mathbf{M} \in \mathbb{R}^{9\times 9}$, $\mathbf{w} \in \mathbb{R}^9$. Note that each mapping between points in the world and lines generates only two linearly independent rows. As a result, it is expected that $\operatorname{rank}(\mathbf{M}) \leqslant 6$. Moreover, we can derive the following theorem

Theorem 9.2. *Consider a set of three points defined in the world coordinate system* $\left\{\mathbf{p}_i^{(\mathcal{W})}\right\}$ *and their correspondent lines in the camera coordinate system* $\left\{\left(\mathbf{d}_i^{(\mathcal{C})}, \mathbf{m}_i^{(\mathcal{C})}\right)\right\}$ *for* $i = 1, 2, 3$. *If the three points define a plane that does not pass through the origin, the dimension of the column–space of* $\mathbf{M}$ *at Equation (9.20) will be rank*$(\mathbf{M}) = 6$.

The *proof* of this theorem is presented in Section D.2. Even if the plane contains the origin a simple change of coordinate system ensures that the conditions of the theorem hold.

9.2.1 Minimal Absolute Pose Problem for General Cameras

In this sub-section we describe a direct method for minimal pose problem using general cameras.

Let us consider matrix $\mathbf{N} \in \mathbb{R}^{9\times 10}$ such that the solution vector $\operatorname{vec}(\mathbf{H})$ is represented in homogeneous coordinates $\boldsymbol{\xi} \in \mathbb{R}^{10}$, and such that

$$\underbrace{\begin{bmatrix} \mathbf{M} & -\mathbf{w} \end{bmatrix}}_{\mathbf{N}} \boldsymbol{\xi} = \mathbf{0}, \quad \text{where} \quad \boldsymbol{\xi} = \begin{bmatrix} \operatorname{vec}(\mathbf{H}) \\ 1 \end{bmatrix}. \tag{9.21}$$

It is clear that $\operatorname{rank}(\mathbf{N}) = \operatorname{rank}(\mathbf{M})$. Thus and from Theorem 9.2, $\operatorname{rank}(\mathbf{N}) = 6$ and, since $\operatorname{rank}(\mathbf{N}) + \operatorname{nullity}(\mathbf{N}) = 10$, $\operatorname{nullity}(\mathbf{N}) = 4$. Let us consider the null-space of $\mathbf{N}$ represented by

$$\operatorname{null}(\mathbf{N}) \doteq \{\alpha_1\mathbf{f}_1 + \alpha_2\mathbf{f}_2 + \alpha_3\mathbf{f}_3 + \alpha_4\mathbf{f}_4 : \alpha_1, \alpha_2, \alpha_3, \alpha_4 \in \mathbb{R}\} \tag{9.22}$$

where the $\mathbf{f}_i \in \mathbb{R}^{10}$ are the vectors defining its basis, and $\boldsymbol{\xi} \subset \text{null}(\mathbf{N})$. Note that $\boldsymbol{\xi}$ has four degrees of freedom. However from Equation (9.21), we see that the tenth element of vector $\boldsymbol{\xi}$ must be $\xi_{10} = 1$.

Without loss of generality, we can consider that the tenth elements of the i^{th} bases $\mathbf{f}_i$ are equal to one, for $i = 1, 2, 3, 4$. Using this result, we can define a constraint $\alpha_4 = 1 - \alpha_1 - \alpha_2 - \alpha_3$. Using this constraint and redefining the basis as $\breve{\mathbf{f}}_1 = \mathbf{f}_1 - \mathbf{f}_4$, $\breve{\mathbf{f}}_2 = \mathbf{f}_2 - \mathbf{f}_4$, $\breve{\mathbf{f}}_3 = \mathbf{f}_3 - \mathbf{f}_4$ and $\breve{\mathbf{f}}_4 = \mathbf{f}_4$, we define the affine space $\mathcal{R}$ as:

$$\mathcal{R} \doteq \left\{ \alpha_1 \breve{\mathbf{f}}_1 + \alpha_2 \breve{\mathbf{f}}_2 + \alpha_3 \breve{\mathbf{f}}_3 + \breve{\mathbf{f}}_4 : \alpha_i \in \mathbb{R} \right\}, \tag{9.23}$$

in which $\boldsymbol{\xi} \in \mathcal{R}$ – using this representation we ensure that for any α_i, the tenth element of the vector $\boldsymbol{\xi}$ is always equals to $\xi_{10} = 1$. Note that in this representation, the space has only three degrees of freedom.

For simplicity, we can use numerical methods such as the *eigen decomposition* to compute basis $\mathbf{f}_i$ and then derive $\breve{\mathbf{f}}_i$ as described. However, an analytical solution for the basis $\breve{\mathbf{f}}_i$ is derived in the Appendix B.

Un-staking the vectors $\breve{\mathbf{f}}_i$ to $\breve{\mathbf{F}}_i$ we define $\mathbf{H}$ as a function of the unknowns α_1, α_2 and α_3 as

$$\mathbf{H} = \alpha_1 \breve{\mathbf{F}}_1 + \alpha_2 \breve{\mathbf{F}}_2 + \alpha_3 \breve{\mathbf{F}}_3 + \breve{\mathbf{F}}_4. \tag{9.24}$$

The estimated *homography* matrix must represents a rigid–body transformation for points at $\boldsymbol{\Pi}^{(\mathcal{W})}$, which means that the distance between any two points after the application of the *homography* map must be preserved. Formally,

$$\left\| \mathbf{p}_i^{(\mathcal{W})} - \mathbf{p}_j^{(\mathcal{W})} \right\| = \left\| \mathbf{p}_i^{(\mathcal{C})} - \mathbf{p}_j^{(\mathcal{C})} \right\| \tag{9.25}$$

and, since $\mathbf{p}_i^{(\mathcal{C})} = \mathbf{H}\mathbf{p}_i^{(\mathcal{W})}$,

$$\left\| \mathbf{p}_i^{(\mathcal{W})} - \mathbf{p}_j^{(\mathcal{W})} \right\| = \left\| \mathbf{H}\mathbf{p}_i^{(\mathcal{W})} - \mathbf{H}\mathbf{p}_j^{(\mathcal{W})} \right\|. \tag{9.26}$$

Using this result and considering $\mathbf{q}_{i,j}^{(\mathcal{W})} = \mathbf{p}_i^{(\mathcal{W})} - \mathbf{p}_j^{(\mathcal{W})}$ for the three world points, we can define three constraints that the estimated *homography* matrix must verify

$$\mathbf{q}_{i,j}^{(\mathcal{W})^T} \mathbf{q}_{i,j}^{(\mathcal{W})} = \mathbf{q}_{i,j}^{(\mathcal{W})^T} \mathbf{H}^T \mathbf{H} \mathbf{q}_{i,j}^{(\mathcal{W})} \quad \text{for} \quad (i,j) = \{(1,2), (1,3), (2,3)\}. \tag{9.27}$$

The application of these three constraints on the space of the *homogra-*

$\kappa_1^{(i,j)} = \mathbf{q}_{i,j}^{(\mathcal{W})\,T} \breve{\mathbf{F}}_1^T \breve{\mathbf{F}}_1 \mathbf{q}_{i,j}^{(\mathcal{W})}$	$\kappa_2^{(i,j)} = \mathbf{q}_{i,j}^{(\mathcal{W})\,T} \breve{\mathbf{F}}_2^T \breve{\mathbf{F}}_2 \mathbf{q}_{i,j}^{(\mathcal{W})}$
$\kappa_3^{(i,j)} = \mathbf{q}_{i,j}^{(\mathcal{W})\,T} \breve{\mathbf{F}}_3^T \breve{\mathbf{F}}_3 \mathbf{q}_{i,j}^{(\mathcal{W})}$	$\kappa_4^{(i,j)} = \mathbf{q}_{i,j}^{(\mathcal{W})\,T} \left(\breve{\mathbf{F}}_1^T \breve{\mathbf{F}}_2 + \breve{\mathbf{F}}_2^T \breve{\mathbf{F}}_1 \right) \mathbf{q}_{i,j}^{(\mathcal{W})}$
$\kappa_5^{(i,j)} = \mathbf{q}_{i,j}^{(\mathcal{W})\,T} \left(\breve{\mathbf{F}}_1^T \breve{\mathbf{F}}_3 + \breve{\mathbf{F}}_3^T \breve{\mathbf{F}}_1 \right) \mathbf{q}_{i,j}^{(\mathcal{W})}$	$\kappa_6^{(i,j)} = \mathbf{q}_{i,j}^{(\mathcal{W})\,T} \left(\breve{\mathbf{F}}_2^T \breve{\mathbf{F}}_3 + \breve{\mathbf{F}}_3^T \breve{\mathbf{F}}_2 \right) \mathbf{q}_{i,j}^{(\mathcal{W})}$
$\kappa_7^{(i,j)} = \mathbf{q}_{i,j}^{(\mathcal{W})\,T} \left(\breve{\mathbf{F}}_1^T \breve{\mathbf{F}}_4 + \breve{\mathbf{F}}_4^T \breve{\mathbf{F}}_1 \right) \mathbf{q}_{i,j}^{(\mathcal{W})}$	$\kappa_8^{(i,j)} = \mathbf{q}_{i,j}^{(\mathcal{W})\,T} \left(\breve{\mathbf{F}}_2^T \breve{\mathbf{F}}_4 + \breve{\mathbf{F}}_4^T \breve{\mathbf{F}}_2 \right) \mathbf{q}_{i,j}^{(\mathcal{W})}$
$\kappa_9^{(i,j)} = \mathbf{q}_{i,j}^{(\mathcal{W})\,T} \left(\breve{\mathbf{F}}_3^T \breve{\mathbf{F}}_4 + \breve{\mathbf{F}}_4^T \breve{\mathbf{F}}_3 \right) \mathbf{q}_{i,j}^{(\mathcal{W})}$	$\kappa_{10}^{(i,j)} = \mathbf{q}_{i,j}^{(\mathcal{W})\,T} \left(\breve{\mathbf{F}}_4^T \breve{\mathbf{F}}_4 - \mathbf{I} \right) \mathbf{q}_{i,j}^{(\mathcal{W})}$

Table 9.1: *In this table we show the coefficient parameters of the polynomial equations derived in Equation (9.28)*

phy matrix defined in the Equation (9.24) will generate three constraints $g_{i,j}(\alpha_1, \alpha_2, \alpha_3) = 0$ on the unknowns α_1, α_2 and α_3, where $g_{i,j}(\alpha_1, \alpha_2, \alpha_3)$ is such that

$$g_{i,j}(\alpha_1, \alpha_2, \alpha_3) = \alpha_1^2 \kappa_1^{(i,j)} + \alpha_2^2 \kappa_2^{(i,j)} + \alpha_3^2 \kappa_3^{(i,j)} + \alpha_1 \alpha_2 \kappa_4^{(i,j)} + \alpha_1 \alpha_3 \kappa_5^{(i,j)} + \\ + \alpha_2 \alpha_3 \kappa_6^{(i,j)} + \alpha_1 \kappa_7^{(i,j)} + \alpha_2 \kappa_8^{(i,j)} + \alpha_3 \kappa_9^{(i,j)} + \kappa_{10}^{(i,j)}. \quad (9.28)$$

The coefficients of the polynomial equations, $\kappa_n^{(i,j)}$ for $n = 1, \ldots, 10$, are shown in Table 9.1.

To conclude, the solutions for the *homography* matrix are given by the Equation (9.24), where α_1, α_2 and α_3 verify

$$g_{1,2}(\alpha_1, \alpha_2, \alpha_3) = g_{1,3}(\alpha_1, \alpha_2, \alpha_3) = g_{2,3}(\alpha_1, \alpha_2, \alpha_3) = 0. \quad (9.29)$$

Note that each one of these functions represents a quadric. As a result, the solutions for vector $(\breve{\alpha}_1, \breve{\alpha}_2, \breve{\alpha}_3)$ that verify the constraints of Equations (9.29), can be seen as corresponding to the points where the three quadrics intersects. Moreover, from *Bézout's* theorem [CLO04] and since we have three constraints of degree two, we can conclude that we have up to eight solutions. A general method to estimate the intersection of quadrics is described in Section C.

A necessary condition for the decomposition of the *homography* matrix described in Section 9.1 is that the estimated matrices $\mathbf{H}$ verify $\mathbf{H} \in \mathcal{H}$. This is the same as to prove that $\mathbf{H}$ verifies the conditions of a rigid transformation for any point that belongs to the plane $\mathbf{\Pi}^{(\mathcal{W})}$. Therefore, this requirement must be met to ensure that the estimates $\mathbf{R} \in \mathcal{SO}(3)$ and $\mathbf{t} \in \mathbb{R}^3$.

From the following Proposition, all the solutions for $\mathbf{H}$ that are obtained

from the intersection of the three quadrics (Equations (9.29)) verify the condition $\mathbf{H} \in \mathcal{H}$ and, from Theorem 9.1, will generate a single solution for $\mathbf{R} \in \mathcal{SO}(3)$ and $\mathbf{t} \in \mathbb{R}^3$.

Proposition 9.2. *A matrix* $\mathbf{H} \in \mathbb{R}^{3\times3}$ *that satisfies* ${\mathbf{q}_{i,j}^{(\mathcal{W})}}^T \mathbf{q}_{i,j}^{(\mathcal{W})} = {\mathbf{q}_{i,j}^{(\mathcal{W})}}^T \mathbf{H}^T\mathbf{H}\mathbf{q}_{i,j}^{(\mathcal{W})}$ *for* $(i,j) = \{(1,2),(1,3),(2,3)\}$ *and* $\mathbf{q}_{i,j}^{(\mathcal{W})} = \mathbf{p}_i^{(\mathcal{W})} - \mathbf{p}_j^{(\mathcal{W})}$, *defines a rigid transformation for any points* $\mathbf{p}^{(\mathcal{W})} \in \mathbf{\Pi}^{(\mathcal{W})}$ *where* $\mathbf{\Pi}^{(\mathcal{W})} \doteq \mathbf{p}_1^{(\mathcal{W})} \cup \mathbf{p}_2^{(\mathcal{W})} \cup \mathbf{p}_3^{(\mathcal{W})}$, *and as a result verifies the Proposition 9.1.*

The *proof* of this Proposition is presented in the Appendix D.3.

9.2.2 Minimal Absolute Pose Problem for Central Cameras

Let us now consider the case of a central/perspective camera. Without loss of generality, we consider as the origin of the camera coordinate system the center of projection, i.e., the point where all the rays intersect. Since all the rays pass through the origin, the moments of the lines will be zero and we can write $\mathbf{w} = \mathbf{0}$. In this case, from the algebraic relationship of Equation (9.20), the solutions for $\text{vec}(\mathbf{H})$ will belong to the null-space of $\mathbf{M}$. From Theorem 9.2, we have $\text{nullity}(\mathbf{M}) = 3$, which means that we can define $\text{vec}(\mathbf{H}) \in \text{null}(\mathbf{M})$, such that

$$\text{null}(\mathbf{M}) \doteq \{\alpha_1\mathbf{f}_1 + \alpha_2\mathbf{f}_2 + \alpha_3\mathbf{f}_3 : \alpha_1,\alpha_2,\alpha_3 \in \mathbb{R}\} \tag{9.30}$$

where $\mathbf{f}_i$, for $i = 1,2,3$, are the vectors corresponding to the basis for the null-space. Similar to the general case, these basis can be computed using numerical methods such as *eigen decomposition* or using an analytical solution derived in the Appendix B.

Un-staking the columns $\mathbf{f}_i$ to $\mathbf{F}_i$, we can define $\mathbf{H}$ as a function of unknowns α_1, α_2 and α_3 such that

$$\mathbf{H} = \alpha_1\mathbf{F}_1 + \alpha_2\mathbf{F}_2 + \alpha_3\mathbf{F}_3. \tag{9.31}$$

As we describe in Proposition 9.1, the *homography* matrix can be defined up to a scale factor. If we consider $\breve{\mathbf{H}} = \alpha_3^{-1}\mathbf{H}$, we can write

$$\breve{\mathbf{H}} = \breve{\alpha}_1\mathbf{F}_1 + \breve{\alpha}_2\mathbf{F}_2 + \mathbf{F}_3. \tag{9.32}$$

where $\breve{\alpha}_1 = \alpha_3^{-1}\alpha_1$ and $\breve{\alpha}_2 = \alpha_3^{-1}\alpha_2$.

Like the general central case, the *homography* matrix must represent a rigid transformation for points in the plane $\mathbf{\Pi}^{(\mathcal{W})}$. As a result, the three

$\mu_1^{(i,j)} = \mathbf{q}_{i,j}^{(\mathcal{W})\,T} \mathbf{F}_1^T \mathbf{F}_1 \mathbf{q}_{i,j}^{(\mathcal{W})}$	$\mu_2^{(i,j)} = \mathbf{q}_{i,j}^{(\mathcal{W})\,T} \mathbf{F}_2^T \mathbf{F}_2 \mathbf{q}_{i,j}^{(\mathcal{W})}$
$\mu_3^{(i,j)} = \mathbf{q}_{i,j}^{(\mathcal{W})\,T} \mathbf{q}_{i,j}^{(\mathcal{W})}$	$\mu_4^{(i,j)} = \mathbf{q}_{i,j}^{(\mathcal{W})\,T} \left(\mathbf{F}_1^T \mathbf{F}_2 + \mathbf{F}_2^T \mathbf{F}_1\right) \mathbf{q}_{i,j}^{(\mathcal{W})}$
$\mu_5^{(i,j)} = \mathbf{q}_{i,j}^{(\mathcal{W})\,T} \left(\mathbf{F}_1^T \mathbf{F}_3 + \mathbf{F}_3^T \mathbf{F}_1\right) \mathbf{q}_{i,j}^{(\mathcal{W})}$	$\mu_6^{(i,j)} = \mathbf{q}_{i,j}^{(\mathcal{W})\,T} \left(\mathbf{F}_2^T \mathbf{F}_3 + \mathbf{F}_3^T \mathbf{F}_2\right) \mathbf{q}_{i,j}^{(\mathcal{W})}$
$\mu_7^{(i,j)} = \mathbf{q}_{i,j}^{(\mathcal{W})\,T} \mathbf{F}_3^T \mathbf{F}_3 \mathbf{q}_{i,j}^{(\mathcal{W})}$	

Table 9.2: *In this table we show the coefficient parameters of the quadratic equations of Equation (9.34).*

following constraints can be applied

$$\breve{\alpha}_3^2\, \mathbf{q}_{i,j}^{(\mathcal{W})\,T} \mathbf{q}_{i,j}^{(\mathcal{W})} = \mathbf{q}_{i,j}^{(\mathcal{W})\,T} \breve{\mathbf{H}}^T \breve{\mathbf{H}} \mathbf{q}_{i,j}^{(\mathcal{W})} \quad \text{for} \quad (i,j) = \{(1,2),(1,3),(2,3)\} \tag{9.33}$$

where $\breve{\alpha}_3 = \alpha_3^{-1}$. Applying these constraints on the *homography* matrix space defined in Equation (9.32), we get three quadric surfaces $g_{i,j}\left(\breve{\alpha}_1, \breve{\alpha}_2, \breve{\alpha}_3\right) = 0$, such that

$$\begin{aligned} g_{i,j}\left(\breve{\alpha}_1, \breve{\alpha}_2, \breve{\alpha}_3\right) = {} & \breve{\alpha}_1^2 \mu_1^{(i,j)} + \breve{\alpha}_2^2 \mu_2^{(i,j)} + \\ & + \breve{\alpha}_3^2 \mu_3^{(i,j)} + \breve{\alpha}_1 \breve{\alpha}_2 \mu_4^{(i,j)} + \breve{\alpha}_1 \mu_5^{(i,j)} + \breve{\alpha}_2 \mu_6^{(i,j)} + \mu_7^{(i,j)}. \end{aligned} \tag{9.34}$$

The coefficients of the polynomial equations $\mu_m^{(i,j)}$ are shown in Table (9.2).

The solution for the estimated *homography* matrix is then given by the unknowns $\breve{\alpha}_1$, $\breve{\alpha}_2$ and $\breve{\alpha}_3$ that verifies

$$g_{1,2}\left(\breve{\alpha}_1, \breve{\alpha}_2, \breve{\alpha}_3\right) = g_{1,3}\left(\breve{\alpha}_1, \breve{\alpha}_2, \breve{\alpha}_3\right) = g_{2,3}\left(\breve{\alpha}_1, \breve{\alpha}_2, \breve{\alpha}_3\right) = 0 \tag{9.35}$$

Similar to the general case, each constraint $g_{i,j}\left(\breve{\alpha}_1, \breve{\alpha}_2, \breve{\alpha}_3\right)$ represents a quadric. As a result, we can have up to eight solutions for the unknowns $\left(\breve{\alpha}_1, \breve{\alpha}_2, \breve{\alpha}_3\right)$ that correspond to the points where the three quadrics intersect. In this case and from Equation (9.34), we can see that we have some monomials missing. Using this information, we derived a closed-form solution for the estimates of the unknowns $\breve{\alpha}_1, \breve{\alpha}_2, \breve{\alpha}_3$. The solver is described in Section C.2. Note that in this case Proposition 9.2 is also verified and as a result $\mathbf{H} \in \mathcal{H}$. This means that for each estimate $\mathbf{H} \in \mathcal{H}$ we have a single $\mathbf{R} \in \mathcal{SO}(3)$ and $\mathbf{t} \in \mathbb{R}^3$.

Note that in the case of a minimal pose problem for a central camera, we have up to four solutions, [HLON94]. Let us consider the solutions for $\mathrm{vec}(\mathbf{H})$ as $\mathbf{u}_i$, for $i = 1, \ldots, n$ where $n \leqslant 8$ is the number of possible solutions.

Algorithm 9.1 A Direct Solution for the Minimal General Absolute Pose Problem – Section 9.2.1.

1. Compute the null-space of matrix $\mathbf{N}$, which is obtained from the set of correspondences between world points and lines. We can use numerical methods such as *eigen decomposition* or the analytical solution derived in Appendix B;
2. Using the basis of the estimated null-space, compute the coefficients of the quadric surfaces;
3. Compute the intersection of the quadric surfaces – using the method described in Appendix C.1:
   (a) Compute the matrices $\mathbf{C}_0$, $\mathbf{C}_1$ and $\mathbf{C}_2$ using the coefficients of the polynomial equations;
   (b) Using these matrices compute matrix $\mathbf{C}$, as shown in Equations (C.3);
   (c) Compute the *eigen decomposition* of matrix $\mathbf{C}$;
   (d) For each *eigenvalue*, set α_1 equal to the inverse of the *eigenvalue* and α_2 and α_3 are assigned to the correspondent *eigenvector* elements;
4. For each set of $\{\alpha_1, \alpha_2, \alpha_3\}$, compute the estimates for the *homography* matrix; and decompose each estimate using the method described in Section 9.1.

From Equation (9.20) and since $\mathbf{w} = \mathbf{0}$, it is straightforward to verify that the solutions can be grouped into pairs $\{\mathbf{u}_l, \mathbf{u}_m\}$, where $\mathbf{u}_m = -\mathbf{u}_l$ – if we have a solution $\mathbf{u}_l$ that verifies $\mathbf{M}\mathbf{u}_l = \mathbf{0}$, then we will have $\mathbf{u}_m = -\mathbf{u}_l$ that verifies $\mathbf{M}\mathbf{u}_m = \mathbf{0}$. As a result, if we consider only solutions corresponding to points in front of the camera (which is usually used in these cases [HLON94]), one of the solutions from each pair can be eliminated and, as a result, we will have up to four solutions also.

9.3 Algorithm Outline

Despite the complexity in the derivation of some steps of the method, the result algorithm is very simple, intuitive and it can be easily implemented.

Algorithm 9.2 A Direct Solution for the Minimal Central Absolute Pose Problem – Section 9.2.2.

1. Compute the null-space of the matrix $\mathbf{M}$. We can use the *eigen decomposition* or the analytical solution derived in Appendix B;

2. Using the basis of the estimated null-space, compute the coefficients of the polynomial equations;

3. Compute the points where the three quadric surfaces intersect. Using the method derived in Appendix C.2:

   (a) Using the coefficients of the quadric surfaces, get the polynomial equations $\Phi_1[\widetilde{\alpha}_1, \widetilde{\alpha}_2]$ $\Phi_4[\widetilde{\alpha}_1]$, $\Gamma[\widetilde{\alpha}_1]$ and $\Phi_7[\widetilde{\alpha}_1]$ using Equations (C.4), (C.6) and (C.8);

   (b) Compute the roots of the fourth degree equation $\Phi_7[\widetilde{\alpha}_1]$ and set the $\widetilde{\alpha}_1$ equal to the roots. Compute $\widetilde{\alpha}_2$ and $\widetilde{\alpha}_3$ using Equations (C.4) ans (C.6).

   (c) Recover the set of unknowns $\{\alpha_1, \alpha_2, \alpha_3\}$.

   We note that this method computes the roots in closed-form;

4. For each set of $\{\alpha_1, \alpha_2, \alpha_3\}$, compute the estimates for the *homography* matrix; and decompose each estimate using the method described in Section 9.1.

In terms of formulation, we note that our algorithm requires: the computation of the null-space of $\mathbf{M}$ or $\mathbf{N}$ – we can use numerical methods such as *eigen decomposition* or an analytical solution derived in the Appendix B; solving the problem of finding the 3D points where three quadrics intersect – Appendix C.1 or Appendix C.2, for general or central cameras respectively; and decomposing the estimated *homography* matrices – using the method described in Section 9.1.

The proposed methods are described in Algorithm 9.1 and 9.2 for general and central camera models respectively. In the proposed algorithms we used the minimal solvers proposed in Section C.1 and C.2. However, there exist faster methods that can be used.

Chapter 10

Formalization – Method 2

The approach proposed in this chapter is an extension of the formalization introduced in Chapter 9. Again, the aim is to estimate the rotation $\mathbf{R} \in \mathcal{SO}(3)$ and translation $\mathbf{t} \in \mathbb{R}^3$ parameters that define the rigid transformation between the world and camera coordinate systems.

Similar to the previous case and without loss of generality, let us consider the *homography* to represent the transformation between the world and camera coordinate systems – Equation (9.3). Without loss of generality, we can apply an additional known transformation to the data points and plane coordinates, $\mathbf{R}^\dagger \in \mathcal{SO}(3)$ and $\mathbf{t}^\dagger \in \mathbb{R}^3$. We consider a transformation such that

$$\breve{\mathbf{p}}^{(\mathcal{W})} = \mathbf{R}^\dagger \mathbf{p}^{(\mathcal{W})} + \mathbf{t}^\dagger \tag{10.1}$$

$$\breve{\mathbf{\Pi}}^{(\mathcal{W})} \doteq \left[\breve{\zeta}^{(\mathcal{W})}, \breve{\boldsymbol{\pi}}^{(\mathcal{W})}\right] = \left[\zeta^{(\mathcal{W})} - \mathbf{t}^\dagger \cdot \mathbf{R}^\dagger \boldsymbol{\pi}^{(\mathcal{W})}, \mathbf{R}^\dagger \boldsymbol{\pi}^{(\mathcal{W})}\right], \tag{10.2}$$

which makes $\breve{\boldsymbol{\pi}}^{(\mathcal{W})}$ parallel to the z–axis. The rotation $\mathbf{R}^\dagger \in \mathcal{SO}(3)$ is given by

$$\mathbf{R}^\dagger = \begin{bmatrix} \mathbf{d}_4 & \mathbf{d}_5 & \mathbf{d}_6 \end{bmatrix}^T \text{ where } \mathbf{d}_6 = \boldsymbol{\pi}^{(\mathcal{W})}. \tag{10.3}$$

To compute $\mathbf{d}_4$ we consider two vectors $\mathbf{n}_-$ and $\mathbf{n}_+$, such that

$$\mathbf{n}_- = \mathbf{d}_6 \times (0,1,0) \quad \text{and} \quad \mathbf{n}_+ = \mathbf{d}_6 \times (1,0,0)\,. \tag{10.4}$$

We assign $\mathbf{n}$ as the vector with the largest magnitude from $\mathbf{n}_-$ and $\mathbf{n}_+$. We define $\mathbf{d}_4 = \frac{\mathbf{n}}{|\mathbf{n}|}$ and, to conclude,

$$\mathbf{d}_5 = \mathbf{d}_6 \times \mathbf{d}_4. \tag{10.5}$$

To define a specific depth for the plane, we can use

$$\mathbf{t}^{\dagger} = \mathbf{R}^{\dagger}\left(\frac{1}{3}\sum_{i=1}^{3}\mathbf{p}_i^{(\mathcal{W})}\right) - \begin{bmatrix} 0 & 0 & \breve{\zeta}^{(\mathcal{W})} \end{bmatrix}^T. \tag{10.6}$$

The choice for $\breve{\zeta}^{(\mathcal{W})}$ is not irrelevant. In the next section we describe the constraints for the selection of this parameter.

Using this additional transformation, we can rewrite Equation (9.3) as

$$\mathbf{p}^{(\mathcal{C})} = \underbrace{\left(\mathbf{R} + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \frac{1}{\breve{\zeta}^{(\mathcal{W})}}\mathbf{t} \end{bmatrix}\right)}_{\mathbf{H}} \breve{\mathbf{p}}^{(\mathcal{W})}, \tag{10.7}$$

which means that the aimed *homography* matrix is given by

$$\mathbf{H} = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \left(\mathbf{r}_3 + \frac{1}{\breve{\zeta}^{(\mathcal{W})}}\mathbf{t}\right) \end{bmatrix} \tag{10.8}$$

where $\mathbf{r}_i$ is the i^{th} column of the rotation matrix $\mathbf{R}$. From the fact that $\mathbf{R}$ must belong to the special orthogonal group, the following three constraints can be easily derived

$$\mathbf{r}_1 \cdot \mathbf{r}_1 = 1, \quad \text{which implies} \quad \mathbf{h}_1 \cdot \mathbf{h}_1 = 1 \tag{10.9}$$
$$\mathbf{r}_2 \cdot \mathbf{r}_2 = 1, \quad \text{which implies} \quad \mathbf{h}_2 \cdot \mathbf{h}_2 = 1 \tag{10.10}$$
$$\mathbf{r}_1 \cdot \mathbf{r}_2 = 0, \quad \text{which implies} \quad \mathbf{h}_1 \cdot \mathbf{h}_2 = 0. \tag{10.11}$$

From now on, we will consider that the data-set points in the world are represented in this coordinate system.

10.1 Proposed Approach

Let us consider that the 3D straight lines are represented in *Plücker* coordinates $\mathbf{l}^{(\mathcal{C})}\mathbb{R} \doteq \left(\mathbf{d}^{(\mathcal{C})}, \mathbf{m}^{(\mathcal{C})}\right)$, Section 3.2.1. Similar to the problem fomulation proposed in Chapter 9, we define the incident relation between 3D points and lines – in the world and camera coordinate systems respectively, as

$$\left(\breve{\mathbf{p}}^{(\mathcal{W})\,T} \otimes \widehat{\mathbf{d}}^{(\mathcal{C})}\right) \mathrm{vec}\,(\mathbf{H}) = \mathbf{m}^{(\mathcal{C})}. \tag{10.12}$$

For three world points and their corresponding 3D lines the following algebraic relation can be derived

$$\underbrace{\begin{bmatrix} \breve{\mathbf{p}}_1^{(\mathcal{W})\,T} \otimes \widehat{\mathbf{d}}_1^{(\mathcal{C})} \\ \breve{\mathbf{p}}_2^{(\mathcal{W})\,T} \otimes \widehat{\mathbf{d}}_2^{(\mathcal{C})} \\ \breve{\mathbf{p}}_3^{(\mathcal{W})\,T} \otimes \widehat{\mathbf{d}}_3^{(\mathcal{C})} \end{bmatrix}}_{\mathbf{M}} \text{vec}\,(\mathbf{H}) = \underbrace{\begin{bmatrix} \mathbf{m}_1^{(\mathcal{C})} \\ \mathbf{m}_2^{(\mathcal{C})} \\ \mathbf{m}_3^{(\mathcal{C})} \end{bmatrix}}_{\mathbf{w}} \tag{10.13}$$

where $\mathbf{M} \in \mathbb{R}^{9\times 9}$, $\mathbf{w} \in \mathbb{R}^9$. In Theorem 9.2, we proved the if the three world points $\breve{\mathbf{p}}_i^{(\mathcal{C})}$ do not define a two dimensional subspace, dimension of the column space of the matrix $\mathbf{M}$ will be six. Note that we can choose the transformation parameters $\mathbf{R}^\dagger$ and $\mathbf{t}^\dagger$, such that the plane does not pass through the origin. For that, we have to ensure that the $\breve{\zeta}^{(\mathcal{W})} \neq 0$.

10.1.1 Minimal Absolute Pose for General Camera Models

Let us first consider the general case – where the 3D lines in the camera coordinate system can or can not intersect at a single point. Defining a matrix $\mathbf{N} \in \mathbb{R}^{9\times 10}$ such that

$$\underbrace{\begin{bmatrix} \mathbf{M} & -\mathbf{w} \end{bmatrix}}_{\mathbf{N}} \boldsymbol{\xi} = \mathbf{0}, \quad \text{where} \quad \boldsymbol{\xi} = \begin{bmatrix} \text{vec}\,(\mathbf{H}) \\ 1 \end{bmatrix}. \tag{10.14}$$

Again, from Theorem 9.2, we know that $\text{rank}\,(\mathbf{N}) = 6$. Since

$$\text{rank}\,(\mathbf{N}) + \text{nullity}\,(\mathbf{N}) = 10, \tag{10.15}$$

the dimension of the null space is $\text{nullity}\,(\mathbf{N}) = 4$ and $\boldsymbol{\xi} \subset \text{null}\,(\mathbf{N})$, such that

$$\text{null}\,(\mathbf{N}) \doteq \{\alpha_1\mathbf{e}_1 + \alpha_2\mathbf{e}_2 + \alpha_3\mathbf{e}_3 + \alpha_4\mathbf{e}_4 : \alpha_1, \alpha_2, \alpha_3, \alpha_4 \in \mathbb{R}\}\,. \tag{10.16}$$

$\mathbf{e}_i \in \mathbb{R}^{10}$ are the vectors defining the basis of the null-space. Similar to the case derived in Chapter 9.2.1, we define an affine space $\mathcal{Q}$

$$\mathcal{Q} \doteq \{\alpha_1\breve{\mathbf{e}}_1 + \alpha_2\breve{\mathbf{e}}_2 + \alpha_3\breve{\mathbf{e}}_3 + \breve{\mathbf{e}}_4 : \alpha_1, \alpha_2, \alpha_3 \in \mathbb{R}\} \tag{10.17}$$

such that $\boldsymbol{\xi} \in \mathcal{Q}$. To compute the basis $\breve{\mathbf{e}}_i$, we can use *singular value decomposition* to estimate $\mathbf{e}_i$ and derived $\breve{\mathbf{e}}_i$ as suggested. Or we can use an analytical solution derived in the Section B.

Defining the matrices $\breve{\mathbf{E}}_i \in \mathbb{R}^{3\times 3}$ as the un–stacking matrices of the vectors $\breve{\mathbf{e}}_i$, we can define $\mathbf{H}$ as a function of the unknowns α_1, α_2 and α_3 such that

$$\mathbf{H} = \alpha_1\breve{\mathbf{E}}_1 + \alpha_2\breve{\mathbf{E}}_2 + \alpha_3\breve{\mathbf{E}}_3 + \breve{\mathbf{E}}_4. \tag{10.18}$$

Let us consider the vectors $\mathbf{f}_i^{(j)}$ as the i^{th} column of the matrix $\breve{\mathbf{E}}_j$. Using these vectors and Equation (10.18), we define the two affine spaces for the two first columns of the estimated *homography* matrix as

$$\mathbf{h}_1 = \alpha_1\mathbf{f}_1^{(1)} + \alpha_2\mathbf{f}_1^{(2)} + \alpha_3\mathbf{f}_1^{(3)} + \mathbf{f}_1^{(4)} \tag{10.19}$$

$$\mathbf{h}_2 = \alpha_1\mathbf{f}_2^{(1)} + \alpha_2\mathbf{f}_2^{(2)} + \alpha_3\mathbf{f}_2^{(3)} + \mathbf{f}_2^{(4)}. \tag{10.20}$$

Since the first two columns of the *homography* matrix must correspond to the two first columns of a rotation matrix, they must be orthonormal. As a result, the constraints defined by Equations (10.9-10.11), must be verified. The application of these constraints on the space of solutions for $\mathbf{h}_1$ and $\mathbf{h}_2$ – Equations (10.19) and (10.20), generate three constraint equations, such that

$$g_{1,1}(\alpha_1,\alpha_2,\alpha_3) = g_{2,2}(\alpha_1,\alpha_2,\alpha_3) = g_{1,2}(\alpha_1,\alpha_2,\alpha_3) = 0. \tag{10.21}$$

Each function $g_i(\alpha_1,\alpha_2,\alpha_3)$ represents a second degree polynomial equation with three unknowns

$$\begin{aligned} g_{i,j}(\alpha_1,\alpha_2,\alpha_3) = {} & \alpha_1^2\kappa_1^{(i,j)} + \alpha_2^2\kappa_2^{(i,j)} + \alpha_3^2\kappa_3^{(i,j)} + \alpha_1\alpha_2\kappa_4^{(i,j)} + \\ & + \alpha_1\alpha_3\kappa_5^{(i,j)} + \alpha_2\alpha_3\kappa_6^{(i,j)} + \alpha_1\kappa_7^{(i,j)} + \alpha_2\kappa_8^{(i,j)} + \alpha_3\kappa_9^{(i,j)} + \kappa_{10}^{(i,j)}. \end{aligned} \tag{10.22}$$

The parameters of the coefficients, $\kappa_n^{(i,j)}$ for $n = 1,\ldots,10$ are shown in the Table 10.1.

To conclude, the estimates for the *homography* matrix are given by the Equation (10.18), for the solutions of $(\alpha_1,\alpha_2,\alpha_3)$ that are given by the intersection of the quadrics, Equation (10.21). In the literature, there are a wide variety of methods that can be used to solve this problem. In the Appendix C.1, we propose a very simple and intuitive method based on *polynomial eigenvalues* [KBP12a].

10.1.2 Minimal Absolute Pose for Central Camera

Let us consider the central case and, without loss of generality, the origin of the coordinate system at the point where all the 3D lines intersect, which means that $\mathbf{m}_i^{(\mathcal{C})} = 0$ for $i = 1,2,3$. From the algebraic relation defined in

(i,j)	$(1,2)$	$(1,1)$	$(2,2)$
$\kappa_1^{(i,j)}$	$\mathbf{f}_1^{(1)} \cdot \mathbf{f}_2^{(1)}$	$\mathbf{f}_1^{(1)} \cdot \mathbf{f}_1^{(1)}$	$\mathbf{f}_2^{(1)} \cdot \mathbf{f}_2^{(1)}$
$\kappa_2^{(i,j)}$	$\mathbf{f}_1^{(2)} \cdot \mathbf{f}_2^{(2)}$	$\mathbf{f}_1^{(2)} \cdot \mathbf{f}_1^{(2)}$	$\mathbf{f}_2^{(2)} \cdot \mathbf{f}_2^{(2)}$
$\kappa_3^{(i,j)}$	$\mathbf{f}_1^{(3)} \cdot \mathbf{f}_2^{(3)}$	$\mathbf{f}_1^{(3)} \cdot \mathbf{f}_1^{(3)}$	$\mathbf{f}_2^{(3)} \cdot \mathbf{f}_2^{(3)}$
$\kappa_4^{(i,j)}$	$\mathbf{f}_1^{(1)} \cdot \mathbf{f}_2^{(2)} + \mathbf{f}_1^{(2)} \cdot \mathbf{f}_2^{(1)}$	$2\mathbf{f}_1^{(1)} \cdot \mathbf{f}_1^{(2)}$	$2\mathbf{f}_2^{(1)} \cdot \mathbf{f}_2^{(2)}$
$\kappa_5^{(i,j)}$	$\mathbf{f}_1^{(1)} \cdot \mathbf{f}_2^{(3)} + \mathbf{f}_1^{(3)} \cdot \mathbf{f}_2^{(1)}$	$2\mathbf{f}_1^{(1)} \cdot \mathbf{f}_1^{(3)}$	$2\mathbf{f}_2^{(1)} \cdot \mathbf{f}_2^{(3)}$
$\kappa_6^{(i,j)}$	$\mathbf{f}_1^{(2)} \cdot \mathbf{f}_2^{(3)} + \mathbf{f}_1^{(3)} \cdot \mathbf{f}_2^{(2)}$	$2\mathbf{f}_1^{(2)} \cdot \mathbf{f}_1^{(3)}$	$2\mathbf{f}_2^{(2)} \cdot \mathbf{f}_2^{(3)}$
$\kappa_7^{(i,j)}$	$\mathbf{f}_1^{(1)} \cdot \mathbf{f}_2^{(4)} + \mathbf{f}_1^{(4)} \cdot \mathbf{f}_2^{(1)}$	$2\mathbf{f}_1^{(1)} \cdot \mathbf{f}_1^{(4)}$	$2\mathbf{f}_2^{(1)} \cdot \mathbf{f}_2^{(4)}$
$\kappa_8^{(i,j)}$	$\mathbf{f}_1^{(2)} \cdot \mathbf{f}_2^{(4)} + \mathbf{f}_1^{(4)} \cdot \mathbf{f}_2^{(2)}$	$2\mathbf{f}_1^{(2)} \cdot \mathbf{f}_1^{(4)}$	$2\mathbf{f}_2^{(2)} \cdot \mathbf{f}_2^{(4)}$
$\kappa_9^{(i,j)}$	$\mathbf{f}_1^{(3)} \cdot \mathbf{f}_2^{(4)} + \mathbf{f}_1^{(4)} \cdot \mathbf{f}_2^{(3)}$	$2\mathbf{f}_1^{(3)} \cdot \mathbf{f}_1^{(4)}$	$2\mathbf{f}_2^{(3)} \cdot \mathbf{f}_2^{(4)}$
$\kappa_{10}^{(i,j)}$	$\mathbf{f}_1^{(4)} \cdot \mathbf{f}_2^{(4)}$	$\mathbf{f}_1^{(4)} \cdot \mathbf{f}_1^{(4)} - 1$	$\mathbf{f}_2^{(4)} \cdot \mathbf{f}_2^{(4)} - 1$

Table 10.1: *In this table, we show the coefficient parameters $\kappa_n^{(i,j)}$ of the polynomial equations $g_{i,j}(a,b,c)$ – Equation (10.22).*

Equation (10.13) and similar to the case derived in Section 9.2.2,

$$\text{vec}(\mathbf{H}) \in \text{null}(\mathbf{M}) \tag{10.23}$$

where $\text{null}(\mathbf{M})$ denotes the null-space of the matrix $\mathbf{M}$. From Theorem 9.2, the $\text{nullity}(\mathbf{M}) = 3$ and, as a result,

$$\text{null}(\mathbf{M}) \doteq \{\alpha_1\mathbf{e}_1 + \alpha_2\mathbf{e}_2 + \alpha_3\mathbf{e}_3 : \alpha_1, \alpha_2, \alpha_3 \in \mathbb{R}\}. \tag{10.24}$$

To compute the basis $\mathbf{e}_i$, we can use either the *eigen decomposition* of $\mathbf{M}$ or an analytical solution, derived in Section B.

Unstacking vectors $\mathbf{e}_i$ into matrices $\mathbf{E}_i$, we define a three degree of freedom solution for the *homography* matrix as

$$\mathbf{H} = \alpha_1\mathbf{E}_1 + \alpha_2\mathbf{E}_2 + \alpha_3\mathbf{E}_3. \tag{10.25}$$

Similar to the general case, we define vector spaces for the first two columns of the *homography* matrix, $\mathbf{h}_1$ and $\mathbf{h}_2$, as

$$\mathbf{h}_1 = \alpha_1\mathbf{f}_1^{(1)} + \alpha_2\mathbf{f}_1^{(2)} + \alpha_3\mathbf{f}_1^{(3)} \tag{10.26}$$

$$\mathbf{h}_2 = \alpha_1\mathbf{f}_2^{(1)} + \alpha_2\mathbf{f}_2^{(2)} + \alpha_3\mathbf{f}_2^{(3)}. \tag{10.27}$$

The vector $\mathbf{f}_j^{(i)}$ is the j^{th} column of the matrix $\mathbf{E}_i$. Without loss of generality, we define $\breve{\mathbf{h}}_i = \mathbf{h}_i/\alpha_1$. Using this formulation, $\breve{\mathbf{h}}_1$ and $\breve{\mathbf{h}}_2$ represent two

(i)	(1)	(2)
$\mu_1^{(j)}$	$\mathbf{f}_1^{(2)} \cdot \mathbf{f}_2^{(2)}$	$\mathbf{f}_1^{(2)} \cdot \mathbf{f}_1^{(2)} - \mathbf{f}_2^{(2)} \cdot \mathbf{f}_2^{(2)}$
$\mu_2^{(i)}$	$\mathbf{f}_1^{(3)} \cdot \mathbf{f}_2^{(3)}$	$\mathbf{f}_1^{(3)} \cdot \mathbf{f}_1^{(3)} - \mathbf{f}_2^{(3)} \cdot \mathbf{f}_2^{(3)}$
$\mu_3^{(i)}$	$\mathbf{f}_1^{(2)} \cdot \mathbf{f}_2^{(3)} + \mathbf{f}_1^{(3)} \cdot \mathbf{f}_2^{(2)}$	$2\mathbf{f}_1^{(2)} \cdot \mathbf{f}_1^{(3)} - 2\mathbf{f}_2^{(2)} \cdot \mathbf{f}_2^{(3)}$
$\mu_4^{(i)}$	$\mathbf{f}_1^{(1)} \cdot \mathbf{f}_2^{(2)} + \mathbf{f}_1^{(2)} \cdot \mathbf{f}_2^{(1)}$	$2\mathbf{f}_1^{(1)} \cdot \mathbf{f}_1^{(2)} - 2\mathbf{f}_2^{(1)} \cdot \mathbf{f}_2^{(2)}$
$\mu_5^{(i)}$	$\mathbf{f}_1^{(1)} \cdot \mathbf{f}_2^{(3)} + \mathbf{f}_1^{(3)} \cdot \mathbf{f}_2^{(1)}$	$2\mathbf{f}_1^{(1)} \cdot \mathbf{f}_1^{(3)} - 2\mathbf{f}_2^{(1)} \cdot \mathbf{f}_2^{(3)}$
$\mu_6^{(i)}$	$\mathbf{f}_1^{(1)} \cdot \mathbf{f}_2^{(1)}$	$\mathbf{f}_1^{(1)} \cdot \mathbf{f}_1^{(1)} - \mathbf{f}_2^{(1)} \cdot \mathbf{f}_2^{(1)}$

Table 10.2: *In this table, we show the coefficient parameters $\mu_n^{(i,j)}$ of the polynomial equations $g_i(a, b, c)$ – Equation (10.30).*

dimensional affine spaces such that

$$\breve{\mathbf{h}}_1 = \mathbf{f}_1^{(1)} + \breve{\alpha}_2\mathbf{f}_1^{(2)} + \breve{\alpha}_3\mathbf{f}_1^{(3)} \text{ and } \breve{\mathbf{h}}_2 = \mathbf{f}_2^{(1)} + \breve{\alpha}_2\mathbf{f}_2^{(2)} + \breve{\alpha}_3\mathbf{f}_2^{(3)} \tag{10.28}$$

where $\breve{\alpha}_2 = \alpha_2/\alpha_1$ and $\breve{\alpha}_3 = \alpha_3/\alpha_1$. Since $\breve{\mathbf{h}}_1$ and $\breve{\mathbf{h}}_2$ must represent orthogonal basis and has the same norm, the constraints defined by Equations (10.9-10.11) can be applied to $\breve{\mathbf{h}}_1$ and $\breve{\mathbf{h}}_2$ such that

$$\breve{\mathbf{h}}_1 \cdot \breve{\mathbf{h}}_2 = 0 \text{ and } \breve{\mathbf{h}}_1 \cdot \breve{\mathbf{h}}_1 - \breve{\mathbf{h}}_2 \cdot \breve{\mathbf{h}}_2 = 0. \tag{10.29}$$

Replacing the columns $\breve{\mathbf{h}}_i$ using Equation (10.28), we define two constraints on the unknowns $\breve{\alpha}_2$ and $\breve{\alpha}_3$. These constraints are given by two polynomial equations $g_i(\breve{\alpha}_2, \breve{\alpha}_3) = 0$ of the form

$$g_i(\breve{\alpha}_2, \breve{\alpha}_3) = \breve{\alpha}_2^2\mu_1^{(i)} + \breve{\alpha}_3^2\mu_2^{(i)} + \breve{\alpha}_2\mu_4^{(i)} + \breve{\alpha}_2\breve{\alpha}_3\mu_3^{(i)} + \breve{\alpha}_3\mu_5^{(i)} + \mu_6^{(i)}, \tag{10.30}$$

where the coefficient parameters $\mu_i^{(j)}$ are shown in Table 10.2.

Thus, the solution for the proposed problem – central case, is given by the set of unknowns $\breve{\alpha}_2$ and $\breve{\alpha}_3$ such that

$$g_1(\breve{\alpha}_2, \breve{\alpha}_3) = g_2(\breve{\alpha}_2, \breve{\alpha}_3) = 0. \tag{10.31}$$

The formulation of the Equation (10.31) represents the estimation of the intersection points between two conic lines. From the *Bézout's* theorem [CLO04], the theoretical maximum number of solutions for this problem is four. The solutions for Equation (10.31) can be computed in closed-form, from a simple derivation of the method described in Section C.2.

To conclude the algorithm, we recover the original unknowns α_i using

$$\alpha_1 = \pm \left| \breve{\mathbf{h}}_1 \right|^{-1}, \quad \alpha_2 = \breve{\alpha}_2 \alpha_1 \quad \text{and} \quad \alpha_3 = \breve{\alpha}_3 \alpha_1. \tag{10.32}$$

$\breve{\mathbf{h}}_1$ is computed using Equation (10.28) and the estimated solutions for $\breve{\alpha}_2$ and $\breve{\alpha}_3$.

To conclude, given the set of values for $(\alpha_1, \alpha_2, \alpha_3)$, the estimates for the *homography* matrix are given by the solutions of Equation (10.25).

Note that if we have a solution as $(\alpha_1, \alpha_2, \alpha_3)$, from Equations (10.32) and (10.13) – with $\mathbf{w} = \mathbf{0}$, the solutions $(-\alpha_1, -\alpha_2, -\alpha_3)$ will also verify the same contraints. This means that we can have up to eight solutions. However and from Equation (10.25), this pair of solutions will generate respectively the pair $\mathbf{H}$ and $-\mathbf{H}$ for the solution of *homography* matrix. Moreover and from Equation (10.7), these two estimates for the *homography* matrix will generate the pair $\mathbf{p}^{(\mathcal{C})}$ and $-\mathbf{p}^{(\mathcal{C})}$ for the world point in the camera coordinate system. As a result, if we consider only solutions corresponding to points in front of the camera – which is usually used in these cases [HLON94], one of the solutions from each pair can be eliminated and, as a result, we will have up to four solutions.

10.1.3 Recovery of the Pose

Since pose estimation is specified the transformation parameters between the world and camera coordinate systems, a decomposition of the estimated *homography* matrix into a rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$ is required.

Note that as a result of the pre–defined transformation of the data set, each solution for *homography* matrix will verify Equation (10.8). As a result and since the rotation matrix must belong to the special orthogonal group, the correspondent rotation matrix will be

$$\mathbf{R} = \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2 & (\mathbf{h}_1 \times \mathbf{h}_2) \end{bmatrix} \tag{10.33}$$

where the column vectors $\mathbf{h}_i$ correspond to the i^{th} columns of the estimated *homography* matrix $\mathbf{H}$.

From Equation (10.8) and since we already know the rotation matrix, we can define the translation vector as

$$\mathbf{t} = \breve{\zeta}^{(\mathcal{W})} \left(\mathbf{h}_3 - \mathbf{h}_1 \times \mathbf{h}_2 \right). \tag{10.34}$$

Note that as a result of the application of a pre–defined transformation

Algorithm 10.1 Robust Solution for the Minimal General Absolute Pose Problem – Section 10.1.1.

1 Compute the additional transformation parameters $\mathbf{R}^\dagger$ and $\mathbf{t}^\dagger$ using Equations (10.1). Using Equations (10.1) and (10.2) we compute the coordinates of the points $\breve{\mathbf{p}}_i^{(\mathcal{W})}$ and $\breve{\mathbf{\Pi}}^{(\mathcal{W})}$;

2-4 Compute the Algorithm 9.1, steps 1 to 3;

5 For each set of $\{\alpha_1, \alpha_2, \alpha_3\}$, compute the estimates for the *homography* matrix as described in Equation (10.18); get the parameters $\mathbf{R}$ and $\mathbf{t}$ from Equations (10.33) and (10.34) respectively. To conclude, recover the pose using Equation 10.35.

to the data set, the rotation and translation defined by the decomposition of the *homography* matrix will not represent the pose of the camera. We have to take into account the transformation defined by $\mathbf{R}^\dagger$ and $\mathbf{t}^\dagger$. Consequently, we define the pose parameters by the rotation matrix $\mathbf{R}_{\text{out}}$ and translation vector $\mathbf{t}_{\text{out}}$, such that

$$\mathbf{R}_{\text{out}} = \mathbf{R}\mathbf{R}^\dagger \text{ and } \mathbf{t}_{\text{out}} = \mathbf{R}\mathbf{t}^\dagger + \mathbf{t}. \tag{10.35}$$

10.2 Algorithms

In this section we describe the algorithms for the second proposed formulation for the minimal absolute pose problem using general and central cameras.

The main step of the algorithms consists on the estimation of the points where three quadrics intersect – for the general case; and the points where two conic lines intersect – for the central case. The Algorithm 10.1 shows the method for general camera models and the Algorithm 10.2 summarizes the method for central cameras.

Algorithm 10.2 Robust Solution for the Minimal Central Absolute Pose Problem – Section 10.1.2.

1 Compute the additional transformation parameters $\mathbf{R}^{\dagger}$ and $\mathbf{t}^{\dagger}$ using Equations (10.1). Using Equations (10.1) and (10.2) we compute the coordinates of the points $\breve{\mathbf{p}}_i^{(\mathcal{W})}$ and $\breve{\mathbf{\Pi}}^{(\mathcal{W})}$;

2-3 Compute the first and second steps of the Algorithm 9.2;

4 Compute the points where the two conic lines intersect. Using a simple derivation of the method proposed in Appendix C.2:

(a) Get the polynomial equations $\Phi_4[\breve{\alpha}_3]$, $\Phi_5[\breve{\alpha}_3]$, $\Phi_6[\breve{\alpha}_3]$, $\Gamma[\breve{\alpha}_3]$ and $\Phi_7[\breve{\alpha}_3]$ using Equations (C.6) to (C.8) - using the coefficients of the conic lines;

(b) Compute the roots of the polynomial $\Phi_7[\breve{\alpha}_3]$ and set the $\breve{\alpha}_3$ equal to the roots. Using $\Phi_4[\breve{\alpha}_3]$ and $\Gamma[\breve{\alpha}_3]$, compute $\breve{\alpha}_2$ using Equation (C.6).

(c) Recover the set of unknowns $\{\alpha_1, \alpha_2, \alpha_3\}$, using the Equation (10.32).

We note that this method computes the roots in closed-form;

5 For each set of $\{\alpha_1, \alpha_2, \alpha_3\}$, compute the estimates for the *homography* matrix as described in Equation (10.25); get the parameters $\mathbf{R}$ and $\mathbf{t}$ from Equations (10.33) and (10.34) respectively. To conclude, recover the pose using Equation (10.35).

Chapter 11

Experiments

In this section we evaluate the proposed methods against the state–of–the–art algorithms. For experiments with both central and non-central camera configurations, we consider synthetic data-sets. The following parameters are analyzed: the numerical errors; number of solutions; and computation time. In addition, to measure the robustness of each method, we consider three critical configurations of the non-central cameras, namely: two linear cameras and one orthographic camera. We compute the pose for configurations close to these cases.

For the case of general (non-central) cameras there is no comparison between the previously published methods. As a result, we compare the algorithms proposed in this thesis, against both Nistér & Stewénius [Nis04a, NS07] and Chen & Chang [CC02, CC04a] algorithms in terms of numerical accuracy, number of solutions, computation time and robustness.

With the exception of computational effort, the results for both the methods proposed in this thesis are very similar. As a result, for the numerical accuracy, number of solutions and the tests of robustness, we only display the results corresponding to the second formulation. For the method proposed in this thesis, we consider both the case where matrix $\mathbf{N}$ is computed using the *singular value decomposition* and the case where matrix $\mathbf{N}$ is computed using the analytical approach derived in Appendix B.

For that purpose we consider a cube with 200 units of side length. For the data-set generation, we consider lines defined by a point and a direction. Three points that belong to the line $\mathbf{x}_i^{(\mathcal{C})}$ were defined (inside that cube). Random directions $\mathbf{d}_i^{(\mathcal{C})}$ were computed where, without loss of generality, $\left|\mathbf{d}_i^{(\mathcal{C})}\right| = 1$. Three depths λ_i are randomly generated (with values ranging from 20 to 500) and the coordinates of the 3D points in the camera coordinates $\mathbf{p}_i^{(\mathcal{C})}$ are computed using Equation (8.1). A rigid transformation was randomly

generated ($\mathbf{R} \in \mathcal{SO}(3)$ and $\mathbf{t} \in \mathbb{R}^3$). The rigid transformation generated was applied to the set of points such that $\mathbf{p}_i^{(\mathcal{C})} \mapsto \mathbf{p}_i^{(\mathcal{W})}$. For the data-set $\left\{\mathbf{l}_i^{(\mathcal{C})}, \mathbf{p}_i^{(\mathcal{W})}\right\}$ the pose is estimated using the corresponding algorithms. Note that we can easily get the *Plücker* coordinates from a point and a direction [PW01], as

$$\mathbf{l}_i^{(\mathcal{C})} \doteq \left(\mathbf{d}_i^{(\mathcal{C})}, \mathbf{m}_i^{(\mathcal{C})}\right) = \left(\mathbf{d}_i^{(\mathcal{C})}, \mathbf{x}_i^{(\mathcal{C})} \times \mathbf{d}_i^{(\mathcal{C})}\right). \tag{11.1}$$

This procedure is repeated for 10^6 trials where, for each trial, a new pose and data are randomly generated.

In addition, we also consider the case of central cameras. We considered the state–of–the–art algorithm proposed by Kneip *et al.* [KSS11], which computes the pose in closed-form. The general method proposed by Nistér and Stewénius [Nis04a, NS07] has a variation for central cameras which can also be computed in closed-form. We also considered this method. For the method proposed in this thesis, we consider both the case described in Section 10.1.2 – Algorithm 10.2, and the general case, Section 10.1.1 – Algorithm 10.1. For the general case – Algorithm 10.1, we consider that matrix $\mathbf{N}$ is estimated by using the *singular value decomposition*. For the central case – Algorithm 10.2, we consider that the matrix $\mathbf{M}$ is estimated by using both *singular value decomposition* and using the analytical solution derived in Appendix B. The last case computes pose in closed-form as Kneip *et al.* and Nistér and Stewénius methods.

For the computation of the data-set, we consider a static camera with intrinsic parameters given by matrix $\mathbf{K} \in \mathbb{R}^{3\times3}$ [HZ00]. We randomly generate three image points $\mathbf{u}_i^{(\mathcal{I})}$ (the superscript $(\mathcal{I})$ represents the image space). Using the camera matrix $\mathbf{K}$, we estimate the directions $\mathbf{d}_i^{(\mathcal{C})}$ which correspond to the image points $\mathbf{u}_i^{(\mathcal{I})}$

$$\mathbf{d}_i^{(\mathcal{C})}\mathbb{R} = \mathbf{K}^{-1}\mathbf{u}_i^{(\mathcal{I})}. \tag{11.2}$$

Without loss of generality, we consider $\left|\mathbf{d}_i^{(\mathcal{C})}\right| = 1$. We randomly generate three depths λ_i with values ranging between 20 to 500 units. Using this depths, we compute the world points in the camera coordinate system $\mathbf{p}_i^{(\mathcal{C})} = \lambda_i \mathbf{d}_i^{(\mathcal{C})}$. To conclude, a rigid transformation was randomly generated ($\mathbf{R} \in \mathcal{SO}(3)$ and $\mathbf{t} \in \mathbb{R}^3$) and applied to the 3D points such that $\mathbf{p}_i^{(\mathcal{W})} = \mathbf{R}\mathbf{p}_i^{(\mathcal{C})} + \mathbf{t}$. The generated data-set $\left\{\mathbf{d}_i^{(\mathcal{C})}, \mathbf{p}_i^{(\mathcal{W})}\right\}$, for $i = 1, 2, 3$, is used to estimate the pose, using the correspondent algorithms.

This procedure is repeated for 10^6 trials where, for each trial, a new pose

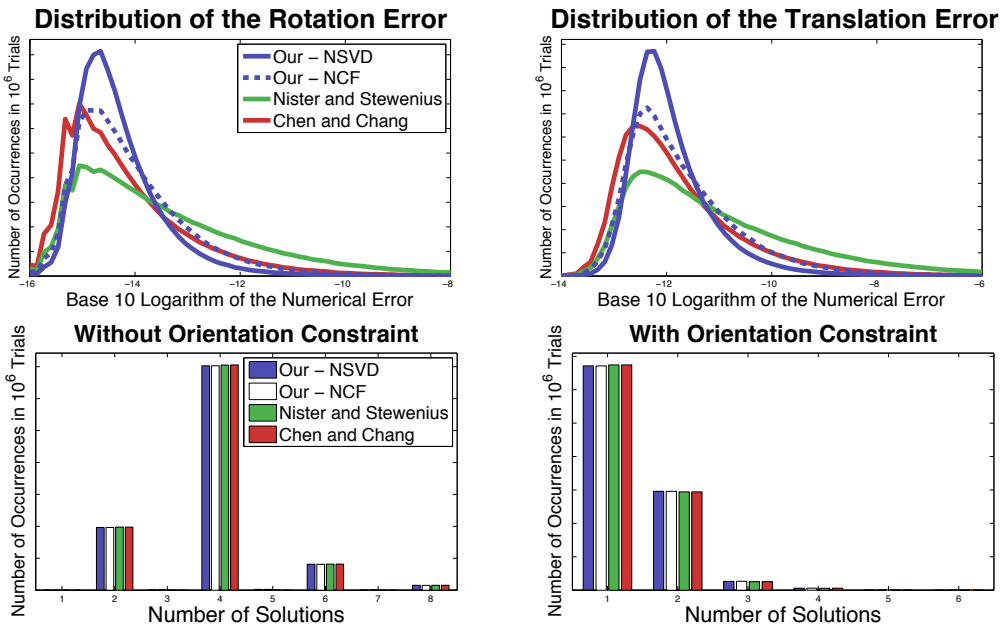


Figure 11.1: *In this figure we show the numerical distribution of the errors and the distribution of the number of solutions for general (non-central) camera models. The proposed method is compared against the state–of–the–art algorithms proposed by Nistér and Stewénius at [Nis04a, NS07] and by Chen and Chang at [CC02, CC04a]. For our method, we consider both the case where the null-space of* $\mathbf{N}$ *is computed using singular value decomposition, denoted as* `Our - NSVD`*; and the case where we use the analytical solution developed in Appendix B, denoted as* `Our - NCF`*.*

and data are randomly generated.

In the following sections we describe: the metrics used in the numerical accuracy tests; the experiments that evaluate the number os solutions; and describe the computation effort required by all the proposed algorithms.

11.1 Numerical Errors

For each of the 10^6 trials, we compute the estimated poses $(\mathbf{R}_{\text{out}}, \mathbf{t}_{\text{out}})$ using the algorithms mentioned. From the set of possible solutions, we eliminate the solutions that do not fit the problem, using the distance from the estimated solutions to the *ground-truth* rotation and translations $(\mathbf{R}_{\text{gt}}, \mathbf{t}_{\text{gt}})$.

For each one of the trials we evaluate the error on the rotation parameters

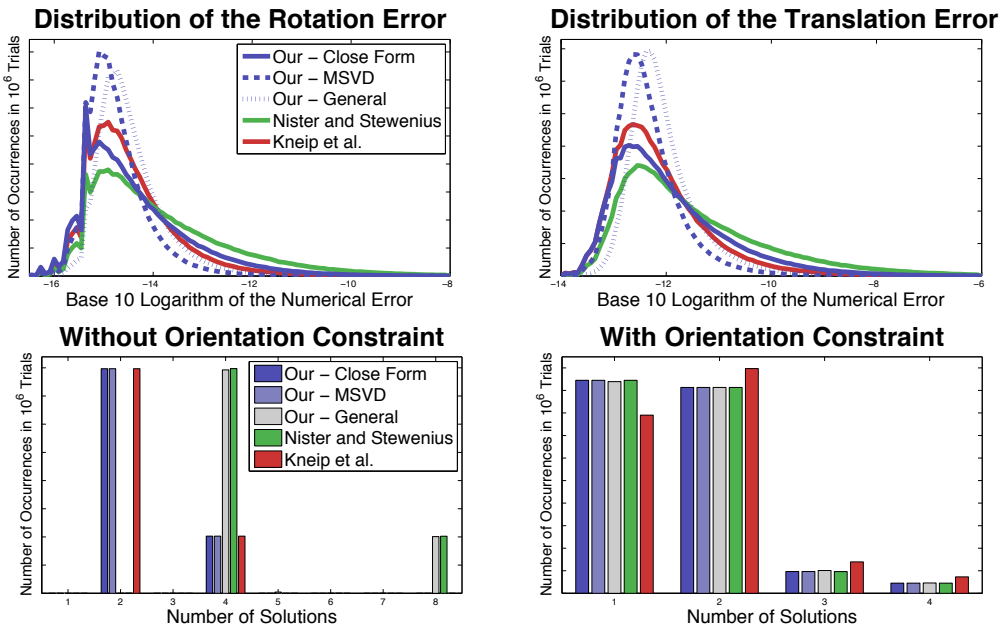


Figure 11.2: *In this figure we show the numerical distribution of the errors and the distribution of the number of solutions for central camera models. The proposed method is compared against the state–of–the–art algorithm proposed by Kneip et al. [KSS11] and the variation Nistér and Stewénius at [Nis04a, NS07] for central cameras (can be solved in close–form). For the algorithm proposed in this article: we consider the case when the null-space of the matrix* $\mathbf{M}$ *is computed using the closed-form solution – Apendix B, denoted as* `Our - Closed-Form`*; and the case when the null-space is computed using the singular value decomposition, denoted as* `Our - MSVD`*. In addition, we consider our general case – used in the experiments of Figure 11.1, denoted as* `Our - General`*.*

by computing ϵ, such that

$$\epsilon = \sqrt{\left(\theta^1_{\text{gt}} - \theta^1_{\text{out}}\right)^2 + \left(\theta^2_{\text{gt}} - \theta^2_{\text{out}}\right)^2 + \left(\theta^3_{\text{gt}} - \theta^3_{\text{out}}\right)^2} \tag{11.3}$$

where $\left(\theta^1_{\text{gt}}, \theta^2_{\text{gt}}, \theta^3_{\text{gt}}\right)$ and $(\theta^1_{\text{out}}, \theta^2_{\text{out}}, \theta^3_{\text{out}})$ are the *ground–truth* and estimated rotation angles, for each of the 10^6 trials. The numerical error for the translation is computed using the norm of the vector $\mathbf{t}_{\text{out}} - \mathbf{t}_{\text{gt}}$.

The distribution of the numerical errors on the value of the rotation and translation are shown in Figures 11.1 and 11.2 for the general (non-central) and for central camera models respectively.

11.2 Number of Solutions

In Figures 11.1 and 11.2 we show the results in terms of number of solutions, for the general (non-central) and central camera models respectively.

To eliminate abnormal solutions, it is usual to consider the constraint that points should be in front of the camera or in the "positive direction" in the case of non-central cameras. In the framework of generalized camera models, we consider this constraint as $\mathbf{d}_i^{(\mathcal{C})^T}\left(\mathbf{p}_i^{(\mathcal{C})} - \mathbf{x}_i^{(\mathcal{C})}\right) > 0$ for $i = 1, 2, 3$, where: $\mathbf{p}_i^{(\mathcal{C})}$ is the i^{th} estimated point in the camera coordinate system; and $\mathbf{x}_i^{(\mathcal{C})}$ is the i^{th} point in the camera coordinate system that defines the forward/positive direction. In the experiments, we denote this constraint as the "Orientation Constraint".

11.3 Computation Time

We have implemented all the algorithms in MATLAB using a *Intel core i7–3930k* with a $3.2GHz$ processor. All the three methods required a set of algebraic transformations. However, in this section we will consider only the steps that require the most significant computational effort, which correspond to iterative procedures such as *eigen decompositions* or *singular value decompositions*.

The analysis of the computation time is divided into two sections: central and non-central cases. All the computation times that are shown in the chapter were computed as the median of the respective computation times, for all the 10^6 trials.

General (Non-Central) Case

As stated in the corresponding papers, most of the computational effort for both Nistér & Stewénius and Chen & Chang methods is spent on finding the roots of the eight degree polynomial equations. To compute the roots, and considering speed and accuracy, the *companion matrix* method was suggested by both Nistér & Stewénius [Nis04a, NS07]; and Chen & Chang [CC02, CC04a] authors. The *companion matrix* corresponds to the computation of a *eigen decomposition* of a 8×8 matrix. Using this method, we were able to compute the roots in $43\mu s$. However, we remark that there are faster algorithms than the *companion matrix* methods, as for example, the *Sturm sequences* [Nis04a, NS07, Nis04b].

In addition and as we mentioned in Chapter 8, for Chen & Chang algorithm the computation of a *singular value decomposition* (in this case of a 3×3 matrix) is necessary, which takes $19\mu s$ for each valid solution – estimation of the rotation and translation parameters from two set of three 3D points.

In the case of the second formulation of the method proposed in this thesis and as was derived by Guo at [Guo12], the intersection of quadrics is computed by determining the roots of a eight degree polynomial equation. Using the *companion matrix* as the previous methods, it takes $43\mu s$ to solve the roots. However, a very simple and easy to implement approach can also be used, based on the representation of the problem using the *polynomial eigenvalue* [KBP12b]. This approach requires the computation of an *eigen decomposition* of a 14×14 matrix – see Appendix C.1, that takes $67\mu s$ – we used this minimal solver in the experiments. In our case, we consider two possibilities for the estimation of the null-space of the matrix $\mathbf{N}$, namely by using the *singular value decomposition* or by using a closed-form analytical solution – Appendix B. The time spent on the computation of the null-space is $40\mu s$, when using the *singular value decomposition*.

For the first method proposed in this thesis, we have to take into account the computation effort spent on the decomposition of the *homography* matrix. If we use the approach proposed in [MSKS04], it requires an *eigen decomposition* of a 3×3 matrix which takes $19\mu s$. We note that there are other methods that can perform this decomposition faster [MV07].

The Case of Central Cameras

For the case of central cameras and considering the second formulation proposed in this thesis, we consider both the computation of the null-space of the matrix $\mathbf{M}$ using *eigen decomposition* and the analytical closed-form solution derived in Appendix B. If we consider the first case, we have to take into account the computation time spent on the estimation of a *eigen decomposition* of a 9×9 matrix, which requires $33\mu s$. On the other hand and since the computation of the conic lines can be performed in closed-form – see Appendix C.2, if we use the analytical solution to estimate the basis of the null-space, all the algorithm can be implemented based on closed-form expressions.

The methods proposed by Kneip *et al.* [KSS11] and the variation for central cameras of the method proposed by Nistér and Stewénius at [Nis04a, NS07] estimate both the rotation and translation parameters directly and in closed-form.

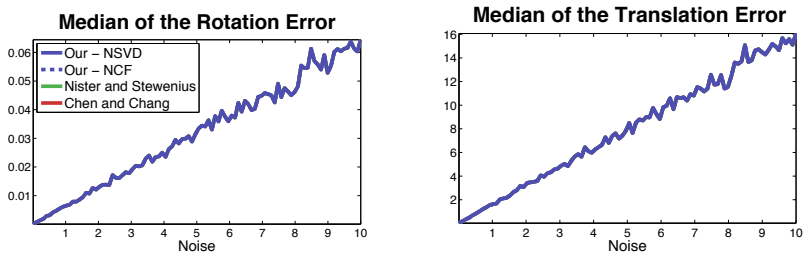


Figure 11.3: *In this figure we show the analysis of the error for data with noise. We consider the* `Noise` *variable as the standard deviation of three random noise vector* $\mathbf{n}_i^{(\mathcal{W})}$ *and compute the pose using the data-set* $\left\{\mathbf{l}_i^{(\mathcal{C})}, \breve{\mathbf{p}}_i^{(\mathcal{W})}\right\}$*, where* $\breve{\mathbf{p}}_i^{(\mathcal{W})} = \mathbf{p}_i^{(\mathcal{W})} + \mathbf{n}_i^{(\mathcal{W})}$*. For each value of the* `Noise`*, we use* 10^3 *trials and the errors are computed as the median of all the errors. As it can be seen from the figures, the four plots overlap, which means that the errors are modeled only by the noise.*

11.4 Experiments with Noise

As was emphasized and discussed by Nistér and Stewénius [NS07], for experiments with noise, solutions to the minimal problems yield similar results which means that these tests are not relevant for comparison. If both the problem formulation and minimal solver are right, the solutions must be the same (in number and value). If we add noise to the data, and since the solutions must be the same, errors in rotation and translation estimates must be the same. If we plot the errors as a function of noise, the plots for the four methods should overlap. However, to see the effects of the noise on the general minimal absolute pose, we add noise to the 3D world points and plot the errors. The results were obtained for the general (non-central) case. The results of the errors for rotation are shown in Figure 11.3.

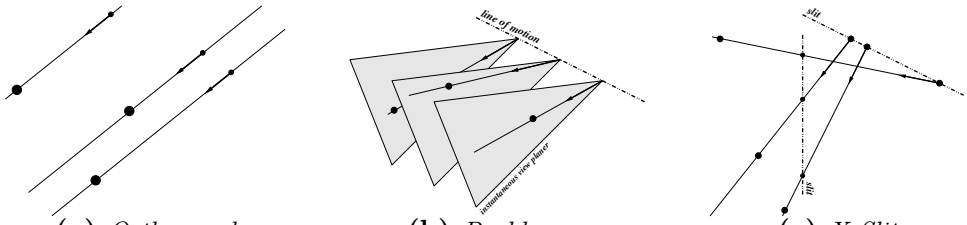
For experiments with real data, the same analysis can be made.

11.5 Critical Configurations

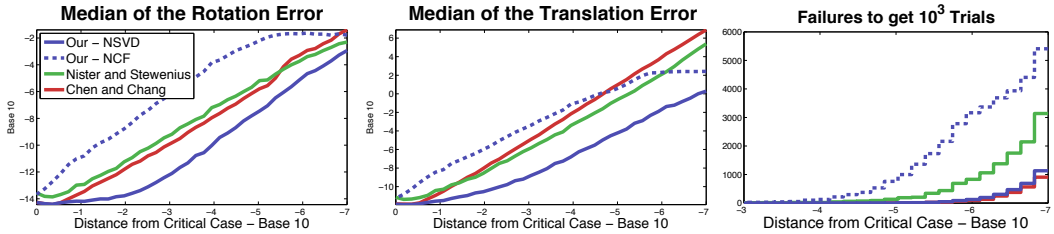
We consider following critical cases: *orthographic* configuration (affine camera) [HZ00] where the pose is degenerate – there is an infinite number of solutions; the classical *linear pushbroom* configuration [GH97, HZ00] (Chen & Chang in [CC04a] state that these camera models can not be handled by their general method) – these types of cameras are used in a wide variety of applications that go from CT to satellite imaging; and the X-Slits cameras [Kin92, ZFPW03] – used in photography to create new images.

In this section we analyze the stability and robustness of the proposed algorithms when dealing with these three types of configurations. The analysis considers the behaviors of the algorithms in a range of configurations that approach the critical cases. For that purpose, we consider the same procedure for data-set generation previously defined. However, instead of considering general directions and points: $\mathbf{d}_i^{(\mathcal{C})}$ and $\mathbf{x}_i^{(\mathcal{C})}$, we constrain the random data to the proposed critical configurations. We compute three vectors $\mathbf{v}_i^{(\mathcal{C})}$ with random directions and norms randomly distributed according to a gaussian distribution with standard deviations defined by the variable called `Distance from Critical Case`. The 3D points in the camera coordinates are then given by $\mathbf{p}_i^{(\mathcal{C})} = \lambda_i \breve{\mathbf{d}}_i^{(\mathcal{C})} + \mathbf{x}_i^{(\mathcal{C})}$, where $\breve{\mathbf{d}}_i^{(\mathcal{C})} = \mathbf{d}_i^{(\mathcal{C})} + \mathbf{v}_i^{(\mathcal{C})}$.

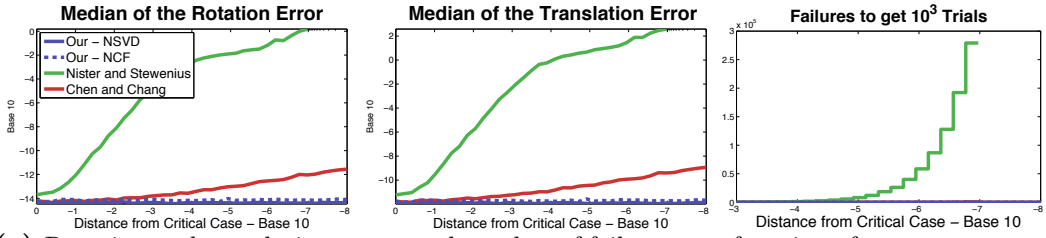
As the variable `Distance from Critical Case` goes from one to zero, the configuration approaches the critical case. From each value of this variable, we compute the median from 10^3 trials. If in some trial the algorithm fails, we ignore that trial and randomly generate a new one. This procedure is repeated until 10^3 trials with estimates are obtained. The results are shown in Figure 11.4. In addition to the errors in the rotation and translation parameters, we show the number of failures generated by each algorithm in the process of getting 10^3 trials with estimates for the rotation and translation.



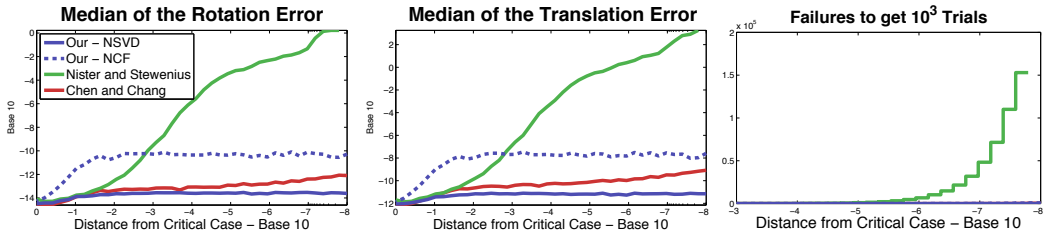
(a) *Orthogonal* **(b)** *Pushbroom* **(c)** *X-Slit*



(d) Rotation and translation errors and number of failures as a function of `Distance from Critical Case` for *Orthogonal* cameras – Figures (a).



(e) Rotation and translation errors and number of failures as a function of `Distance from Critical Case` for *Pushbroom* cameras – Figures (b).



(f) Rotation and translation errors and number of failures as a function of `Distance from Critical Case` for *X-Slit* camera models – Figures (c).

Figure 11.4: *In this figure we evaluate the stability and robustness of the proposed algorithms. We consider the algorithms that we used in previous tests, for general camera models. For that, we consider three critical cases: orthogonal configuration (a) and (d); pushbroom cameras (b) and (e); and X-Slit cameras (c) and (f). To analyze the stability and robustness by computing the errors in rotation and translation parameters that define the pose as a function of a* `Distance from Critical Case` *– see the Section 11.5. Note that the errors are shown in a* 10*-base representation. The errors are computed as a median of* 10^3 *valid trials. Close to critical cases it is possible that some algorithms fail to compute a solution. As a result, and for each algorithm, we show the number of failures that we get until* 10^3 *valid trials are obtained.*

Chapter 12

Discussion

In this chapter we analyse and discuss the experimental results of Chapter 11. We consider both the analysis for the general and the central cases.

12.1 Analysis for the General Case

In terms of the number of solutions – Figure 11.1, before and after the application of the orientation constraint, as expected, the results are similar for the three methods – the algorithm proposed in this thesis and both the Nistér & Stewénius and Chen & Chang methods. As a result, this comparison does not bring additional information concerning the relative merits of the algorithms. Thus, in this section we discuss the experimental results in terms of numerical accuracy, computational effort and robustness to critical configurations.

Numerical Accuracy

In terms of numerical accuracy – Figure 11.1, it can be seen that the three algorithms perform similarly. We note that our method performs slightly better when taking into account the variation of the distribution of the errors – specially for the case where the matrix $\mathbf{N}$ is computed using *singular value decomposition*. The algorithm proposed by Nister & Stewénius performs slightly worse, specially due to a larger variation on the distribution of the numerical errors.

Computational Effort

In terms of computational effort, the algorithm proposed by Nistér & Stewénius and the second method proposed in this thesis gives the rota-

Methods:	*Nistér & Stewénius*	*Our – NCF**	*Our – NSVD**	*Chen & Chang*
Times	$43\mu s$	$43\mu s$	$83\mu s$	$43 + K19\mu s$

Table 12.1: *In this table we summarize the analysis of the main computational effort required in the main computation steps of the proposed algorithms. K – represents the number of valid solutions given by the algorithms. The * symbol means that it corresponds to the computational effort of the minimal solver that uses the companion matrix. We note that for our method, other minimal solvers can be used.*

tion and translation parameters directly. The main computation steps of these two algorithms are the computation of the eight solutions: estimation of the roots of an eighth degree polynomial equation; and finding the points where three quadrics intersect, respectively. In the experiments and because of its simplicity and easiness, for our method, we used an approach based on *polynomial eigenvalues*. This approach is slower when compared with *companion matrix*, which is used by Nistér & Stewénius. However, as shown by Guo at [Guo12], it is possible to derive a solution for the intersection points of the quadrics estimating the roots of an eighth degree polynomial equations. As a result, our method can also be computed using the *companion matrix* as Nistér & Stewénius, which means that it takes the same computation effort.

We note that for the second formulation of the method proposed in this thesis, we give two possibilities for the estimation of the null-space of the matrix $\mathbf{N} \in \mathbb{R}^{9\times 10}$ – Equation (10.14), namely: an analytical solution derived in Appendix B; and using the *singular value decomposition*. The first approach (analytical) does not add computational effort and, as a result, keeps the same computational effort as the Nistér & Stewénius method. However, the second case (*singular value decomposition*) slows the method in $40\mu s$.

The main computation step of the method proposed by Chen & Chang is also the estimation of the roots of an eight degree polynomial equation. However and since the method estimates the 3D points in the camera coordinate system and since the pose is given by the rotation and translation parameters – that define the transformation between the world and camera coordinate systems, it requires the computation of a *singular value decomposition* of each valid solution (up to eight), which takes $19\mu s$.

To conclude the analysis of the computational effort, we summarize the previous paragraphs in the Table 12.1. From the table, we note two aspects: our method using the analytical solution for the null-space of $\mathbf{N}$ as the same computational effort as the fastest method; our method using the *singular value decomposition* is faster than Chen & Chang method for the case where

$K > 2$ which, according to Figure 11.1, happens in most of the cases.

Critical Configurations

Let us consider the general calibration processes. For any specific camera model it is expected that, due to noise, the 3D straight lines generated by the general camera models do not fit the underlying camera geometry. For instance, when we calibrate a perspective camera with noisy data and using general methods such as [GN01, GN05, SR04] or even the method proposed in this thesis –Part II, the probability of the lines intersecting at a single point is very small – even zero. However, depending on the quality of the calibration data and calibration procedure, all the 3D lines must pass close to a common point.

The same ideia can be applied for instance for *X-Slit* cameras – Figure 11.4(c). If we have noise data and use general calibration procedures, the 3D lines generated from the general camera model will not pass through the two slits that define the camera geometry. However and again depending on the quality of the calibration, they must pass close to these slits.

If we consider *Pushbroom* cameras – Figure 11.4(b), the same idea is applied. With noisy data, the 3D lines will not pass through the line of motion nor belong to the instantaneous view plane.

Since the minimal data does not fit the underlying camera geometry, general methods for the pose must be used. Thus, the minimal absolute pose problem for general camera models must work properly for the cases where the configurations are close to these specific camera models as it should work in general case – where lines are randomly generated. Moreover, when we consider classic configurations such as the linear cameras.

As a result of what was described in previous paragraphs, we evaluate the robustness of the method poposed in this thesis against the state–of–the–art methods – Section 11.5, when considering configurations that approximate *X-Slit* and *Pushbroom* camera models. As we can see from the results, our method behaves significantly better than the sate–of–the–art method in almost all the tests. In fact, the results from our method almost do not suffer any changes. The results are more convincing when compared with the Nistér & Stewénius method – we have a difference on the median of the errors up to 10^{13} better; and we had to stop to run the Nistér & Stewénius algorithm earlier because of the high number of failures. When comparing with Chen & Chang method, the results are not as significant as when compared with Nistér & Stewénius algorithm. However, from the experiments we see that our algorithm performs better, with a difference on the median of the errors up to 10^3.

Other configurations using different types of linear cameras were tested and the results were very similar to the configurations studied in Section 11.5.

Note that when the distances between the three 3D points and the imaging device increases, we are becoming close to the "weak perspective" – orthographic camera [HZ00]. Thus, in addition to the previous defined critical configurations, we consider configurations close to the *orthographic* case – a degenerate case. The pose problem should have an infinite number of solutions when considering an *orthographic* configuration and, as a result, the algorithms should fail.

To evaluate the robustness of the methods in this degenerate case, we tested the algorithms with configurations close to the degenerate case. From the experiments and considering the case where matrix $\mathbf{N}$ is computed using the *singular value decomposition*, we see that our algorithm is the most robust. The median of the numerical errors for the proposed algorithm are up to 10^6 better. When our method has a median of the error close to one – for the translation, the other methods have values close to 10^6 for the errors, which is a very significant difference. As a result, we can conclude that our method performs significantly better wen compared with the state–of–the–art algorithms. However, when using the proposed analytical solution to the null-space of $\mathbf{N}$, the results are not as good as when using *singular value decomposition*. Moreover and in most cases, the approach performs slightly worse than the state–of–the–art method. We also note that other solutions for the analytical null-space can be derived, which can improve these results.

To conclude, we see that the method proposed in this thesis is significantly more robust than the state–of–the–art algorithms.

12.2 Analysis for the Central Case

One of the goals of this dissertation was to derive a new solution for the general case. Under this assumption, the analysis of the central case is a special configuration. From the analysis of the numerical errors and number of solutions, Figure 11.2, we see that the general method performs similarly to the Nistér & Stwénius and to the state–of–the–art method for central cameras proposed by Kneip *et al.*

However, in Section 10.1.2, we proposed a derivation of the method proposed in Section 10.1.1, for the specific case of central cameras. From Figure 11.2, we see that this method performs similarly to the state–of–the–art methods in terms of both the numerical errors and the number of solutions. In terms of computation time, we also see that this method can be computed in closed-form, similarly to the state–of–the–art algorithm of Kneip *et al.*

Part IV: Pose Estimation Using Lines

Contents

13 Introduction **90**
13.1 Our Approach 91
13.2 Notations 92
13.2.1 Intersection of Lines 92
13.2.2 Rigid Transformation Applied to Lines 92

14 Formalization **93**
14.1 Analytical Solution for the Pose 93
14.2 Degenerate Cases 96
14.3 Non-Linear Optimization 97
14.4 Algorithm 99

15 Experiments **100**
15.1 Synthetic Data 100
15.2 Convergence of the Non-Linear Method 104
15.3 Experiments with Real Images 105

16 Discussion **107**

Chapter 13

Introduction

As we already described, the problem of pose computation consists on the estimation of six parameters (rotation and translation) that define relative position and orientation between the world and camera coordinate systems. A common approach is to use a set of correspondences between 3D points in the world coordinate system and pixels in the image. Using the imaging model, the image pixels are associated to straight 3D lines in the camera coordinate system – Section 2.1, and pose is computed using the incidence relation between 3D points and correspondent lines. In this part, we study the non-minimal case, where more than three 3D points of known coordinates and their correspondent images are assumed to be known.

Most of the methods described in the literature were derived for the case of central camera models, *eg.* [MNLF07, ACB98, HR11, LHM00, AD03]. However, non-central cameras have also been subject of a significant volume of research specially due to their wide field of view, image resolution, and also because they model new imaging systems. Significant part of the research concerns camera systems where reflections and/or refractions occur. In this paper we address the pose problem for the case of general camera models – Section 2.1. Usually, pose in this kind of systems is estimated using 3D points whose coordinates in world reference system are known and their corresponding pixels, as it is the case of the approaches described in [CC04b, SP08].

The determination of point correspondences (in this case between 3D points and their images) is still a difficult problem and current solutions are error-prone. In this paper we want to avoid this procedure by using coordinates of 3D lines defined in the world coordinate system, instead of 3D points.

When considering non-central cameras, several models exist. For instance, non-central catadioptric cameras using quadric mirrors [SGN06] or

multiple planar mirrors [GN99]. Yet another example is the case of cameras where refraction has to be dealt with [TSKS12]. The goal of this paper is to describe a solution for the pose, within the framework of the generalized camera model. For that purpose, we use the generalized camera model (GCM) as proposed in [GN05] and described in Section 2.1. In this part, we refer to each 3D straight line, associated to each image pixel, as "projection line".

13.1 Our Approach

Let us consider a 3D straight line with known coordinates. Using any imaging device, the image of this line is made up by a set of image pixels. Depending on the complexity of the imaging system, this set of pixels can consist on a non-continuous curve in the image plane. On the other hand, if the imaging system is smooth the association between pixels and "projection lines" is smooth – see Part II, and the curve in the image must be continuous. As a result and since we are considering the general case, we do not take into account any geometric constraints between pixels corresponding to the image of the same 3D straight line (expressed in the world coordinate system). We only assume that we have a set of pixels whose coordinates are known and which correspond to a given 3D straight line.

Using this framework we propose an analytical solution for the pose problem. The solution is based in the following two observations:

- It is well known that we can determine the 3D coordinates of a straight line from the coordinates of four or more incident skewed lines [TM99] – see Figure 14.1. As a result and since we are interested in the general case (where the "projection lines" are unconstrained), from four or more image pixels (corresponding to the same 3D straight line) and their correspondent "projection lines", we can hence estimate the coordinates of the 3D straight line in the camera coordinate system.

- Using the coordinates of the 3D lines estimated in the camera coordinate system and since we know the coordinates of the same lines in the world reference frame, the pose is computed by finding the parameters of the rotation and translation that define the transformation of the lines from the world to the camera coordinate system.

Analytical solutions are useful as a result of their computational speed but, in general, they generate good quality estimates only in the case of noiseless data. However, and in the case of noisy data, those estimates can be improved by means of a non-linear optimization.

13.2 Notations

In this section, we create two notations that will be usefull in the next chapters.

13.2.1 Intersection of Lines

3D straight lines have four degrees of freedom. There are many ways to represent 3D lines. We use the six-tuple *Plücker* coordinates – Section 3.2.1, $\mathcal{G} \doteq (\mathbf{g}, \breve{\mathbf{g}}) \subset \mathbb{R}^6$, where $\mathbf{g} \in \mathbb{R}^3$ and $\breve{\mathbf{g}} \in \mathbb{R}^3$ are respectively the direction and moment of the line $\mathcal{G}$. One of the main advantages of the use of *Plücker* coordinates is the possibility of dealing with incidence relations [PW01]. Let us consider two lines $\mathcal{G} \doteq (\mathbf{g}, \breve{\mathbf{g}})$ and $\mathcal{H} \doteq \left(\mathbf{h}, \breve{\mathbf{h}}\right)$. They intersect if and only if

$$\Omega(\mathcal{G}, \mathcal{H}) = 0 \Leftrightarrow \mathbf{g} \cdot \breve{\mathbf{h}} + \breve{\mathbf{g}} \cdot \mathbf{h} = 0. \tag{13.1}$$

13.2.2 Rigid Transformation Applied to Lines

Let us consider the rigid transformation between the world and camera coordinate systems defined by the rotation $\mathbf{R} \in \mathcal{SO}(3)$ and translation $\mathbf{t} \in \mathbb{R}^3$. Using lines represented in *Plücker* coordinates as $\mathcal{G}^{(\mathcal{W})} \doteq \left(\mathbf{g}^{(\mathcal{W})}, \breve{\mathbf{g}}^{(\mathcal{W})}\right)$ and $\mathcal{G}^{(\mathcal{C})} \doteq \left(\mathbf{g}^{(\mathcal{C})}, \breve{\mathbf{g}}^{(\mathcal{C})}\right)$ – for the same line represented in the world and camera coordinate systems respectively. According to Result 3.1 derived in Section 3.3.1, it is possible to define the linear operator

$$\mathcal{G}^{(\mathcal{C})} \doteq \Psi\left(\mathcal{G}^{(\mathcal{W})}\right), \tag{13.2}$$

such that

$$\begin{bmatrix} \mathbf{g}^{(\mathcal{C})} \\ \breve{\mathbf{g}}^{(\mathcal{C})} \end{bmatrix} \sim \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{E} & \mathbf{R} \end{bmatrix} \begin{bmatrix} \mathbf{g}^{(\mathcal{W})} \\ \breve{\mathbf{g}}^{(\mathcal{W})} \end{bmatrix}, \text{ and } \mathbf{E} \doteq \widehat{\mathbf{t}}\mathbf{R}. \tag{13.3}$$

Note that matrix $\mathbf{E} \doteq \widehat{\mathbf{t}}\mathbf{R}$ is known as the *essential* matrix [HZ00, MSKS04].

Chapter 14

Formalization

Let us consider the problem of pose as the determination of the rotation and translation parameters that define a specific rigid transformation. Using the framework of generalized imaging systems, the image of any 3D straight line is a non-parametric curve in the image plane. Depending on the smoothness of the imaging system, the curve can be continuous or discontinuous. As a result, we consider that we have a set of pixels that belong to the image of a 3D straight line. Since we consider that the camera is calibrated (according to the GCM), we know the coordinates of the associated "projection lines" (for the above mentioned pixels), in the camera coordinate system. In addition, we know that these "projection lines" must intersect the correspondent 3D straight line in the camera coordinate system.

We consider lines represented in *Plücker* coordinates – Section 3.2.1. We denote the i^{th} known 3D straight lines, in the world coordinate system, as $\mathcal{G}_i^{(\mathcal{W})} \doteq \left(\mathbf{g}_i^{(\mathcal{W})}, \breve{\mathbf{g}}_i^{(\mathcal{W})}\right)$. The j^{th} "projection line" associated with the j^{th} pixel of the i^{th} 3D straight line is denoted by $\mathcal{H}_{j,i}^{(\mathcal{C})} \doteq \left(\mathbf{h}_{j,i}^{(\mathcal{C})}, \breve{\mathbf{h}}_{j,i}^{(\mathcal{C})}\right)$, in the camera coordinate system. A scheme of this representation is shown in Figure 14.1.

14.1 Analytical Solution for the Pose

Analytical solutions for the pose are very important because they are computationally fast and also because they allow better insights of the problem. In many cases – as in the case of the "complete" method proposed forward in this chapter, these solutions are also used as the first estimates for an iterative refinement. In this section we derive an analytical solution for the estimation of both the rotation $\mathbf{R}_0 \in \mathcal{SO}(3)$ and translation $\mathbf{t}_0 \in \mathbb{R}^3$ parameters that define the pose for general non-central cameras.

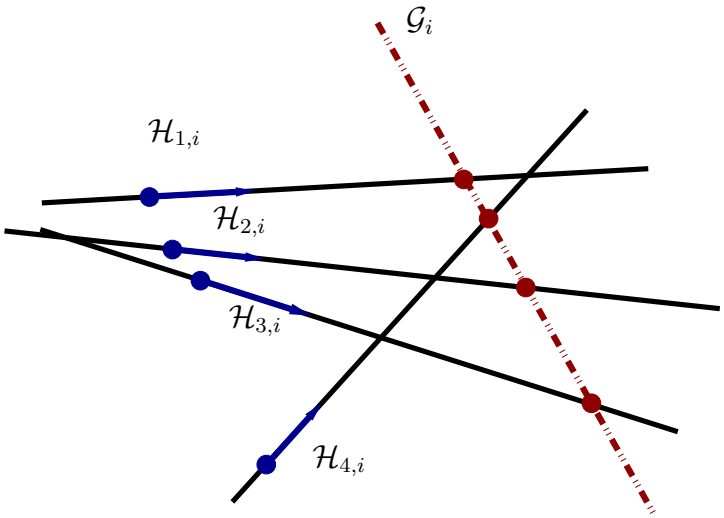


Figure 14.1: *In this figure we show an example of the intersection between a world line $\mathcal{G}_i$ and four "projection lines" $\mathcal{H}_{j,i}$, for $j = 1, \ldots, 4$.*

From the constraint that ensures that two lines intersect – Equation (13.1), and for five or more "projection lines" ($\mathcal{H}_{j,i}^{(\mathcal{C})}$ for $j = 1, \ldots, N_i$, where $N_i \geqslant 5$) that are incident with the 3D straight line $\mathcal{G}_i^{(\mathcal{C})}$ in the camera coordinate system – see Figure 14.1, we define the following algebraic relation

$$\underbrace{\begin{bmatrix} \breve{\mathbf{h}}_{1,i}^{(\mathcal{C})\,T} & \mathbf{h}_{1,i}^{(\mathcal{C})\,T} \\ \vdots & \vdots \\ \breve{\mathbf{h}}_{N_i,i}^{(\mathcal{C})\;T} & \mathbf{h}_{N_i,i}^{(\mathcal{C})\;T} \end{bmatrix}}_{\mathbf{A}_i} \begin{bmatrix} \mathbf{g}_i^{(\mathcal{C})} \\ \breve{\mathbf{g}}_i^{(\mathcal{C})} \end{bmatrix} = \mathbf{0}, \tag{14.1}$$

where $\mathbf{A}_i \in \mathbb{R}^{N_i \times 6}$. Thus, we can conclude that

$$\left(\mathbf{g}_i^{(\mathcal{C})},\ \breve{\mathbf{g}}_i^{(\mathcal{C})}\right) \sim \text{null}\left(\mathbf{A}_i\right). \tag{14.2}$$

For data with noise, matrices $\mathbf{A}_i$ will generally have $\text{rank}\left(\mathbf{A}_i\right) = 6$ which means that there will be no direct solution for Equation (14.2). However, a solution can be computed in the least-squares sense. Note that line $\mathcal{G}_i^{(\mathcal{C})}$ can be represented up to a scale factor. As a result, it is easy to see that the solution for the coordinates of this line can be computed non-iteratively by computing an inverse of a 5×5 matrix.

We also note that for noisy data, the solution for $\mathcal{G}_i^{(\mathcal{C})}$ may not verify the

Klein quadric constraint: $\mathbf{g}_i^{(\mathcal{C})} \cdot \breve{\mathbf{g}}_i^{(\mathcal{C})} = 0$. However, we note that there exist *Plücker* correction algorithms to recover the coordinates meeting the Klein quadric constraint [BS05].

Let us now assume that we have M 3D straight lines in the world coordinate system. For each one of them, we have five or more "projection lines" – $N_i \geqslant 5$, for $i = 1, \ldots, M$. We can compute line coordinates $\mathcal{G}_i^{(\mathcal{C})}$ using Equation (14.2), for all i.

Note that for all the estimated coordinates of the 3D straight lines $\mathcal{G}_i^{(\mathcal{C})}$, we know the correspondent coordinates in the world coordinate system $\mathcal{G}_i^{(\mathcal{W})}$. As a result, the rotation and translation parameters can be estimated such that all the set $\left\{\mathcal{G}_i^{(\mathcal{C})} \leftrightarrow \mathcal{G}_i^{(\mathcal{W})}\right\}$ verifies the transformation defined in Equation (13.2).

Let us consider the matching between directions $\left\{\mathbf{g}_i^{(\mathcal{C})} \leftrightarrow \mathbf{g}_i^{(\mathcal{W})}\right\}$. From Equation (13.3), we can see that the rotation matrix must verify

$$\mathbf{g}_i^{(\mathcal{C})} \sim \mathbf{R}_0 \mathbf{g}_i^{(\mathcal{W})}, \quad \forall i = 1, \ldots, M. \tag{14.3}$$

which means

$$\underbrace{\begin{bmatrix} \mathbf{g}_1^{(\mathcal{C})} & \ldots & \mathbf{g}_M^{(\mathcal{C})} \end{bmatrix}}_{\mathbf{P}_1 \in \mathbb{R}^{3\times M}} \sim \mathbf{R}_0 \underbrace{\begin{bmatrix} \mathbf{g}_1^{(\mathcal{W})} & \ldots & \mathbf{g}_M^{(\mathcal{W})} \end{bmatrix}}_{\mathbf{P}_2 \in \mathbb{R}^{3\times M}} \tag{14.4}$$

The rotation matrix $\mathbf{R}_0$ can be estimated solving the *procrustes* problem [Sch66], which can be solved by computing a *Singular Value Decomposition* of a 3×3 matrix. We note that, since we are dealing with 3×3 matrices, there exists analytical solution for the SVD. Moreover, we also ensure that the solution for the rotation matrix $\mathbf{R}_0$ belongs to the space of orthonormal matrices.

From the second row of Equation (13.3) and since $\breve{\mathbf{g}}_i^{(\mathcal{C})} \times \breve{\mathbf{g}}_i^{(\mathcal{C})} = \mathbf{0}_3$, we can define the following algebraic relation:

$$\breve{\mathbf{g}}_i^{(\mathcal{C})} \times \breve{\mathbf{g}}_i^{(\mathcal{C})} = \left(-\widehat{\mathbf{t}}_0 \mathbf{R}_0 \mathbf{g}_i^{(\mathcal{W})} + \mathbf{R}_0 \breve{\mathbf{g}}_i^{(\mathcal{W})}\right) \times \breve{\mathbf{g}}_i^{(\mathcal{C})} = \mathbf{0}_3. \tag{14.5}$$

Since $\mathbf{R}_0$ was already computed and using any symbolic toolbox, we derive the following algebraic relation

$$\mathbf{B}_i \mathbf{t}_0 = \mathbf{b}_i \tag{14.6}$$

where $\mathbf{B}_i \in \mathbb{R}^{3\times3}$ and $\mathbf{b}_i \in \mathbb{R}^3$. For $i = 1, \ldots, M$, we have

$$\underbrace{\begin{bmatrix} \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_M \end{bmatrix}}_{\mathbf{B}} \mathbf{t}_0 = \underbrace{\begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_M \end{bmatrix}}_{\mathbf{b}} \tag{14.7}$$

where $\mathbf{B} \in \mathbb{R}^{3M\times3}$ and $\mathbf{b} \in \mathbb{R}^{3M}$. The solution for $\mathbf{t}_0$ can be computed using $\mathbf{t}_0 = \mathbf{B}^\dagger\mathbf{b}$, where $\mathbf{B}^\dagger$ represents the *pseudo-inverse* of $\mathbf{B}$. Note that also $\mathbf{B}^\dagger$ can be computed in closed-form, which means that $\mathbf{t}_0$ can be solved analytically too.

14.2 Degenerate Cases

In this section we analyze the degenerate cases that arise from the analytical approach described in the previous section. Two important issues must be taken into account:

1. Cases where the coordinates of the 3D straight lines in the camera coordinate system can not be estimated from the "projection lines" – that is related to Equation (14.2);

2. The *procrustes* problem can not be solved, Equation (14.4).

Let us consider the first case. This degenerate case happens when matrices $\mathbf{A}_i$ – from Equation (14.1), have rank smaller than five. We note that the rank of these matrices depend on the coordinates of the "projection lines". As a result, it depends on both the coordinates of the curves in the image and on the underlying geometry of the imaging device. To have $\text{rank}\,(\mathbf{A}_i) = 5$, it is straightforward to see that at least five "projection lines" are required. We note, however, that five points is not the minimal case. Lines in space have four degrees of freedom and there exist other approaches to estimate the 3D straight line from four skewed lines, *eg.* [TM99]. For simplicity, we use the formulation proposed in Equation (14.1).

For general non-central cameras, the configuration is not degenerate. On the other hand, for central cameras the "projection lines" that are used to define $\mathcal{G}_i^{(\mathcal{W})} \leftrightarrow \left\{\mathcal{H}_{1,i}^{(\mathcal{C})}, \ldots, \mathcal{H}_{N_i,i}^{(\mathcal{C})}\right\}$ will define a plane in the world. As a result, it is not possible to estimate $\mathcal{G}_i^{(\mathcal{C})}$. We also note that even in non-central cases – where in general lines can be reconstructed, there are some 3D straight lines that, because of their configuration, can not be recovered.

Consider, for instance, the catadioptric systems formed with a perspective camera and an axial symmetric mirror. In this case and if the line is incident on the axis of symmetry, the "projection lines" will define a plane, which means that the 3D straight line can not be recovered. Other degenerate configurations associated to the coordinates of the "projection lines" were identified in [TM99].

Let us now consider the second type of degenerate cases. From [Sch66], the *procrustes* problem has a solution if and only if both $\mathbf{P}_1$ and $\mathbf{P}_2$ – from Equation (14.4), have rank equal to three. The columns of these matrices depend on the coordinates of the directions of the known 3D lines in the world, which means that they must form a three dimensional column space. From this analysis, we can conclude that the 3D straight lines used can not be parallel – in this case $\text{rank}\,(\mathbf{P}_i) = 1$, for $i = 1, 2$; also they can not belong to a single 3D plane, in which case $\text{rank}\,(\mathbf{P}_i) = 2$, for $i = 1, 2$.

14.3 Non-Linear Optimization

For each and all of the M 3D straight line – i^{th} line, and the associated j^{th} "projection line", the following constraint must be verified

$$\Omega\left(\mathcal{G}_i^{(\mathcal{C})}, \mathcal{H}_{j,i}^{(\mathcal{C})}\right) = 0, \quad \forall j. \tag{14.8}$$

Note that for this constraint, the lines must be represented in the same coordinate system. We consider the camera coordinate system which means that the coordinates of $\mathcal{G}_i^{(\mathcal{C})}$ are unknowns. However, we know the coordinates of the 3D straight lines in the world coordinate system $\mathcal{G}_i^{(\mathcal{W})}$. Applying the rigid transformation as suggested in Section 13.2.2 – note that this transformation defines the pose; the constraint defined in Equation (14.8) can be rewritten as

$$\Omega\left(\Psi\left(\mathcal{G}_i^{(\mathcal{W})}\right), \mathcal{H}_{j,i}^{(\mathcal{C})}\right) = 0, \quad \forall j. \tag{14.9}$$

The unknowns of these constraints are only the parameters that define the pose. Thus, developing the Equation (14.9) using Equations (13.1) and (13.3) we derive

$$\mathbf{c}_{i,j} \cdot \mathbf{v} = 0. \tag{14.10}$$

The vector $\mathbf{c}_{i,j} \in \mathbb{R}^{18}$ is known and $\mathbf{v} \in \mathbb{R}^{18}$ is such that

$$\mathbf{v} = (r_1, r_2, \ldots, r_9, e_1, e_2, \ldots, e_9)\,, \tag{14.11}$$

where r_i and e_i are the nine individual elements of the matrices $\mathbf{R}$ and $\mathbf{E}$ respectively.

From Equation (14.10) and (14.11), we see that we have eighteen unknowns. However, these unknowns are not linearly independent. From the properties of the rotation matrix $\mathbf{R} \in \mathcal{SO}(3)$, we have

$$\mathbf{R}^T\mathbf{R} = \mathbf{I} \Rightarrow \mathbf{R}^T\mathbf{R} - \mathbf{I} = \mathbf{0}, \tag{14.12}$$

which correspond to nine constraints. We note that these constraints are not all linearly independent too, [AD03]. In the general case, there are only six which are linearly independent.

Moreover, additional constraints must be taken into account from the *essential* matrix $\mathbf{E}$. We know that $\mathbf{E} \doteq \widehat{\mathbf{t}}\mathbf{R}$, which means that

$$\mathbf{E}\mathbf{R}^T = \widehat{\mathbf{t}} \underbrace{\mathbf{R}\mathbf{R}^T}_{\mathbf{I}} = \widehat{\mathbf{t}}. \tag{14.13}$$

Matrix $\widehat{\mathbf{t}}$ must be as Equation (3.1). Thus, we can define six constraints to the product $\mathbf{E}\mathbf{R}^T$.

Adding the six constraints derived from Equation (14.13) to the previous six linearly independent constraints derived from Equation (14.12), we get twelve constraints. Thus, the problem defined by Equation (14.10) subject to the constraints derived in Equations (14.12) and (14.13) has six degrees of freedom – that corresponds to the six degrees of freedom from the pose problem.

The constraints derived by Equations (14.12) and (14.13) are quadratic constraints and can be represented as

$$\mathbf{v}^T\mathbf{K}_l\mathbf{v} = k_l, \;\; \text{for} \;\; l = 1, \ldots, 15, \tag{14.14}$$

where k_l is zero or one, depending on the constraint equation.

To conclude, stacking all the $\mathbf{c}_{i,j}$ (for all 3D straight lines and their associated "projection lines") into matrix $\mathbf{C}$, the general pose using lines can be obtained by solving the problem

$$\begin{aligned} \min_{\mathbf{v}} \quad & ||\mathbf{C}\mathbf{v}||^2 \\ \text{s.t} \quad & \mathbf{v}^T\mathbf{K}_l\mathbf{v} = k_l, \; l = 1, \ldots, 15. \end{aligned} \tag{14.15}$$

Algorithm 14.1 *Solution for the Pose Using Lines for General Non-Central Cameras.*

Let us consider that we have a set $i = 1, \ldots, M$ where $M \geqslant 3$ of lines in the world coordinates $\mathcal{G}_i^{(\mathcal{W})}$ and that each of these lines is seen by $N_i \geqslant 5$ image points, where $j = 1, \ldots, N_i$. Let us assume that the camera is calibrated according to the general camera model – Section 2.1.

1. Get the "projection lines" $\mathcal{H}_{j,i}^{(\mathcal{W})}$ from image pixels that define the lines, using the General Camera Model;
2. Using the lines $\mathcal{H}_{j,i}^{(\mathcal{W})}$, compute the i^{th} world line in the camera coordinates $\mathcal{G}_i^{(\mathcal{G})}$, using the algebraic relation defined in Equation (14.1) and (14.2). Repeate this process for all i;
3. Get $\mathbf{R}_0$ solgin the *procrustes* Problem [Sch66] defined in Equation (14.4); using $\mathbf{R}_0$, estimate $\mathbf{t}_0$ using Equation (14.7);
4. Using the initial estimates $\mathbf{R}_0$ and $\mathbf{t}_0$, optimize pose using the problem defined by Equation (14.17).

Using *Lagrange multipliers* λ_l, we formulate the *Lagrangian* $\mathcal{L}(\mathbf{v}, \boldsymbol{\lambda})$ of this problem as

$$\mathcal{L}(\mathbf{v}, \boldsymbol{\lambda}) = \mathbf{v}^T\mathbf{C}^T\mathbf{C}\mathbf{v} + \sum_{l=1}^{15} \lambda_l \left(\mathbf{v}^T\mathbf{K}_l\mathbf{v} - k_l\right) \tag{14.16}$$

and the problem defined in Equation (14.15) can be rewritten as

$$\max_{\boldsymbol{\lambda}} \min_{\mathbf{v}} \mathcal{L}(\mathbf{v}, \boldsymbol{\lambda}) \tag{14.17}$$

This problem is well studied in the literature. To solve it, we use the `Matlab` optimization toolbox.

14.4 Algorithm

In this part we proposed a method to estimate the pose using lines, for general non-central cameras. The Algorithm 14.1 summarizes, in four steps, the analytical version derived in Section 14.1.

Chapter 15

Experiments

To evaluate and validate the methods and algorithms described, an experimental analysis with both synthetic and real data was performed.

15.1 Synthetic Data

For synthetic data, we consider the following procedure. We randomly generate M 3D straight lines $\mathcal{G}_i^{(\mathcal{C})}$. To get these lines we randomly generate 3D points $\widetilde{\mathbf{g}}_i^{(\mathcal{C})} \in \mathbb{R}^3$ – in a cube with 200 units of side length; and random directions $\mathbf{g}_i^{(\mathcal{C})}$ – with norm equal to one. In this representation, any point incident with the line can be expressed as $\widetilde{\mathbf{g}}_i^{(\mathcal{C})} + \mu_i \mathbf{g}_i^{(\mathcal{C})}$ for some $\mu_i \in \mathbb{R}$. Using the 3D point and respective direction coordinates, we get the *Plücker* coordinates such that

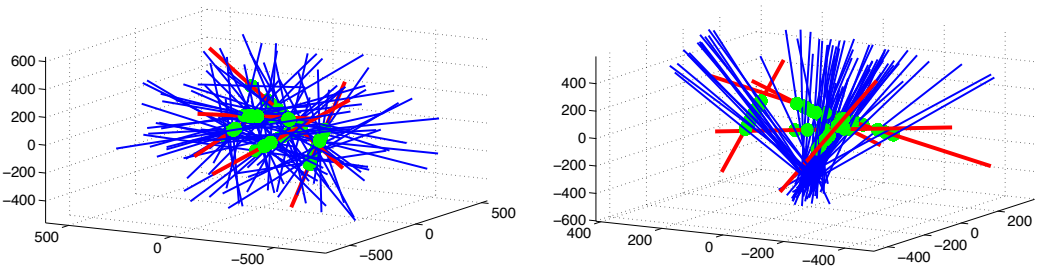
$$\mathcal{G}_i^{(\mathcal{C})} \doteq \left(\mathbf{g}_i^{(\mathcal{C})}, \widetilde{\mathbf{g}}_i^{(\mathcal{C})} \times \mathbf{g}_i^{(\mathcal{C})} \right). \tag{15.1}$$

For more information see [PW01].

For each 3D straight line, we compute N_i "projection lines" $\mathcal{H}_{j,i}^{(\mathcal{C})}$. We randomly choose N_i parameters $\widetilde{\mu}_j$ – for $j = 1, \ldots, N_i$, and compute the coordinates of the 3D points $\widetilde{\mathbf{h}}_{j,i}^{(\mathcal{C})} = \widetilde{\mathbf{g}}_i^{(\mathcal{C})} + \widetilde{\mu}_j \mathbf{g}_i^{(\mathcal{C})}$. Note that $\widetilde{\mathbf{h}}_{j,i}^{(\mathcal{C})}$ belongs to the line $\mathcal{G}_i^{(\mathcal{C})}$. The set of parameters $\widetilde{\mu}_j$ is randomly chosen between -100 and 100. The set of directions $\mathbf{h}_{j,i}^{(\mathcal{C})}$ is randomly computed too. The *Plücker* coordinates of the "projection lines" are then computed as

$$\mathcal{H}_{j,i}^{(\mathcal{C})} \doteq \left(\mathbf{h}_{j,i}^{(\mathcal{C})}, \widetilde{\mathbf{h}}_{j,i}^{(\mathcal{C})} \times \mathbf{h}_{j,i}^{(\mathcal{C})} \right). \tag{15.2}$$

An example of the generation of both $\mathcal{G}_i^{(\mathcal{C})}$ and $\mathcal{H}_{j,i}^{(\mathcal{C})}$ is shown in Fig-



(a) General Case **(b)** Close to Central

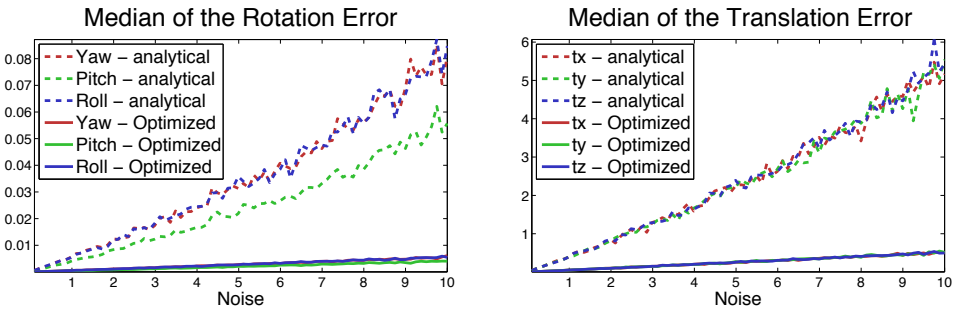
Figure 15.1: *In this figure we show two examples of the generation of the 3D lines and "projection lines" in the camera coordinate system* $\mathcal{H}_{j,i}^{(\mathcal{C})}$ *– blue lines; and* $\mathcal{G}_i^{(\mathcal{C})}$ *– red lines. In green we show the intersection points. In Figure (a), we show an example of the general case. In the case of Figure (b) the directions of the "projection lines" ensure that they pass close to each other – close to central.*

ure 15.1(a).

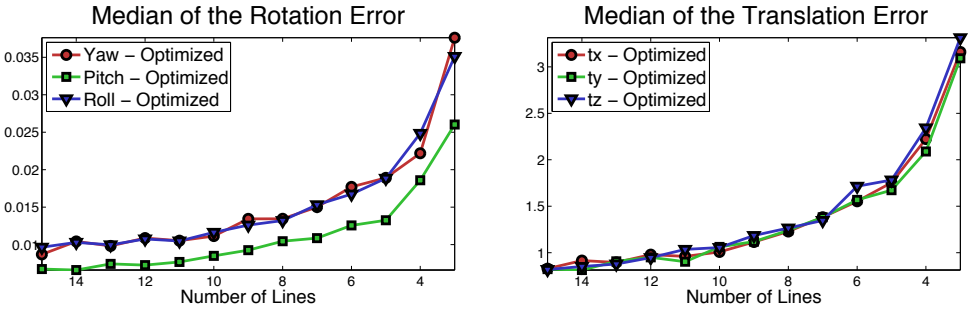
Random ground truth rotations and translation parameters are computed – $\mathbf{R}_{\text{gt}} \in \mathcal{SO}(3)$ and $\mathbf{t}_{\text{gt}} \in \mathbb{R}^3$. We generate $\mathbf{t}_{\text{gt}}$ in a cube with 200 units of side length. Using these parameters, we get the *Plücker* coordinates of the 3D lines in the world coordinate system as $\mathcal{G}_i^{(\mathcal{W})} = \Psi^{-1}\left(\mathcal{G}_i^{(\mathcal{C})}\right)$ – Equation (13.3).

To conclude, pose is computed using the association between $\mathcal{G}_i^{(\mathcal{W})} \leftrightarrow \left\{\mathcal{H}_{1,i}^{(\mathcal{C})}, \ldots, \mathcal{H}_{N_i,i}^{(\mathcal{C})}\right\}$, for all i. For each estimate $\{\mathbf{R}, \mathbf{t}\}$ and ground truth data $\{\mathbf{R}_{gt}, \mathbf{t}_{gt}\}$, we compute the six parameters that define the pose: three angles for the rotation – radians; and three coordinates for the translation. For all the pose parameters, we compute the distance between the ground truth and the estimated parameters.

In many non-central cameras (such as non-central catadioptric cameras with quadric mirrors) and despite the fact that they are non-central, the"projection lines" pass close to each other. Note also that for the analytical solution derived in Section 14.2, the central case is a degenerate case. Therefore it is important that the proposed method be evaluated for those configurations. Instead of considering random directions for $\mathbf{h}_{j,i}^{(\mathcal{C})}$, we constrain those directions in order to to ensure that the "projection lines" pass close to each other. The following procedure is applied: since we already have the 3D coordinates of a point $\widetilde{\mathbf{h}}_{j,i}^{(\mathcal{C})}$ that is incident on the line $\mathcal{H}_{j,i}^{(\mathcal{C})}$, an additional point is enough to compute the coordinates of the direction; since in addition we want that all lines pass close to each other, for each "projection line" we compute a new point $\overline{\mathbf{h}}_{j,i}^{(\mathcal{C})} \in \mathbb{R}^3$ randomly chosen in a



(a) Evaluation of the proposed approach as a function of the `Noise` variable. We consider the general case – Figure 15.1(a). For that purpose, we consider $N_i = 40,\ \forall i$ and $M = 10$.



(b) Evaluation of the proposed approach as a function of the number of 3D lines used. We use $N_i = 40,\ \forall i$ and `Noise` variable with value equals to 20. Note that three 3D lines is the minimal case for the proposed analytical approach.

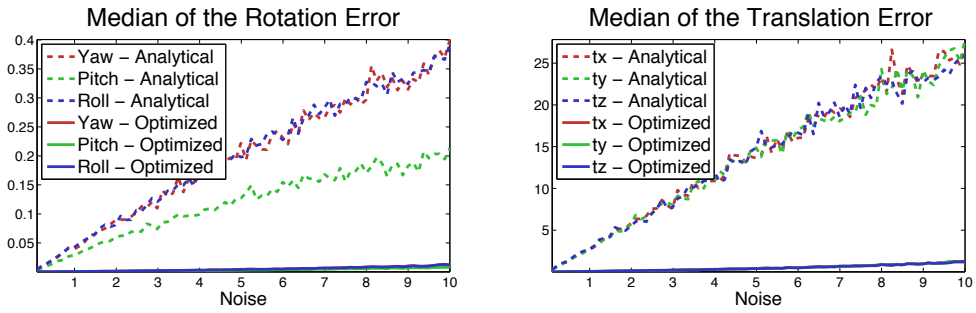
Figure 15.2: *In this figure we show the evaluation of the proposed method using synthetic data. We consider the error for the six parameters that define the pose: three rotation angles – radians; and three for the translation. In Figure (a) we show the results as a function of the* `Noise` *variable. In Figure (b) we show the results for different number of lines.*

cube whose side length is defined by the variable `Deviation from Central Case`. The directions are thus computed as

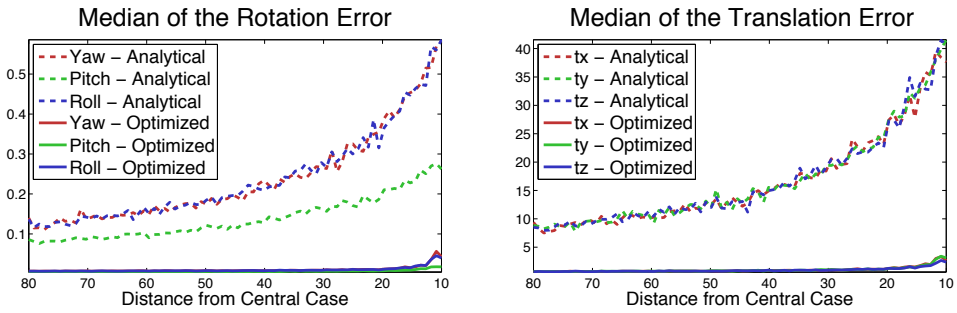
$$\mathbf{h}_{j,i}^{(\mathcal{C})} = \left(\tilde{\mathbf{h}}_{j,i}^{(\mathcal{C})} - \overline{\mathbf{h}}_{j,i}^{(\mathcal{C})}\right) \Big/ \left|\tilde{\mathbf{h}}_{j,i}^{(\mathcal{C})} - \overline{\mathbf{h}}_{j,i}^{(\mathcal{C})}\right| . \tag{15.3}$$

Note that when the variable `Deviation from Central Case` tends to zero, the camera model tends to be central. When it gets higher, it tends to the general case. An example with `Deviation from Central Case` equal to twenty is shown in Figure 15.1(b).

For the first evaluation, we consider data with noise. Instead of consid-



(a) Similar to the case of Figure 15.2(a) but, instead of the general case, we consider `Deviation from Central Case` variable equals to twenty – Figure 15.1(b).



(b) Evaluation of the proposed approach as a function of the `Deviation from Central Case`. We consider the `Noise` variable as 10, $N_i = 40\ \forall i$ and $M = 10$.

Figure 15.3: *In Figure (a) we show the evaluation of the algorithm as a function of the noise, for* `Deviation from Central Case` *equals to twenty. In Figure (b) we study the variation of the error as a function of the* `Deviation from Central Case`*. For each value of the evaluation variable, we consider* 10^3 *trials as described in the text.*

ering "projection lines" as described in Equation (15.2), we use

$$\mathcal{H}_{j,i}^{(\mathcal{C})} \doteq \left(\mathbf{h}_{j,i}^{(\mathcal{C})}, \left(\widetilde{\mathbf{h}}_{j,i}^{(\mathcal{C})} + \boldsymbol{e}_{j,i}^{(\mathcal{C})}\right) \times \mathbf{h}_{j,i}^{(\mathcal{C})}\right) \tag{15.4}$$

where vector $\boldsymbol{e}_{j,i}^{(\mathcal{C})}$ has random direction and random norm with standard deviation equal to the `Noise` variable. We vary the `Noise` variable from one to ten and the results are shown in Figures 15.2(a) and 15.3(a) for the general case and for the case where `Deviation from Central Case` is equal to twenty, respectively. All the results presented in this section were obtained using non-linear optimization with initial values determined using the analytical approach.

In Figure 15.2(b) we evaluate the proposed method by varying the number

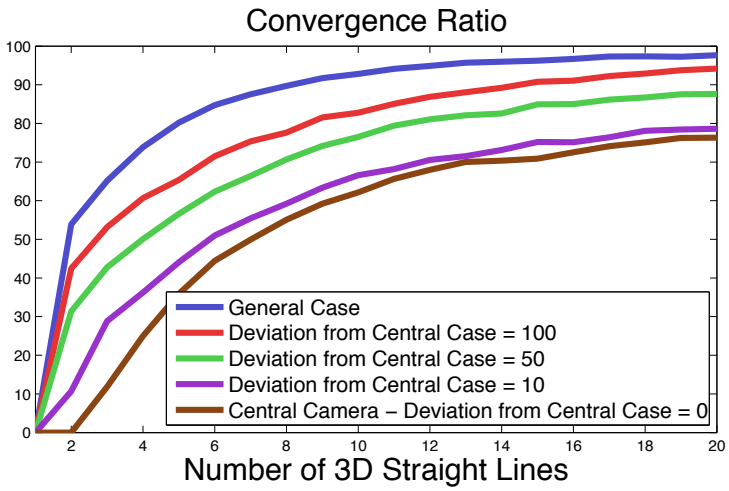


Figure 15.4: *In this figure we evaluate the convergence ratio of the non-linear method proposed in Section 14.3. For this purpose, we vary the number of known 3D straight lines. In addition, we also consider different values for* `Deviation from Central Case` *variable. Note that we consider the case of central camera –* `Deviation from Central Case` *equal to zero.*

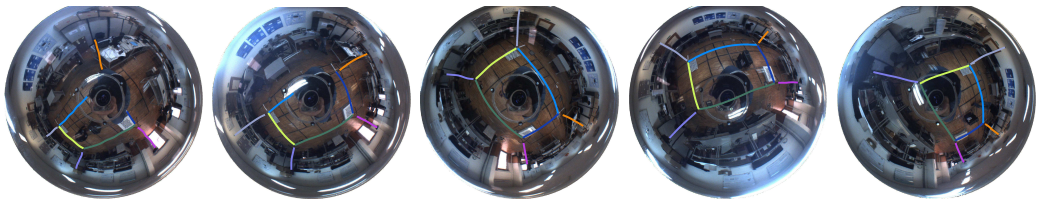
of 3D lines used to compute the pose. We consider the general case and `Noise` variable equal to 20.

In addition, we evaluate the proposed approach as a function of the `Deviation from Central Case` variable – we vary the evaluation variable from 80 to 10. For that purpose, we consider the `Noise` variable as 10. The results are shown in Figure 15.3(b).

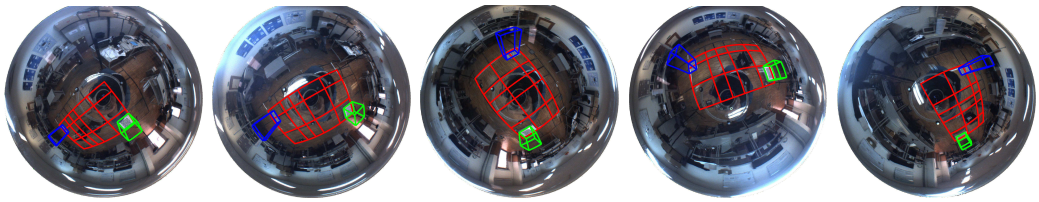
15.2 Convergence of the Non-Linear Method

In the previous experiments we considered the use of the non-linear optimization method to refine the solutions that are obtained from the analytical approach. To further evaluate the non-linear optimization approach defined by Equation (14.17), we analyze its convergence by using inital values that are randomly generated. For that purpose, we consider noiseless synthetic data. The data-set is generated as described in the previous section. For initial values, we consider $\mathbf{t}_0 = \mathbf{0}_3$ and rotation matrix $\mathbf{R}_0$ computed using random rotation angles.

We tested five different values for the variable `Distance from Central Case`. The first case is thus the general case. We also consider values of 100, 50, 10 and 0. Note that when this variable is 0 we have a central camera



(a) Five examples of the images that form the sequence. The colored curves represent the images of known 3D line coordinates. The color of the curves correspond to the color of the known coordinates of the 3D straight lines shown in the Figure 15.6.



(b) Since we know the localization of the non-central catadioptric camera in the world coordinate system, we can create objects in the world and project them to the image [ATR11]. We test the proposed pose estimation using an application of augmented reality. The 3D generated objects are shown in Figure 15.6. In this figure, we show an example of the proposed augmented reality for the same two images of (a).

Figure 15.5: *In Figure (a) we show five examples of images taken from the non-central catadioptric camera. The image curves marked in the images correspond to the data set used for the computation of the pose. In Figure (b) we test the proposed method using an application of augmented reality. The created objects are shown in Figure 15.6.*

model. In addition, we consider $N_i = 40$, for all i.

For each one of these different configurations, we evaluate the non-linear solution for different number of known 3D straight lines. The convergence ratio was computed considering 10^4 trials for each number of lines. We consider that a solution converges if the norm of the errors for the translation and rotation parameters is smaller then 10^{-5}.

15.3 Experiments with Real Images

For the experiments with real data, we consider a non-central calibrated catadioptric camera made up by a perspective camera and a spherical mirror.

We consider a sequence of images taken by moving the camera through a path in the lab. In each image, we determine eight curves that correspond to known straight lines in the world coordinate system. Five examples of these images are shown in Figure 15.5(a). The coordinates of the eight known 3D

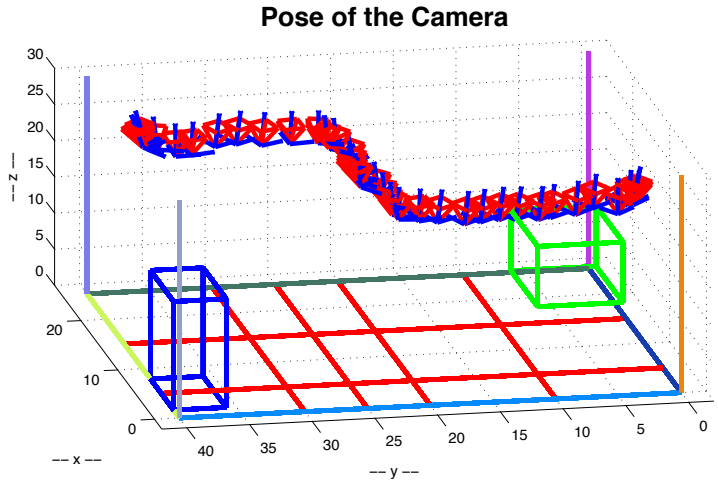


Figure 15.6: *In this figure we show the reconstruction of the motion obtained from a sequence of images taken from the non-central catadioptric camera – formed with a perspective camera and a spherical mirror. In the graphic, we show the recovered position of the perspective camera.*

straight lines in the world coordinate system are shown by the correspondent colors in the 3D Figure 15.6. Using the described data set and the method proposed in this article, we estimate the pose for the sequence. The results are shown in Figure 15.6. In this figure, we show the position and orientation of the perspective camera of the non-central catadioptric system.

The estimation of the pose allows the determination of the localization of the camera in the world coordinate system. Therefore and since we are using a non-central catadioptric camera with a spherical mirror, points in the world can be mapped into the image [ATR11]. To further evaluate the proposed solution for the pose, we consider an application of augmented reality, with the following objects:

- Extra 3D straight lines on the lab's floor – represented as red lines in Figure 15.6;
- Two 3D rectangular parallelepipeds on the chess-boards pattern, located on the floor of the lab. These two parallelepipeds are represented in green and blue – see Figure 15.6.

For each object, we project the points that make up the objects into the images that form the sequence. Five examples of the application of the augmented reality are shown in Figure 15.5(b).

Chapter 16

Discussion

In Section 14.1 we proposed an analytical approach to estimate the pose. When considering general non-central cameras and from Figure 15.2(a), we see that this method has an acceptable performance even in the case of noisy data. However the case of central cameras is degenerate for the non-iterative solution–see Section 14.2. The results obtained with the non-iterative approach deteriorate significantly for configurations close to the central camera models – see Figures 15.3(a)-(b). As a result, we also derived a non-linear optimization approach to estimate pose – Section 14.3.

From the results presented in Figures 15.2 and 15.3 one can conclude that non-linear optimization significantly decreases the errors due to both the noise and to configurations close to the central case. In addition, we also evaluated the convergence ratio of the non-linear optimization method. For the experiments whose results are presented in Figures 15.2 and 15.3 the estimates obtained using the analytical approach were used as initial values that were further improved using the non-linear optimization method – Section 14.3. To evaluate the convergence of the non-linear optimization approach random values for the rotation and zero values for the translation were used as initial estimates. Convergence ratios of up to 95% for the case of general non-central cameras were obtained – see Figure 15.4. When the geometry of the imaging system approaches central projection, the convergence ratio decreases. However, for the central case, the convergence ratios of up to 75% are obtained. Note that the degeneracy of central projection does not affect the non-linear procedure as significantly as it does affect the non-iterative solution. The main reason for that behavior it that multiple 3D lines are considered simultaneously in the problem of Equation (14.15).

In the experiments with real data – Section 15.3, we reconstructed the 3D motion from a sequence of images taken from a non-central imaging system. In addition, we also considered a simple example of an augmented reality

application that validates the solution for the pose.

Part V: Conclusions

Contents

17 Conclusions **110**
17.1 Model and Calibration of Smooth Camera Models 110
17.2 Minimal Absolute Pose Problem 111
17.3 Pose Using Lines 111

Chapter 17

Conclusions

In this dissertation, we addressed the problem of the calibration and pose of general camera models. We propose a new camera model aimed at representing smooth cameras and its correspondent calibration procedure. The conclusions are given in the Section 17.1.

In addition, we also considered the problem of the pose under the framework of generalized camera models. We consider both the problem of the estimation of the pose for minimal data – Section 17.2; and for non-minimal data – Section 17.3.

17.1 Model and Calibration of Smooth Camera Models

The proposed smooth camera model and calibration procedure can be used to calibrate complex smooth camera models, namely in the case of cameras for which no analytical model exists. Despite the fact that this model is for smooth imaging systems, it can also be used in special cases of non–smooth imaging systems.

This approach can model a camera with significantly less parameters than the discrete general camera model. For the model described in this thesis, the number of parameters does not depend on the image size. Instead of the $7NM$ parameters, for an $M \times N$ image, required by the discrete general camera model, this approach only requires $6\,(P+3)$ for P *control points* – see the Part II for more information.

The calibration procedure only requires the 3D coordinates of a single world point for each image point, whereas previous approaches require two or more 3D points for each image point. On the other hand, the calibration parameters which are estimated for a sub-set of image points are implicitly

generalized for all image pixels, which constitutes an important advantage of this method.

17.2 Minimal Absolute Pose Problem

In this thesis, we have presented a new homography-based formulation for the minimal three-point pose problem for generalized cameras – where the constraint that the projecting rays must intersect at a single points is not considered. This method can be used to estimate the pose for imaging models where reflection or refraction occurs. That is the case, for example, of catadioptric systems made up of arbitrary mirrors; or the case of a perspective camera looking at a water environment.

Previous approaches used algebra techniques that lead to a single variable eighth degree polynomial and the pose estimates are given by roots of the polynomial. The aim of the proposed methods is to compute the *homography* matrix that defines the pose between the camera and world coordinate systems. The main important step consists on the estimation of the intersection points of three quadrics.

The first step of the method proposed in this thesis consist on the estimation of a three degrees of freedom solution for the estimate of the *homography* matrix. Then, three constraints that enforce that the estimated matrix belongs to the space of the *homographies* are applied. To conclude, the *homography* matrix is decomposed into rotation and translation parameters that define the pose.

When compared with the state–of–the–art methods, the method proposed in this thesis performs similarly when taking into account the numerical errors and number of solutions. In addition, we can see that it has a similar computational effort as the fastest state–of–the–art method.

The main contribution of the proposed approach is the robustness of the method to critical configurations. We show that our method is significantly more robust than the state–of–the–art algorithms.

17.3 Pose Using Lines

In addition to the minimal absolute pose, in this dissertation we also propose a novel approach for non-minimal pose estimation for general camera models. Instead of the classical approach where correspondence between image and 3D points are known, our method requires the correspondence between known coordinates of 3D straight lines – in the world coordinate system, and

pixels that correspond to their images. Our approach significantly reduces the difficulties in the acquisition of the data set specially because it does not require the determination of correspondences between image and world points. In addition the method proved to be substantially robust against noise and for a wide variety of configurations, in the difficult setting of highly non-linear and non-central imaging systems.

Part VI: Appendices

Contents

A Unique Solution for the Estimation of the Plücker Coordinates Using Radial Basis Functions **114**

A.1 More on Radial Basis Functions 114
A.2 Introduction 117
A.3 *Rank* of matrix **M** 117
A.3.1 Proof that matrix M can have rank $6P + 17$ 119
A.3.2 The set $\lambda\left(\mathbf{K}^{-1}\mathbf{F}\left(\mathbf{D_{1,2}}\right)^{-1}\mathbf{F}\left(\mathbf{T_{1,2}}\right)\right)$ 126
A.4 Conclusions 127
A.5 Some Matrix Results 128
A.5.1 Rank of $\mathbf{D}_1 - \mathbf{L}\mathbf{D}_2\mathbf{L}^{-1}$ 128
A.5.2 Inverse of Matrices 129
A.5.3 Eigenvector Matrices 130
A.5.4 Intersection Subspace 131

B Analytical Solution for the Null-Space of Both N and M Matrices **133**

C Minimal Solvers **135**

C.1 Solver for the General Case 135
C.2 Solver for Central Camera Models 136

D Proof of Theorems and Propositions **138**

D.1 *proof* of the Theorem 9.1 138
D.2 *Proof* of the Theorem 9.2 140
D.3 *Proof* of the Proposition 9.2 142

Appendix A

Unique Solution for the Estimation of the Plücker Coordinates Using Radial Basis Functions

In this appendix we aim to prove the Theorem 5.1, formalized in Section 5.2.2. To prove that, under the conditions specified, the rank of the calibration matrix $\mathbf{M}$ is, *in general*, $\text{rank}(\mathbf{M}) = 6P + 17$, we consider random data $\{\mathbf{x}_i\}$ (coordinates of the image points) and $\{\mathbf{p}_i\}$ (coordinates of the 3D points). The goal is to prove that for these data, the probability of $\text{rank}(\mathbf{M}) \neq 6P+17$ is zero, or the probability of $\text{rank}(\mathbf{M}) = 6P + 17$ is one.

The *proof* is given by Sections A.3 and A.4. In Section A.1 we give a more detailed analysis of *Radial Basis Functions*. In Section A.2, we introduce the problem. In Section A.5, we derive some required matrix results, for the *proof*.

A.1 More on Radial Basis Functions

Radial Basis Functions are are frequently used in approximating functions ($f : \mathbb{R}^2 \mapsto \mathbb{R}$) by means of least squares fitting. For the case proposed in the thesis – Section 5.2, the interpolant is given by

$$s(\mathbf{x}) = a_0 + \mathbf{a}_{\mathrm{x}}^T \mathbf{x} + \sum_{i=1}^{P} w_i \phi(||\mathbf{x} - \mathbf{c}_i||) = \underbrace{\begin{bmatrix} \boldsymbol{\phi}(\mathbf{x}) & \mathbf{p}(\mathbf{x}) \end{bmatrix}}_{\mathbf{r}(\mathbf{x})} \underbrace{\begin{bmatrix} \mathbf{w} \\ \mathbf{a} \end{bmatrix}}_{\mathbf{h_{wa}}} \tag{A.1}$$

In this section we describe the typical problem of finding the unknown vector $\mathbf{h_{wa}}$ for a set of interpolant conditions

$$s(\mathbf{x}_i) = f(\mathbf{x}_i), \quad \forall i = 1, \ldots, P. \tag{A.2}$$

For a set $\{\mathbf{c}_i\}$, we define

$$\mathbf{\Phi} = \begin{bmatrix} \phi(||\mathbf{x}_1 - \mathbf{c}_1||) & \phi(||\mathbf{x}_1 - \mathbf{c}_3||) & \ldots & \phi(||\mathbf{x}_1 - \mathbf{c}_P||) \\ \phi(||\mathbf{x}_2 - \mathbf{c}_1||) & \phi(||\mathbf{x}_2 - \mathbf{c}_3||) & \ldots & \phi(||\mathbf{x}_2 - \mathbf{c}_P||) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(||\mathbf{x}_P - \mathbf{c}_1||) & \phi(||\mathbf{x}_P - \mathbf{c}_3||) & \ldots & \phi(||\mathbf{x}_P - \mathbf{c}_P||) \end{bmatrix} \tag{A.3}$$

where $\mathbf{\Phi} \in \mathbb{R}^{P \times P}$. In [Wen05, Buh03] the authors prove that, for $\{\mathbf{x}_i = \mathbf{c}_i\}$ where $i = 1, \ldots, P$, $\mathbf{\Phi}$ is *conditional positive definite*.

For *scattered* set $\{\mathbf{c}_i\}$, where $\{\mathbf{x}_i \neq \mathbf{c}_j\}$ for $i, j = 1, \ldots, P$, the authors of [QSW93, SW93] prove that $\mathbf{\Phi}$ is *conditional positive definite*, where each *control point* has to be associated to a data point $\{\mathbf{x}_i\}$, that satisfies $d \leqslant q\epsilon$, where $0 < \epsilon \leqslant 1$, $d = \max\{||\mathbf{x}_i - \mathbf{c}_i||\}$ and $2q = \min_{j \neq i}\{||\mathbf{c}_i - \mathbf{c}_j||\}$. Quak *et al.* [QSW93] also proved that $\phi_1(r) = (\beta_1^2 + r^2)^{1/2}$ and $\phi_2(r) = e^{-\beta_2 r^2}$ are good choices for *radial basis functions*, because, choosing an appropriate β_1 and β_2, they reduce the negative effects of small values of q and ϵ respectively.

From Equation (A.1), for a set P of $\{\mathbf{x}_i\}$ we can write

$$\mathbf{s} = \underbrace{\begin{bmatrix} \mathbf{\Phi} & \mathbf{K}^T \end{bmatrix}}_{\mathbf{R}} \mathbf{h_{wa}} \tag{A.4}$$

where $\mathbf{K} \in \mathbb{R}^{3 \times P}$ is the stacking of $\mathbf{p}(\mathbf{x}_i)$ and $\mathbf{s} = \begin{pmatrix} s(\mathbf{x}_1) & \ldots & s(\mathbf{x}_P) \end{pmatrix}^T$.

From Equation (A.4), we have $P + 3$ unknowns and only P equations. To eliminate the extra degrees of freedom, additional constraints are needed. We use the additional constraints resulting from the conditional positive definiteness of the space of solutions of $\mathbf{w}$ [Wen05]

$$\sum_{i=1}^{P} w_i \mathbf{p}(\mathbf{x}) = \mathbf{K}\mathbf{w} = \mathbf{0}. \tag{A.5}$$

Putting all together, we write

$$\begin{bmatrix} \mathbf{s} \\ \mathbf{0} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{\Phi} & \mathbf{K}^T \\ \mathbf{K} & \mathbf{0} \end{bmatrix}}_{\mathbf{\Gamma}} \mathbf{h_{wa}} \tag{A.6}$$

which has only one solution when $\mathbf{\Gamma} \in \mathbb{R}^{P+3 \times P+3}$ is *full–rank*.

We want to prove that $\mathbf{\Gamma}$ is *full–rank* or $\text{rank}\,(\mathbf{\Gamma}) = P+3$. Moreover, from linear algebra properties, $\text{nullity}\,(\mathbf{\Gamma}) = 0$ implies $\text{rank}\,(\mathbf{\Gamma}) = P+3$. Thus, computing the *null–space* of $\mathbf{\Gamma}$, we have

$$\begin{bmatrix} \mathbf{\Phi} & \mathbf{K}^T \\ \mathbf{K} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \mathbf{u} \end{bmatrix} = \mathbf{0}. \tag{A.7}$$

If we prove that the only solution to Equations (A.7) implies $\mathbf{v} = \mathbf{0}$ and $\mathbf{u} = \mathbf{0}$, then we conclude that $\text{nullity}\,(\mathbf{\Gamma}) = 0$ and as a result $\text{rank}\,(\mathbf{\Gamma}) = P+3$.

If $\mathbf{v} \in \mathbb{R}^P$ and $\mathbf{u} \in \mathbb{R}^3$ constitutes the *null-space* of the matrix $\mathbf{\Gamma}$, the following relation must be verified

$$\boldsymbol{w}^T \begin{bmatrix} \mathbf{\Phi} & \mathbf{K}^T \\ \mathbf{K} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \mathbf{u} \end{bmatrix} = 0, \tag{A.8}$$

for any vector $\boldsymbol{w} \in \mathbb{R}^{P+3}$. If we consider $\boldsymbol{w} = (\mathbf{v}, \mathbf{0})$, we have

$$\mathbf{v}^T\mathbf{\Phi}\mathbf{v} + \mathbf{v}^T\mathbf{K}^T\mathbf{u} = 0 \Rightarrow \mathbf{v}^T\mathbf{\Phi}\mathbf{v} + (\mathbf{K}\mathbf{v})^T\,\mathbf{u} = 0. \tag{A.9}$$

We know from Equation (A.7) that $\mathbf{K}\mathbf{v} = 0$. As a result, Equation (A.9) can be rewritten such that

$$\mathbf{v}^T\mathbf{\Phi}\mathbf{v} = 0. \tag{A.10}$$

We know from previous statements that $\mathbf{\Phi}$ is *conditional positive definite*, which means that $\mathbf{v}^T\mathbf{\Phi}\mathbf{v} > 0$ for any non–zero vector $\mathbf{v}$. As a result, Equation (A.7) is only verified if $\mathbf{v} = \mathbf{0}$. Note that this solution does not depend on the vector $\mathbf{u}$.

Since we already proved that the vector $\mathbf{v}$ must be $\mathbf{0}$, from the first row of Equation (A.7), we have

$$\mathbf{K}^T\mathbf{u} = \mathbf{0}. \tag{A.11}$$

If the set $\{\mathbf{x}_i\}$, for $i = 1, \ldots, P$ with $P \geqslant 3$, forms a *full–column rank* matrix $\mathbf{K}^T$, $\text{rank}\left(\mathbf{K}^T\right) = 3$, Equation (A.11) is only verified for $\mathbf{u} = \mathbf{0}$. As a result, we conclude that $\text{nullity}\,(\mathbf{\Gamma}) = 0$ and $\text{rank}\,(\mathbf{\Gamma}) = P+3$.

A.2 Introduction

The goal of the appendix is to study and analyze the relationship between the number N of point correspondences $\{\mathbf{x}_i \mapsto \mathbf{p}_i\}$ required for the calibration and the rank of the calibration matrix $\mathbf{M}$.

The calibration parameters are computed by estimating a non-zero vector $\text{vec}(\mathbf{H_{wa}})$ that satisfies

$$\underbrace{\begin{bmatrix} \mathbf{Q}(\mathbf{p}_1) \otimes \mathbf{r}(\mathbf{x}_1) \\ \mathbf{Q}(\mathbf{p}_2) \otimes \mathbf{r}(\mathbf{x}_2) \\ \vdots \\ \mathbf{Q}(\mathbf{p}_N) \otimes \mathbf{r}(\mathbf{x}_N) \\ \mathbf{D} \end{bmatrix}}_{\mathbf{M}} \text{vec}(\mathbf{H_{wa}}) = \mathbf{0} \tag{A.12}$$

where $\mathbf{Q}(\mathbf{p}_i)$ is the incident relation between a point in the world $\mathbf{p}_i \in \mathbb{R}^3$ and a line generated from an image point $\mathbf{x}_i$

$$\mathbf{Q}(\mathbf{p}_i) = \begin{bmatrix} \widehat{\mathbf{p}}_i & -\mathbf{I} \\ \mathbf{0}^T & \mathbf{p}_i^T \end{bmatrix} \tag{A.13}$$

To have a unique solution, we must have $\text{nullity}(\mathbf{M}) = 1$ and the solution is any element of the right *null–space* exept the trivial solution $\text{vec}(\mathbf{H_{wa}}) = \mathbf{0}$.

A.3 *Rank* of matrix M

In this section, we study the relationship between the *rank* of matrix $\mathbf{M}$ (Equation (A.12)) and the number of point–correspondences N, used in the calibration process – Section 5.2.2.

Since permuting rows does not change the *rank* of a matrix, $\text{rank}(\mathbf{A}) = \mathbf{C}(\mathbf{ZM})$, for any permmutation matrix $\mathbf{Z}$, and we can study the *rank* of $\mathbf{A}$, instead of $\mathbf{M}$.

From Equation (A.12) and from the definition of the *Kronecker* product Equation (3.4), we can find a matrix $\mathbf{A} = \mathbf{ZM}$ as Equation (A.14). where $\mathbf{Z}$ is a permutation matrix, $\mathbf{p}_i = \left(p_i^{(1)}, p_i^{(2)}, p_i^{(3)}\right)$ and $\mathbf{r}_i = \mathbf{r}(\mathbf{x}_i)$, where $\mathbf{r}(\mathbf{x}_i)$ is

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & -p_1^{(3)}\mathbf{r}_1 & p_1^{(2)}\mathbf{r}_1 & \mathbf{r}_1 & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & -p_N^{(3)}\mathbf{r}_N & p_N^{(2)}\mathbf{r}_N & \mathbf{r}_N & \mathbf{0} & \mathbf{0} \\ p_1^{(3)}\mathbf{r}_1 & \mathbf{0} & -p_1^{(1)}\mathbf{r}_1 & \mathbf{0} & \mathbf{r}_1 & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ p_N^{(3)}\mathbf{r}_N & \mathbf{0} & -p_N^{(1)}\mathbf{r}_N & \mathbf{0} & \mathbf{r}_N & \mathbf{0} \\ -p_1^{(2)}\mathbf{r}_1 & p_1^{(1)}\mathbf{r}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{r}_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -p_N^{(2)}\mathbf{r}_N & p_N^{(1)}\mathbf{r}_N & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{r}_N \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & p_1^{(1)}\mathbf{r}_1 & p_1^{(2)}\mathbf{r}_1 & p_1^{(3)}\mathbf{r}_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & p_N^{(1)}\mathbf{r}_N & p_N^{(2)}\mathbf{r}_N & p_N^{(3)}\mathbf{r}_N \\ & & \mathbf{D} & & & \end{bmatrix} \tag{A.14}$$

as described in Section A.1, Equation (A.1). We define $\mathbf{E}$ and $\mathbf{F}$ as

$$\mathbf{E} = \begin{bmatrix} \mathbf{0} & -p_1^{(3)}\mathbf{r}_1 & p_1^{(2)}\mathbf{r}_1 & \mathbf{r}_1 & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & -p_N^{(3)}\mathbf{r}_N & p_N^{(2)}\mathbf{r}_N & \mathbf{r}_N & \mathbf{0} & \mathbf{0} \\ p_1^{(3)}\mathbf{r}_1 & \mathbf{0} & -p_1^{(1)}\mathbf{r}_1 & \mathbf{0} & \mathbf{r}_1 & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ p_N^{(3)}\mathbf{r}_N & \mathbf{0} & -p_N^{(1)}\mathbf{r}_N & \mathbf{0} & \mathbf{r}_N & \mathbf{0} \\ -p_1^{(2)}\mathbf{r}_1 & p_1^{(1)}\mathbf{r}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{r}_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -p_N^{(2)}\mathbf{r}_N & p_N^{(1)}\mathbf{r}_N & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{r}_N \end{bmatrix} \tag{A.15}$$

and

$$\mathbf{F} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & p_1^{(1)}\mathbf{r}_1 & p_1^{(2)}\mathbf{r}_1 & p_1^{(3)}\mathbf{r}_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & p_N^{(1)}\mathbf{r}_N & p_N^{(2)}\mathbf{r}_N & p_N^{(3)}\mathbf{r}_N \end{bmatrix} \tag{A.16}$$

where $\mathbf{E} \in \mathbb{R}^{3N\times 6P+18}$, $\mathbf{F} \in \mathbb{R}^{N\times 6P+18}$ and we can rewrite $\mathbf{A}$ as

$$\mathbf{A} = \begin{bmatrix} \mathbf{E} \\ \mathbf{F} \\ \mathbf{D} \end{bmatrix}. \tag{A.17}$$

We can see that the rows of $\mathbf{F}$ are linearly dependent on the rows of $\mathbf{E}$.

A.3.1 Proof that matrix M can have rank $6P + 17$

Since the rows of the $\mathbf{F}$ are linearly dependent on the rows of $\mathbf{E}$, we ignore the rows of $\mathbf{F}$ for the rest of the section. Thus, we consider the matrix $\mathbf{A}^{(1)} \in \mathbb{R}^{(3N+18)\times(6P+18)}$ such that

$$\mathbf{A}^{(1)} = \mathbf{Z}^{(1)} \begin{bmatrix} \mathbf{E}^{(1)} \\ \mathbf{D} \end{bmatrix} \tag{A.18}$$

and if we define $\mathbf{D} \in \mathbb{R}^{18\times(6P+18)}$ as

$$\mathbf{D} = \begin{bmatrix} \mathbf{0} & -\mathbf{\Delta}_1\mathbf{P}_1 & \mathbf{\Delta}_2\mathbf{P}_1 & \mathbf{P}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{\Delta}_1\mathbf{P}_1 & \mathbf{0} & -\mathbf{\Delta}_3\mathbf{P}_1 & \mathbf{0} & \mathbf{P}_1 & \mathbf{0} \\ -\mathbf{\Delta}_2\mathbf{P}_1 & \mathbf{\Delta}_3\mathbf{P}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{P}_1 \\ \mathbf{0} & -\mathbf{\Delta}_4\mathbf{P}_2 & \mathbf{\Delta}_5\mathbf{P}_2 & \mathbf{P}_2 & \mathbf{0} & \mathbf{0} \\ \mathbf{\Delta}_4\mathbf{P}_2 & \mathbf{0} & -\mathbf{\Delta}_6\mathbf{P}_2 & \mathbf{0} & \mathbf{P}_2 & \mathbf{0} \\ -\mathbf{\Delta}_5\mathbf{P}_2 & \mathbf{\Delta}_6\mathbf{P}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{P}_2 \end{bmatrix} \tag{A.19}$$

where

$$\mathbf{P}_1 = \begin{bmatrix} \mathbf{K}_1 & \mathbf{0} \end{bmatrix}, \quad \mathbf{P}_2 = \begin{bmatrix} \mathbf{K}_2 & \mathbf{0} \end{bmatrix} \tag{A.20}$$

$\mathbf{P}_i \in \mathbb{R}^{3\times(P+3)}$ and $\mathbf{K}_1, \mathbf{K}_2 \in \mathbb{R}^{3\times P}$ are the stacking of the set $\{\mathbf{p}(\mathbf{x}_i)\}$ for $i = 1, \ldots, P$, and $\{\mathbf{p}(\mathbf{x}_i)\}$ for $i = P+1, \ldots, 2P$ respectively. Matrices $\mathbf{\Delta}_i \in \mathbb{R}^{3\times 3}$ are random diagonal matrices, where $\xi_j^{(i)}$, for $j = 1, 2, 3$ are their diagonal elements.

We see that $\mathbf{E}^{(1)} \in \mathbb{R}^{3N\times(6P+18)}$ and $\mathbf{D} \in \mathbb{R}^{18\times(6P+18)}$. Thus, to have $\operatorname{rank}\left(\mathbf{A}^{(1)}\right) = 6P + 17$, we need at least $N = 2P$.

For a permutation matrix $\mathbf{Z}^{(1)}$, $\mathbf{E}$ with $N = 2P$ and $\mathbf{D}$ as in Equation (A.19), we define $\mathbf{A}^{(1)}$ as in Equation (A.21).

We can express $\mathbf{A}^{(1)}$ as a block of $(P + 3) \times (P + 3)$ matrices

$$\left(\mathbf{A}^{(1)}\right)^T = \begin{bmatrix} \mathbf{0} & \boldsymbol{\Gamma}_1^T\mathbf{T}_1 & -\boldsymbol{\Gamma}_1^T\mathbf{D}_1 & \mathbf{0} & \boldsymbol{\Gamma}_2^T\mathbf{T}_2 & -\boldsymbol{\Gamma}_2^T\mathbf{D}_2 \\ -\boldsymbol{\Gamma}_1^T\mathbf{T}_1 & \mathbf{0} & \boldsymbol{\Gamma}_1^T\mathbf{S}_1 & -\boldsymbol{\Gamma}_2^T\mathbf{T}_2 & \mathbf{0} & \boldsymbol{\Gamma}_2^T\mathbf{S}_2 \\ \boldsymbol{\Gamma}_1^T\mathbf{D}_1 & -\boldsymbol{\Gamma}_1^T\mathbf{S}_1 & \mathbf{0} & \boldsymbol{\Gamma}_2^T\mathbf{D}_2 & -\boldsymbol{\Gamma}_2^T\mathbf{S}_2 & \mathbf{0} \\ \boldsymbol{\Gamma}_1^T & \mathbf{0} & \mathbf{0} & \boldsymbol{\Gamma}_2^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Gamma}_1^T & \mathbf{0} & \mathbf{0} & \boldsymbol{\Gamma}_2^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Gamma}_1^T & \mathbf{0} & \mathbf{0} & \boldsymbol{\Gamma}_2^T \end{bmatrix} \tag{A.22}$$

where $\mathbf{D}_i, \mathbf{T}_i, \mathbf{S}_i \in \mathbb{R}^{(P+3)\times(P+3)}$ are diagonal matrices, whose diagonal elements are equal to respectively $p_n^{(m)}$ and to corresponding elements of diagonal matrices $\boldsymbol{\Delta}_i$ ($\xi_j^{(i)}$, with $j = 1, \ldots, 3$). For instance, diagonal matrix $\mathbf{T}_1$ is

$$\mathbf{T}_1 = \begin{bmatrix} p_1^{(3)} & \ldots & 0 & 0 & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & \ldots & p_P^{(3)} & 0 & 0 & 0 \\ 0 & \ldots & 0 & \xi_1^{(1)} & 0 & 0 \\ 0 & \ldots & 0 & 0 & \xi_2^{(1)} & 0 \\ 0 & \ldots & 0 & 0 & 0 & \xi_3^{(1)} \end{bmatrix}. \tag{A.23}$$

Matrices $\boldsymbol{\Gamma}_1$ and $\boldsymbol{\Gamma}_2$ are

$$\boldsymbol{\Gamma}_1 = \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{P}_1 \end{bmatrix} \quad \text{and} \quad \boldsymbol{\Gamma}_2 = \begin{bmatrix} \mathbf{R}_2 \\ \mathbf{P}_2 \end{bmatrix} \tag{A.24}$$

where $\boldsymbol{\Gamma}_i \in \mathbb{R}^{(P+3)\times(P+3)}$, and

$$\mathbf{R}_1 = \begin{bmatrix} \mathbf{r}_1 \\ \vdots \\ \mathbf{r}_P \end{bmatrix} \quad \text{and} \quad \mathbf{R}_2 = \begin{bmatrix} \mathbf{r}_{P+1} \\ \vdots \\ \mathbf{r}_{2P} \end{bmatrix}. \tag{A.25}$$

Matrices $\mathbf{R}_i \in \mathbb{R}^{P\times(P+3)}$ and $\mathbf{P}_1$ and $\mathbf{P}_2$ are as in Equation (A.20).

We assume that the conditions described in Section A.1 are met for $\boldsymbol{\Gamma}_1$ and

$$\mathbf{A}^{(1)} = \begin{bmatrix}
\mathbf{0} & -p_1^{(3)}\mathbf{r}_1 & p_1^{(2)}\mathbf{r}_1 & \mathbf{r}_1 & \mathbf{0} & \mathbf{0} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\mathbf{0} & -p_P^{(3)}\mathbf{r}_P & p_P^{(2)}\mathbf{r}_P & \mathbf{r}_P & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & -\boldsymbol{\Delta}_1\mathbf{P}_1 & \boldsymbol{\Delta}_2\mathbf{P}_1 & \mathbf{P}_1 & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & -p_{P+1}^{(3)}\mathbf{r}_{P+1} & p_{P+1}^{(2)}\mathbf{r}_{P+1} & \mathbf{r}_{P+1} & \mathbf{0} & \mathbf{0} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\mathbf{0} & -p_{2P}^{(3)}\mathbf{r}_{2P} & p_{2P}^{(2)}\mathbf{r}_{2P} & \mathbf{r}_{2P} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & -\boldsymbol{\Delta}_4\mathbf{P}_2 & \boldsymbol{\Delta}_5\mathbf{P}_2 & \mathbf{P}_2 & \mathbf{0} & \mathbf{0} \\
p_1^{(3)}\mathbf{r}_1 & \mathbf{0} & -p_1^{(1)}\mathbf{r}_1 & \mathbf{0} & \mathbf{r}_1 & \mathbf{0} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
p_P^{(3)}\mathbf{r}_P & \mathbf{0} & -p_P^{(1)}\mathbf{r}_P & \mathbf{0} & \mathbf{r}_P & \mathbf{0} \\
\boldsymbol{\Delta}_1\mathbf{P}_1 & \mathbf{0} & -\boldsymbol{\Delta}_3\mathbf{P}_1 & \mathbf{0} & \mathbf{P}_1 & \mathbf{0} \\
p_{P+1}^{(3)}\mathbf{r}_{P+1} & \mathbf{0} & -p_{P+1}^{(1)}\mathbf{r}_{P+1} & \mathbf{0} & \mathbf{r}_{P+1} & \mathbf{0} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
p_{2P}^{(3)}\mathbf{r}_{2P} & \mathbf{0} & -p_{2P}^{(1)}\mathbf{r}_{2P} & \mathbf{0} & \mathbf{r}_{2P} & \mathbf{0} \\
\boldsymbol{\Delta}_4\mathbf{P}_2 & \mathbf{0} & -\boldsymbol{\Delta}_6\mathbf{P}_2 & \mathbf{0} & \mathbf{P}_2 & \mathbf{0} \\
-p_1^{(2)}\mathbf{r}_1 & p_1^{(1)}\mathbf{r}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{r}_1 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
-p_P^{(2)}\mathbf{r}_P & p_P^{(1)}\mathbf{r}_P & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{r}_P \\
-\boldsymbol{\Delta}_2\mathbf{P}_1 & \boldsymbol{\Delta}_3\mathbf{P}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{P}_1 \\
-p_{P+1}^{(2)}\mathbf{r}_{P+1} & p_{P+1}^{(1)}\mathbf{r}_{P+1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{r}_{P+1} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
-p_{2P}^{(2)}\mathbf{r}_{2P} & p_{2P}^{(1)}\mathbf{r}_{2P} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{r}_{2P} \\
-\boldsymbol{\Delta}_5\mathbf{P}_2 & \boldsymbol{\Delta}_6\mathbf{P}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{P}_2
\end{bmatrix} \tag{A.21}$$

where $\mathbf{A}^{(1)} \in \mathbb{R}^{(6P+18)\times(6P+18)}$.

$\mathbf{\Gamma}_2$, which means that these matrices are *full–rank*. Let us define a matrix

$$\mathbf{N} = \mathbf{G}_1 \left(\mathbf{A}^{(1)}\right)^T \mathbf{G}_2 \tag{A.26}$$

where

$$\mathbf{G}_1 = \begin{bmatrix} \left(\mathbf{\Gamma}_1^T\right)^{-1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \left(\mathbf{\Gamma}_1^T\right)^{-1} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \left(\mathbf{\Gamma}_1^T\right)^{-1} \end{bmatrix} \text{ and } \mathbf{G}_2 = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{L}^{-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{L}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{L}^{-1} \end{bmatrix}. \tag{A.27}$$

Matrices $\mathbf{G}_1, \mathbf{G}_2, \in \mathbb{R}^{(6P+18)\times(6P+18)}$ are *full–rank* matrices and $\mathbf{L} = \left(\mathbf{\Gamma}_1^T\right)^{-1} \mathbf{\Gamma}_2^T$. The pre or post–multiplication by any *full–rank* matrix does not change the *rank* of a matrix. Thus,

$$\text{rank}\,(\mathbf{N}) = \text{rank}\left(\left(\mathbf{A}^{(1)}\right)^T\right) \text{ and } \text{rank}\left(\mathbf{A}^{(1)}\right) = \text{rank}\left(\left(\mathbf{A}^{(1)}\right)^T\right) \tag{A.28}$$

From Section 3.1, we can see that $\text{rank}\,(\mathbf{N}) + \text{nullity}\,(\mathbf{N}) = 6P + 18$. Thus, if we want $\text{rank}\left(\mathbf{A}^{(1)}\right) = \text{rank}\,(\mathbf{N}) = 6P + 17$, we must have $\text{nullity}\,(\mathbf{N}) = 1$. As a result, we need to prove that the *nullity* of $\mathbf{N}$ is one, where $\mathbf{N}$ is

$$\mathbf{N} = \begin{bmatrix} \mathbf{0} & \mathbf{T}_1 & -\mathbf{D}_1 & \mathbf{0} & \mathbf{L}\mathbf{T}_2\mathbf{L}^{-1} & -\mathbf{L}\mathbf{D}_2\mathbf{L}^{-1} \\ -\mathbf{T}_1 & \mathbf{0} & \mathbf{S}_1 & -\mathbf{L}\mathbf{T}_2\mathbf{L}^{-1} & \mathbf{0} & \mathbf{L}\mathbf{S}_2\mathbf{L}^{-1} \\ \mathbf{D}_1 & -\mathbf{S}_1 & \mathbf{0} & \mathbf{L}\mathbf{D}_2\mathbf{L}^{-1} & -\mathbf{L}\mathbf{S}_2\mathbf{L}^{-1} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}, \tag{A.29}$$

which means that $\mathbf{N}\mathbf{v} = \mathbf{0}$ must have a one dimensional subspace of solutions. We consider that $\mathbf{v} = (\mathbf{v}_1, \dots, \mathbf{v}_6) \in \mathbb{R}^{(6P+18)}$ where $\mathbf{v}_i \in \mathbb{R}^{(P+3)}$. From the three last rows of Equation (A.29), we see that the *null–space* of $\mathbf{N}$ must verify

$$\mathbf{v}_1 = -\mathbf{v}_4 \tag{A.30}$$

$$\mathbf{v}_2 = -\mathbf{v}_5 \tag{A.31}$$

$$\mathbf{v}_3 = -\mathbf{v}_6. \tag{A.32}$$

Getting the second, fifth and sixth row of equations of matrix $\mathbf{N}$ and the

third, fifth and sixth row of equations of matrix $\mathbf{N}$ respectively, we can define the following constraints

$$\begin{bmatrix} -\mathbf{T}_1 & \mathbf{0} & \mathbf{S}_1 \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \end{bmatrix} = - \begin{bmatrix} -\mathbf{L}\mathbf{T}_2\mathbf{L}^{-1} & \mathbf{0} & \mathbf{L}\mathbf{S}_2\mathbf{L}^{-1} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{v}_4 \\ \mathbf{v}_5 \\ \mathbf{v}_6 \end{bmatrix} \tag{A.33}$$

and

$$\begin{bmatrix} \mathbf{D}_1 & -\mathbf{S}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \end{bmatrix} = - \begin{bmatrix} \mathbf{L}\mathbf{D}_2\mathbf{L}^{-1} & -\mathbf{L}\mathbf{S}_2\mathbf{L}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{v}_4 \\ \mathbf{v}_5 \\ \mathbf{v}_6 \end{bmatrix}. \tag{A.34}$$

If the diagonal elements of $\mathbf{D}_1$ and $\mathbf{T}_1$ are different from zero, we can define matrices $\mathbf{B}$ and $\mathbf{C}$ as

$$\begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \end{bmatrix} = \mathbf{B} \begin{bmatrix} \mathbf{v}_4 \\ \mathbf{v}_5 \\ \mathbf{v}_6 \end{bmatrix}, \quad \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \end{bmatrix} = \mathbf{C} \begin{bmatrix} \mathbf{v}_4 \\ \mathbf{v}_5 \\ \mathbf{v}_6 \end{bmatrix}. \tag{A.35}$$

Using Appendix A.5.2, we obtain $\mathbf{B}$ and $\mathbf{C}$ such that

$$\begin{aligned} \mathbf{B} = - \begin{bmatrix} -\mathbf{T}_1 & \mathbf{0} & \mathbf{S}_1 \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}^{-1} \begin{bmatrix} -\mathbf{L}\mathbf{T}_2\mathbf{L}^{-1} & \mathbf{0} & \mathbf{L}\mathbf{S}_2\mathbf{L}^{-1} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} = \\ - \begin{bmatrix} -\mathbf{T}_1^{-1} & \mathbf{0} & \mathbf{T}_1^{-1}\mathbf{S}_1 \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} -\mathbf{L}\mathbf{T}_2\mathbf{L}^{-1} & \mathbf{0} & \mathbf{L}\mathbf{S}_2\mathbf{L}^{-1} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \end{aligned} \tag{A.36}$$

and

$$\begin{aligned} \mathbf{C} = - \begin{bmatrix} \mathbf{D}_1 & -\mathbf{S}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{L}\mathbf{D}_2\mathbf{L}^{-1} & -\mathbf{L}\mathbf{S}_2\mathbf{L}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} = \\ - \begin{bmatrix} \mathbf{D}_1^{-1} & \mathbf{D}_1^{-1}\mathbf{S}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{L}\mathbf{D}_2\mathbf{L}^{-1} & -\mathbf{L}\mathbf{S}_2\mathbf{L}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \end{aligned} \tag{A.37}$$

Developing the multiplication we get

$$\mathbf{B} = -\begin{bmatrix} \mathbf{T}_1^{-1}\mathbf{L}\mathbf{T}_2\mathbf{L}^{-1} & \mathbf{0} & -\mathbf{T}_1^{-1}\mathbf{L}\mathbf{S}_2\mathbf{L}^{-1} + \mathbf{T}_1^{-1}\mathbf{S}_1 \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{A.38}$$

$$\mathbf{C} = -\begin{bmatrix} \mathbf{D}_1^{-1}\mathbf{L}\mathbf{D}_2\mathbf{L}^{-1} & -\mathbf{D}_1^{-1}\mathbf{L}\mathbf{S}_2\mathbf{L}^{-1} + \mathbf{D}_1^{-1}\mathbf{S}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{A.39}$$

From linear algebra properties, the sets of *eigenvalues* of $\mathbf{B}$ and $\mathbf{C}$ are respectively

$$\lambda(\mathbf{B}) = \lambda\left(-\mathbf{T}_1^{-1}\mathbf{L}\mathbf{T}_2\mathbf{L}^{-1}\right) \cup \lambda(-\mathbf{I}) \cup \lambda(-\mathbf{I}) \tag{A.40}$$

and

$$\lambda(\mathbf{C}) = \lambda\left(-\mathbf{D}_1^{-1}\mathbf{L}\mathbf{D}_2\mathbf{L}^{-1}\right) \cup \lambda(-\mathbf{I}) \cup \lambda(-\mathbf{I}) \tag{A.41}$$

We define $\mathbf{\Sigma_B}, \mathbf{\Sigma_C}$ as diagonal matrices, whose diagonal elements are the *eigenvalues* of $\mathbf{B}$ and $\mathbf{C}$ respectively

$$\mathbf{\Sigma_B} = \begin{bmatrix} \mathbf{\Sigma}_{-\mathbf{T}_1^{-1}\mathbf{L}\mathbf{T}_2\mathbf{L}^{-1}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{I} \end{bmatrix} \quad \text{and} \quad \mathbf{\Sigma_C} = \begin{bmatrix} \mathbf{\Sigma}_{-\mathbf{D}_1^{-1}\mathbf{L}\mathbf{D}_2\mathbf{L}^{-1}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{I} \end{bmatrix}. \tag{A.42}$$

We can see that the solutions for Equations (A.33) that verify Equations (A.30), (A.30) and (A.32), are defined by the *eigenvectors*, that correspond to the *eigenvalues* $\lambda(\mathbf{B})$ that are equal to -1. On the other hand, solutions for Equations (A.34) that verify Equations (A.30), (A.30) and (A.32), are defined by the *eigenvectors*, that correspond to the *eigenvalues* $\lambda(\mathbf{C})$ that are equal to -1.

If we consider that $\mathbf{T}_i$ and $\mathbf{D}_i$ are random matrices, we can conclude that the probability of $\lambda\left(\mathbf{T}_1^{-1}\mathbf{L}\mathbf{T}_2\mathbf{L}^{-1}\right) \cap \lambda(-\mathbf{I}) = \varnothing$ and $\lambda\left(\mathbf{D}_1^{-1}\mathbf{L}\mathbf{D}_2\mathbf{L}^{-1}\right) \cap \lambda(-\mathbf{I}) = \varnothing$ is equal to one.

From Appendix A.5.3, we conclude that the matrices that correspond to the stacking of *eigenvectors* (*eigenvectors matrices*), $\mathbf{V}$ and $\mathbf{U}$ ($\mathbf{B} = \mathbf{V}\mathbf{\Sigma_B}\mathbf{V}^{-1}$ and $\mathbf{C} = \mathbf{U}\mathbf{\Sigma_C}\mathbf{U}^{-1}$) have the form

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}^{(1)} & \mathbf{0} & \mathbf{V}^{(2)} \\ \mathbf{0} & \mathbf{V}^{(3)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{V}^{(4)} \end{bmatrix} \quad \text{and} \quad \mathbf{U} = \begin{bmatrix} \mathbf{U}^{(1)} & \mathbf{U}^{(2)} & \mathbf{0} \\ \mathbf{0} & \mathbf{U}^{(3)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{U}^{(4)} \end{bmatrix} \tag{A.43}$$

where $\mathbf{V}, \mathbf{U} \in \mathbb{R}^{(3P+9)\times(3P+9)}$.

Since we are only interested in *eigenvectors* associated to *eigenvalues* equal to -1, we only consider the subspaces generated from matrices

$$\breve{\mathbf{V}} = \begin{bmatrix} \mathbf{0} & \mathbf{V}^{(2)} \\ \mathbf{V}^{(3)} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}^{(4)} \end{bmatrix} \quad \text{and} \quad \breve{\mathbf{U}} = \begin{bmatrix} \mathbf{U}^{(2)} & \mathbf{0} \\ \mathbf{U}^{(3)} & \mathbf{0} \\ \mathbf{0} & \mathbf{U}^{(4)} \end{bmatrix} \tag{A.44}$$

where $\breve{\mathbf{V}}, \breve{\mathbf{U}} \in \mathbb{R}^{(3P+9)\times(2P+6)}$.

However, we want solutions that verify $\mathbf{N}\mathbf{v} = \mathbf{0}$, which means that they must belong to both $\breve{\mathbf{V}}$ and $\breve{\mathbf{U}}$ subspaces. As a result, solutions must belong to the intersection of subspaces defined by $\breve{\mathbf{V}}$ and $\breve{\mathbf{U}}$.

From Appendix A.5.3 and Appendix A.5.4, we conclude that the intersection subspace is defined by the column space of

$$\mathbf{W} = \begin{bmatrix} * \\ \mathbf{I} \\ \mathbf{K} \end{bmatrix} \tag{A.45}$$

where $\mathbf{W} \in \mathbb{R}^{(3P+9)\times(P+3)}$. This means that, any linear combination of $\mathbf{W}$ columns ($\mathbf{W}\mathbf{a}$ for any $\mathbf{a} \neq \mathbf{0}$) is a solution for Equations (A.35) that verifies Equation (A.30), (A.31) and (A.32) where

$$\mathbf{v} = (*, \mathbf{a}, \mathbf{K}\mathbf{a}, *, -\mathbf{a}, -\mathbf{K}\mathbf{a}) \tag{A.46}$$

for any vector $\mathbf{a} \in \mathbb{R}^{P+3}$ different from zero.

However, from the first row of equations of $\mathbf{N}$, Equation (A.46) must verify

$$\mathbf{T}_1\mathbf{v}_2 + \mathbf{L}\mathbf{T}_2\mathbf{L}^{-1}\mathbf{v}_5 = \mathbf{D}_1\mathbf{v}_3 + \mathbf{L}\mathbf{D}_2\mathbf{L}^{-1}\mathbf{v}_6, \tag{A.47}$$

which from Equation (A.46) is equal to

$$\underbrace{\left(\mathbf{T}_1 - \mathbf{L}\mathbf{T}_2\mathbf{L}^{-1}\right)}_{\mathbf{F}(\mathbf{T}_{1,2})}\mathbf{a} = \underbrace{\left(\mathbf{D}_1 - \mathbf{L}\mathbf{D}_2\mathbf{L}^{-1}\right)}_{\mathbf{F}(\mathbf{D}_{1,2})}\mathbf{K}\mathbf{a}. \tag{A.48}$$

From Section A.5.1, the previous assumptions that $\lambda\left(\mathbf{T}_1^{-1}\mathbf{L}\mathbf{T}_2\mathbf{L}^{-1}\right) \cap \lambda\left(-\mathbf{I}\right) = \varnothing$ and $\lambda\left(\mathbf{D}_1^{-1}\mathbf{L}\mathbf{D}_2\mathbf{L}^{-1}\right) \cap \lambda\left(-\mathbf{I}\right) = \varnothing$ and assuming that $\mathbf{S}_1, \mathbf{S}_2$ are random matrices which implies that the probability of $\lambda\left(\mathbf{S}_1^{-1}\mathbf{L}\mathbf{S}_2\mathbf{L}^{-1}\right) \cap \lambda\left(-\mathbf{I}\right) = \varnothing$ is one, we see that $\operatorname{rank}\left(\mathbf{F}\left(\mathbf{T}_{1,2}\right)\right) = P+3$, $\operatorname{rank}\left(\mathbf{F}\left(\mathbf{D}_{1,2}\right)\right) = P+3$ and $\operatorname{rank}\left(\mathbf{K}\right) = P+3$. Thus, the constraint corresponding to Equation (A.47) can be rewritten as

$$\mathbf{K}^{-1}\mathbf{F}\left(\mathbf{D}_{1,2}\right)^{-1}\mathbf{F}\left(\mathbf{T}_{1,2}\right)\mathbf{a} = \mathbf{a}. \tag{A.49}$$

As a result, we can see that the dimension of the *null–space* of $\mathbf{N}$ is equal to the number of *eigenvalues* $\lambda\left(\mathbf{K}^{-1}\mathbf{F}\left(\mathbf{D}_{\mathbf{1,2}}\right)^{-1}\mathbf{F}\left(\mathbf{T}_{\mathbf{1,2}}\right)\right)$ that are equal to 1.

A.3.2 The set $\lambda\left(\mathbf{K}^{-1}\mathbf{F}\left(\mathbf{D}_{\mathbf{1,2}}\right)^{-1}\mathbf{F}\left(\mathbf{T}_{\mathbf{1,2}}\right)\right)$

In the previous section, we saw that $\operatorname{rank}\left(\mathbf{A}^{(1)}\right) = \operatorname{rank}\left(\mathbf{N}\right)$. On the other hand, we see that the $\operatorname{nullity}\left(\mathbf{N}\right)$ is equal to the number of *eigenvalues* $\lambda\left(\mathbf{K}^{-1}\mathbf{F}\left(\mathbf{D}_{\mathbf{1,2}}\right)^{-1}\mathbf{F}\left(\mathbf{T}_{\mathbf{1,2}}\right)\right) \cap \lambda\left(\mathbf{I}\right)$ and, since $\mathbf{N}$ is a square matrix, we know that $6P + 18 = \operatorname{rank}\left(\mathbf{N}\right) + \operatorname{nullity}\left(\mathbf{N}\right)$. As a result, $\operatorname{rank}\left(\mathbf{N}\right) = 6P + 17$, implies $\operatorname{nullity}\left(\mathbf{N}\right) = 1$, which means that $\lambda\left(\mathbf{K}^{-1}\mathbf{F}\left(\mathbf{D}_{\mathbf{1,2}}\right)^{-1}\mathbf{F}\left(\mathbf{T}_{\mathbf{1,2}}\right)\right)$ must have one *eigenvalue* equal to 1.

For completely random matrices $\mathbf{\Gamma}_i$, it is expected that the number of *eigenvalues* $\lambda\left(\mathbf{K}^{-1}\mathbf{F}\left(\mathbf{D}_{\mathbf{1,2}}\right)^{-1}\mathbf{F}\left(\mathbf{T}_{\mathbf{1,2}}\right)\right) \cap \lambda\left(\mathbf{I}\right) = \varnothing$. However, we intentionally chose matrix $\mathbf{D}$ as in Equation (A.19). Therefore matrix $\mathbf{D}$ has the following rows

$$\mathbf{Y} = \begin{bmatrix} \mathbf{0} & -\boldsymbol{\xi}_1^{(1)} & \boldsymbol{\xi}_1^{(2)} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{\xi}_1^{(1)} & \mathbf{0} & -\boldsymbol{\xi}_1^{(3)} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ -\boldsymbol{\xi}_1^{(2)} & \boldsymbol{\xi}_1^{(3)} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{0} & -\boldsymbol{\xi}_1^{(4)} & \boldsymbol{\xi}_1^{(5)} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{\xi}_1^{(4)} & \mathbf{0} & -\boldsymbol{\xi}_1^{(6)} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ -\boldsymbol{\xi}_1^{(5)} & \boldsymbol{\xi}_1^{(6)} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \tag{A.50}$$

where $\mathbf{Y} \in \mathbb{R}^{6\times(6P+18)}$,

$$\boldsymbol{\xi}_1^{(i)} = \begin{bmatrix} \xi_1^{(i)} & \dots & \xi_1^{(i)} & 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{1} = \begin{bmatrix} 1 & \dots & 1 & 0 & 0 & 0 \end{bmatrix} \tag{A.51}$$

with $\boldsymbol{\xi}_1^{(i)}, \mathbf{1} \in \mathbb{R}^{1\times(P+3)}$. One concludes that $\operatorname{rank}\left(\mathbf{Y}\right) = \operatorname{rank}\left(\breve{\mathbf{Y}}\right)$ where

$$\breve{\mathbf{Y}} = \begin{bmatrix} 0 & -\xi_1^{(1)} & \xi_1^{(2)} & 1 & 0 & 0 \\ \xi_1^{(1)} & 0 & -\xi_1^{(3)} & 0 & 1 & 0 \\ -\xi_1^{(2)} & \xi_1^{(3)} & 0 & 0 & 0 & 1 \\ 0 & -\xi_1^{(4)} & \xi_1^{(5)} & 1 & 0 & 0 \\ \xi_1^{(4)} & 0 & -\xi_1^{(6)} & 0 & 1 & 0 \\ -\xi_1^{(5)} & \xi_1^{(6)} & 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.52}$$

and $\operatorname{rank}\left(\mathbf{Y}\right) = \operatorname{rank}\left(\breve{\mathbf{Y}}\right) = 5$.

Since the rows of $\mathbf{Y}$ will be the columns of $\left(\mathbf{A}^{(1)}\right)^T$, we see that by choosing the matrix $\mathbf{D}$, we ensure that rank $\left(\left(\mathbf{A}^{(1)}\right)^T\right) \leqslant 6P + 17$, which means that we have one *eigenvector* of $\lambda\left(\mathbf{K}^{-1}\mathbf{F}\left(\mathbf{D}_{\mathbf{1,2}}\right)^{-1}\mathbf{F}\left(\mathbf{T}_{\mathbf{1,2}}\right)\right)$ equal to 1.

Thus, for random elements of the diagonal matrices $\mathbf{D}_i, \mathbf{T}_i, \mathbf{S}_i$, we have nullity $(\mathbf{N}) = 1$ with probability one, which implies rank $\left(\mathbf{A}^{(1)}\right) =$ rank $\left(\left(\mathbf{A}^{(1)}\right)^T\right) =$ rank $(\mathbf{N}) = 6P + 17$.

On the other hand when we are using matrices $\mathbf{D}_i, \mathbf{T}_i, \mathbf{S}_i$ whose elements are the coordinates of points that belong to a single plane or to a $3D$ line, matrix $\mathbf{K}^{-1}\mathbf{F}\left(\mathbf{D}_{\mathbf{1,2}}\right)^{-1}\mathbf{F}\left(\mathbf{T}_{\mathbf{1,2}}\right)$ (from Equation (A.49)) will have more than one eigenvalue equal to one, regardless of the matrix $\mathbf{L}$. This will imply nullity $(\mathbf{N}) > 1$ and, as a result, rank $(\mathbf{N}) < 6P + 17$.

A.4 Conclusions

To obtain the *rank* of the matrix $\mathbf{M}$ we write

$$\mathbf{M} = \mathbf{Z}^{(2)}\mathbf{A}^{(2)} \tag{A.53}$$

where the matrix $\mathbf{A}^{(2)}$ is as

$$\mathbf{A}^{(2)} = \begin{pmatrix} \mathbf{A}^{(1)} \\ \mathbf{F} \end{pmatrix} \tag{A.54}$$

and $\mathbf{A}^{(1)}$ is as in Equation (A.21) and $\mathbf{Z}^{(2)}$ is a permutation matrix.

In Section A.3, we saw that each of the rows of $\mathbf{F}$ is linearly dependent on the rows of $\mathbf{E}$, which are included in matrix $\mathbf{A}^{(1)}$. Thus, we can write rank $\left(\mathbf{A}^{(2)}\right) =$ rank $\left(\mathbf{A}^{(1)}\right) = 6P + 17$. To conclude and since the permutation of rows does not change the *rank* of a matrix, we can write rank $(\mathbf{M}) =$ rank $\left(\mathbf{A}^{(2)}\right) = 6P + 17$.

A.5 Some Matrix Results

A.5.1 Rank of $\mathbf{D}_1 - \mathbf{L}\mathbf{D}_2\mathbf{L}^{-1}$

Considering diagonal *full–rank* matrices $\mathbf{D}_1, \mathbf{D}_2 \in \mathbb{R}^{P\times P}$ and a generic *full–rank* $\mathbf{L} \in \mathbb{R}^{P\times P}$. If we write a matrix $\mathbf{M} \in \mathbb{R}^{2P\times 2P}$ as

$$\mathbf{M} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{L}\mathbf{D}_1\mathbf{L}^{-1} & -\mathbf{L}\mathbf{D}_1\mathbf{L}^{-1} + \mathbf{D}_2 \end{bmatrix}, \tag{A.55}$$

we see that $\text{rank}(\mathbf{M}) = \text{rank}(\mathbf{I}) + \text{rank}\left(-\mathbf{L}\mathbf{D}_1\mathbf{L}^{-1} + \mathbf{D}_2\right)$. If we post–multiply $\mathbf{M}$ by any *non–singular* matrix, the *rank* of the resulting matrix will be the same as the *rank* of $\mathbf{M}$. As a result, we define

$$\mathbf{N} = \mathbf{M}\begin{bmatrix} \mathbf{I} & \mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{A.56}$$

where $\text{rank}(\mathbf{N}) = \text{rank}(\mathbf{M})$ and

$$\mathbf{N} = \begin{bmatrix} \mathbf{I} & \mathbf{I} \\ \mathbf{L}\mathbf{D}_1\mathbf{L}^{-1} & \mathbf{D}_2 \end{bmatrix}. \tag{A.57}$$

We can see that the *null–space* of $\mathbf{N}$ must satisfy

$$\begin{bmatrix} \mathbf{I} & \mathbf{I} \\ \mathbf{L}\mathbf{D}_1\mathbf{L}^{-1} & \mathbf{D}_2 \end{bmatrix}\begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix} = \mathbf{0}, \tag{A.58}$$

which can be rewritten as

$$\begin{cases} \mathbf{v}_1 = -\mathbf{v}_2 \\ -\mathbf{D}_2^{-1}\mathbf{L}\mathbf{D}_1\mathbf{L}^{-1}\mathbf{v}_1 = \mathbf{v}_2 \end{cases} \tag{A.59}$$

and $\text{nullity}(\mathbf{N}) = n$, where n is the number of *eigenvalues* of $-\mathbf{D}_2^{-1}\mathbf{L}\mathbf{D}_1\mathbf{L}^{-1}$ equal to one.

If there are no *eigenvalues* equal to one, then $\text{nullity}(\mathbf{N}) = 0$, which implies $\text{nullity}(\mathbf{M}) = \text{nullity}(\mathbf{N}) = 2P$ and $\text{rank}\left(-\mathbf{L}\mathbf{D}_1\mathbf{L}^{-1} + \mathbf{D}_2\right) = P$.

A.5.2 Inverse of Matrices

In this section we describe how to get the inverses of the matrix

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} & \mathbf{A}_2 \\ \mathbf{0} & \mathbf{A}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_4 \end{bmatrix} \tag{A.60}$$

where $\mathbf{A}$ is *full–rank*. The inverse must satisfy $\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$, thus

$$\begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \mathbf{X}_3 \\ \mathbf{X}_4 & \mathbf{X}_5 & \mathbf{X}_6 \\ \mathbf{X}_7 & \mathbf{X}_8 & \mathbf{X}_9 \end{bmatrix}\begin{bmatrix} \mathbf{A}_1 & \mathbf{0} & \mathbf{A}_2 \\ \mathbf{0} & \mathbf{A}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_4 \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}. \tag{A.61}$$

We define the three next systems

$$\left\{\begin{array}{l}\mathbf{X}_1\mathbf{A}_1=\mathbf{I}\\ \mathbf{X}_4\mathbf{A}_1=\mathbf{0}\\ \mathbf{X}_7\mathbf{A}_1=\mathbf{0}\end{array}\right., \quad \left\{\begin{array}{l}\mathbf{X}_2\mathbf{A}_3=\mathbf{0}\\ \mathbf{X}_5\mathbf{A}_3=\mathbf{I}\\ \mathbf{X}_8\mathbf{A}_3=\mathbf{0}\end{array}\right. \quad \text{and} \quad \left\{\begin{array}{l}\mathbf{X}_1\mathbf{A}_2+\mathbf{X}_3\mathbf{A}_4=\mathbf{0}\\ \mathbf{X}_4\mathbf{A}_2+\mathbf{X}_6\mathbf{A}_4=\mathbf{0}\\ \mathbf{X}_7\mathbf{A}_2+\mathbf{X}_9\mathbf{A}_4=\mathbf{I}\end{array}\right. . \tag{A.62}$$

From the first system and since $\mathbf{A}_1$ must be *full-rank*, we get $\mathbf{X}_7=\mathbf{X}_4=\mathbf{0}$ and $\mathbf{X}_1=\mathbf{A}_1^{-1}$. From the second system, we get $\mathbf{X}_2=\mathbf{X}_8=\mathbf{0}$ and $\mathbf{X}_5=\mathbf{A}_3^{-1}$. Since $\mathbf{X}_4=\mathbf{X}_7=\mathbf{0}$, we can rewrite the third system as

$$\left\{\begin{array}{l}\mathbf{X}_1\mathbf{A}_2+\mathbf{X}_3\mathbf{A}_4=\mathbf{0} \implies \mathbf{X}_3=-\mathbf{A}_1^{-1}\mathbf{A}_2\mathbf{A}_4^{-1}\\ \mathbf{X}_6\mathbf{A}_4=\mathbf{0}\\ \mathbf{X}_9\mathbf{A}_4=\mathbf{I}\end{array}\right. \tag{A.63}$$

and we can write $\mathbf{X}_6=\mathbf{0}$ and $\mathbf{X}_9=\mathbf{A}_4^{-1}$. Putting all together we write

$$\mathbf{A}^{-1}=\begin{bmatrix}\mathbf{A}_1^{-1} & \mathbf{0} & -\mathbf{A}_1^{-1}\mathbf{A}_2\mathbf{A}_4^{-1}\\ \mathbf{0} & \mathbf{A}_3^{-1} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{A}_4^{-1}\end{bmatrix}. \tag{A.64}$$

Using the same method, we can prove that

$$\mathbf{B}^{-1}=\begin{bmatrix}\mathbf{B}_1 & \mathbf{B}_2 & \mathbf{0}\\ \mathbf{0} & \mathbf{B}_3 & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{B}_4\end{bmatrix}^{-1}=\begin{bmatrix}\mathbf{B}_1^{-1} & -\mathbf{B}_1^{-1}\mathbf{B}_2\mathbf{B}_3^{-1} & \mathbf{0}\\ \mathbf{0} & \mathbf{B}_3^{-1} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{B}_4^{-1}\end{bmatrix}. \tag{A.65}$$

A.5.3 Eigenvector Matrices

Suppose we want to know the structure of the *eigenvector matrix* (let us call it $\mathbf{V}_\mathbf{A}$) of a matrix

$$\mathbf{A}=\begin{bmatrix}\mathbf{A}_1 & \mathbf{0} & \mathbf{A}_2\\ \mathbf{0} & -\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & -\mathbf{I}\end{bmatrix} \tag{A.66}$$

where $\mathbf{A}$ is *full–rank*. We know that $\mathbf{V}_\mathbf{A}$ must satisfy $\mathbf{A}\mathbf{V}_\mathbf{A}=\mathbf{V}_\mathbf{A}\mathbf{\Sigma}_\mathbf{A}$, where $\mathbf{\Sigma}_\mathbf{A}$ is a diagonal matrix whose diagonal elements are $\lambda(\mathbf{A})$. $\lambda(\mathbf{A})$ represents the set of all *eigenvalues* of the matrix $\mathbf{A}$.

It is well known from linear algebra that the *eigenvalues* of a block triangular

matrix are the union of of the *eigenvalues* of the diagonal submatrices, such that

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_1 & * & \dots & * \\ \mathbf{0} & \mathbf{K}_2 & \dots & * \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{K}_N \end{bmatrix} \tag{A.67}$$

implies $\lambda(\mathbf{K}) = \lambda(\mathbf{K}_1) \cup \lambda(\mathbf{K}_2) \cup \cdots \cup \lambda(\mathbf{K}_N)$, for matrices $\mathbf{K}_i$ with any dimension.

As a result, from Equation (A.66), we have $\lambda(\mathbf{A}) = \lambda(\mathbf{A}_1) \cup \lambda(-\mathbf{I}) \cup \lambda(-\mathbf{I})$. Thus, we write

$$\begin{bmatrix} \mathbf{A}_1 & \mathbf{0} & \mathbf{A}_2 \\ \mathbf{0} & -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \mathbf{X}_3 \\ \mathbf{X}_4 & \mathbf{X}_5 & \mathbf{X}_6 \\ \mathbf{X}_7 & \mathbf{X}_8 & \mathbf{X}_9 \end{bmatrix} = \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \mathbf{X}_3 \\ \mathbf{X}_4 & \mathbf{X}_5 & \mathbf{X}_6 \\ \mathbf{X}_7 & \mathbf{X}_8 & \mathbf{X}_9 \end{bmatrix} \begin{bmatrix} \mathbf{\Sigma}_{\mathbf{A}_1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_{-\mathbf{I}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{\Sigma}_{-\mathbf{I}} \end{bmatrix}. \tag{A.68}$$

Using this representation we define the system

$$\begin{cases} \mathbf{A}_1\mathbf{X}_1 + \mathbf{A}_2\mathbf{X}_7 = \mathbf{X}_1\mathbf{\Sigma}_{\mathbf{A}_1} \\ -\mathbf{X}_4 = \mathbf{X}_4\mathbf{\Sigma}_{\mathbf{A}_1} \\ -\mathbf{X}_7 = \mathbf{X}_7\mathbf{\Sigma}_{\mathbf{A}_1} \end{cases}. \tag{A.69}$$

If we consider that matrix $\mathbf{A}_1$ is a random matrix, the probability of $\lambda(\mathbf{A}_1) \cap \lambda(-\mathbf{I}) = \varnothing$ is equal to one, which from Equation (A.69) implies that $\mathbf{X}_4 = \mathbf{X}_7 = \mathbf{0}$ and as a result $\mathbf{X}_1 = \mathbf{V}_{\mathbf{A}_1}$, where $\mathbf{V}_{\mathbf{A}_1}$ is the *eigenvector matrix* of $\mathbf{A}_1$.

The remaining equations from Equation (A.68) must verify

$$\begin{cases} \mathbf{A}_1\mathbf{X}_2 + \mathbf{A}_2\mathbf{X}_8 = -\mathbf{X}_2 \\ -\mathbf{X}_5 = -\mathbf{X}_5 \\ -\mathbf{X}_8 = -\mathbf{X}_8 \end{cases} \quad \text{and} \quad \begin{cases} \mathbf{A}_1\mathbf{X}_3 + \mathbf{A}_2\mathbf{X}_9 = -\mathbf{X}_3 \\ -\mathbf{X}_6 = -\mathbf{X}_6 \\ -\mathbf{X}_9 = -\mathbf{X}_9 \end{cases}. \tag{A.70}$$

We are interested in the subspace of *eigenvectors*. Thus, we can define a set of *eigenvector* basis where $\mathbf{X}_8 = \mathbf{X}_6 = \mathbf{0}$ and $\mathbf{X}_5 = \mathbf{X}_9 = \mathbf{I}$. Developing the system of equations, we get

$$\mathbf{V}_{\mathbf{A}} = \begin{bmatrix} \mathbf{V}_{\mathbf{A}_1} & \mathbf{0} & (-\mathbf{I} - \mathbf{A}_1)^{-1}\mathbf{A}_2 \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}. \tag{A.71}$$

If we apply the same method to the matrix

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_1 & \mathbf{B}_2 & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{I} \end{bmatrix} \tag{A.72}$$

and considering that $\mathbf{B}_1$ is a random matrix ($\lambda(\mathbf{B}_1) \cap \lambda(-\mathbf{I}) = \varnothing$), we get

$$\mathbf{V_B} = \begin{bmatrix} \mathbf{V}_{\mathbf{B}_1} & (-\mathbf{I} - \mathbf{B}_1)^{-1}\mathbf{B}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}. \tag{A.73}$$

A.5.4 Intersection Subspace

In this section, we study the intersection subspace between *eigenvector matrices* $\mathbf{V_A}$ of Equation (A.71) and $\mathbf{V_B}$ of Equation (A.73), that correspond to *eigenvalues* equal to minus one.

Since we are only interested in the *eigenvectors* that correspond to *eigenvalues* equal to minus one, from Appendix A.5.3, we can define

$$\breve{\mathbf{V}}_{\mathbf{A}} = \begin{bmatrix} \mathbf{0} & (-\mathbf{I} - \mathbf{A}_1)^{-1}\mathbf{A}_2 \\ \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \quad \text{and} \quad \breve{\mathbf{V}}_{\mathbf{B}} = \begin{bmatrix} (-\mathbf{I} - \mathbf{B}_1)^{-1}\mathbf{B}_2 & \mathbf{0} \\ \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \tag{A.74}$$

and the basis for the intersection subspace can be obtained from the solution of the following Equation

$$\underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{A}_3^{-1}\mathbf{A}_2 & \mathbf{B}_3^{-1}\mathbf{B}_2 \\ \mathbf{I} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \end{bmatrix}}_{\mathbf{M}} \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \mathbf{X}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{X}_4 \end{bmatrix} \tag{A.75}$$

where $\mathbf{M}$ is *full-rank*, $\mathbf{A}_3 = -\mathbf{I} - \mathbf{A}_1$ and $\mathbf{B}_3 = -\mathbf{I} - \mathbf{B}_1$. Note that $\mathbf{A}_1$ and $\mathbf{B}_1$ are random matrices which means that the probability of $\lambda(\mathbf{A}_1) \cap \lambda(\mathbf{I}) = \varnothing$ and $\lambda(\mathbf{B}_1) \cap \lambda(\mathbf{I}) = \varnothing$ is one and, from Appendix A.5.1, we know that $\mathbf{A}_3$ and $\mathbf{B}_3$ have inverses.

The subspace of solution for Equation (A.75) can be defined as

$$\begin{cases} \mathbf{X}_1 = -\mathbf{K} \\ \mathbf{X}_3 = \mathbf{K} \\ \mathbf{X}_2 = -\mathbf{A}_2^{-1}\mathbf{A}_3\mathbf{B}_3^{-1}\mathbf{B}_2\mathbf{K} \\ \mathbf{X}_4 = -\mathbf{A}_2^{-1}\mathbf{A}_3\mathbf{B}_3^{-1}\mathbf{B}_2\mathbf{K} \end{cases}, \tag{A.76}$$

for any *non–singular* matrix $\mathbf{K}$.

We are interested in defining the basis for the intersection subspace. Thus, we can write $\mathbf{K} = \mathbf{I}$ and

$$\begin{cases} \mathbf{X}_1 = -\mathbf{I} \\ \mathbf{X}_3 = \mathbf{I} \\ \mathbf{X}_2 = -\mathbf{A}_2^{-1}\mathbf{A}_3\mathbf{B}_3^{-1}\mathbf{B}_2 \\ \mathbf{X}_4 = -\mathbf{A}_2^{-1}\mathbf{A}_3\mathbf{B}_3^{-1}\mathbf{B}_2 \end{cases} . \tag{A.77}$$

Using $\mathbf{X}_1$ and $\mathbf{X}_2$ we can determine the intersection subspace from

$$\begin{bmatrix} \mathbf{0} & \mathbf{A}_3^{-1}\mathbf{A}_2 \\ \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \end{bmatrix} = \begin{bmatrix} -\mathbf{B}_3^{-1}\mathbf{B}_2 \\ -\mathbf{I} \\ -\mathbf{A}_2^{-1}\mathbf{A}_3\mathbf{B}_3^{-1}\mathbf{B}_2 \end{bmatrix} . \tag{A.78}$$

Appendix B

Analytical Solution for the Null-Space of Both N and M Matrices

For the computation of the null-space of the matrix $\mathbf{N}$, let us first consider the null $(\mathbf{M})$.

From the definition of the *Kronecker* product – Section 3.1, we can rewrite the matrix $\mathbf{M}$ (from Equation (9.20)) as

$$\mathbf{M} = \begin{bmatrix} p_{1,1}^{(\mathcal{W})}\widehat{\mathbf{d}}_1^{(\mathcal{C})} & p_{1,2}^{(\mathcal{W})}\widehat{\mathbf{d}}_1^{(\mathcal{C})} & p_{1,3}^{(\mathcal{W})}\widehat{\mathbf{d}}_1^{(\mathcal{C})} \\ p_{2,1}^{(\mathcal{W})}\widehat{\mathbf{d}}_2^{(\mathcal{C})} & p_{2,2}^{(\mathcal{W})}\widehat{\mathbf{d}}_2^{(\mathcal{C})} & p_{2,3}^{(\mathcal{W})}\widehat{\mathbf{d}}_2^{(\mathcal{C})} \\ p_{3,1}^{(\mathcal{W})}\widehat{\mathbf{d}}_3^{(\mathcal{C})} & p_{3,2}^{(\mathcal{W})}\widehat{\mathbf{d}}_3^{(\mathcal{C})} & p_{3,3}^{(\mathcal{W})}\widehat{\mathbf{d}}_3^{(\mathcal{C})} \end{bmatrix} \tag{B.1}$$

where $p_{i,j}^{(\mathcal{W})}$ is the j^{th} element of the vector $\mathbf{p}_i^{(\mathcal{W})}$.

To compute the basis for the null space, we use the fact that $\mathbf{d}_i^{(\mathcal{C})} \times \mathbf{d}_i^{(\mathcal{C})} = \mathbf{0}$ and define a vector $\mathbf{q}_i^{(\mathcal{W})}$ orthogonal to both $\mathbf{p}_j^{(\mathcal{W})}$ and $\mathbf{p}_k^{(\mathcal{W})}$ such that

$$\mathbf{q}_1^{(\mathcal{W})} = \mathbf{p}_2^{(\mathcal{W})} \times \mathbf{p}_3^{(\mathcal{W})}, \;\; \mathbf{q}_2^{(\mathcal{W})} = \mathbf{p}_1^{(\mathcal{W})} \times \mathbf{p}_3^{(\mathcal{W})} \;\text{ and }\; \mathbf{q}_3^{(\mathcal{W})} = \mathbf{p}_1^{(\mathcal{W})} \times \mathbf{p}_2^{(\mathcal{W})}, \tag{B.2}$$

From the Theorem 9.2, the null-space of $\mathbf{M}$ has dimension three. As a result, we define the three basis for null $(\mathbf{M})$ as

$$\breve{\mathbf{f}}_1 = \begin{bmatrix} q_{1,1}^{(\mathcal{W})}\mathbf{d}_1^{(\mathcal{C})}, & q_{1,2}^{(\mathcal{W})}\mathbf{d}_1^{(\mathcal{C})}, & q_{1,3}^{(\mathcal{W})}\mathbf{d}_1^{(\mathcal{C})} \end{bmatrix} \tag{B.3}$$

$$\breve{\mathbf{f}}_2 = \begin{bmatrix} q_{2,1}^{(\mathcal{W})}\mathbf{d}_2^{(\mathcal{C})}, & q_{2,2}^{(\mathcal{W})}\mathbf{d}_2^{(\mathcal{C})}, & q_{2,3}^{(\mathcal{W})}\mathbf{d}_2^{(\mathcal{C})} \end{bmatrix} \tag{B.4}$$

$$\breve{\mathbf{f}}_3 = \begin{bmatrix} q_{3,1}^{(\mathcal{W})}\mathbf{d}_3^{(\mathcal{C})}, & q_{3,2}^{(\mathcal{W})}\mathbf{d}_3^{(\mathcal{C})}, & q_{3,3}^{(\mathcal{W})}\mathbf{d}_3^{(\mathcal{C})} \end{bmatrix} \tag{B.5}$$

where $q_{i,j}^{(\mathcal{W})}$ is the j^{th} element of the vector $\mathbf{q}_i^{(\mathcal{W})}$.

Let us now consider the matrix $\mathbf{N}$ as described in Equation (9.21). Considering vectors $\breve{\mathbf{f}}_i$ from Equations (B.3-B.5) as the basis for the null-space of the matrix $\mathbf{M}$, and from definition of matrix $\mathbf{N}$, we can define the basis $\mathbf{f}_i = \left[\breve{\mathbf{f}}_i, 0\right]$ for $i = 1, 2, 3$, where $\mathbf{f}_i$ are three basis for the null-space of $\mathbf{N}$. However, since the dimension of the null-space of $\mathbf{N}$ is four, there is one basis left.

From geometric properties, we know that $\mathbf{m}_i^{(\mathcal{C})} = \mathbf{x}_i^{(\mathcal{C})} \times \mathbf{d}_i^{(\mathcal{C})}$, for any point $\mathbf{x}_i^{(\mathcal{C})}$ that belong to the line. As a result and considering the vector $\mathbf{q}_i^{(\mathcal{W})}$ defined in Equation (B.2), we derived the following vectors

$$\bar{\mathbf{f}}_4^{(1)} = \left[\begin{array}{c} \breve{q}_{1,1}^{(\mathcal{W})}\mathbf{x}_1^{(\mathcal{C})}, \breve{q}_{1,2}^{(\mathcal{W})}\mathbf{x}_1^{(\mathcal{C})}, \breve{q}_{1,3}^{(\mathcal{W})}\mathbf{x}_1^{(\mathcal{C})}, 1 \end{array}\right] \tag{B.6}$$

$$\bar{\mathbf{f}}_4^{(2)} = \left[\begin{array}{c} \breve{q}_{2,1}^{(\mathcal{W})}\mathbf{x}_2^{(\mathcal{C})}, \breve{q}_{2,2}^{(\mathcal{W})}\mathbf{x}_2^{(\mathcal{C})}, \breve{q}_{2,3}^{(\mathcal{W})}\mathbf{x}_2^{(\mathcal{C})}, 1 \end{array}\right] \tag{B.7}$$

$$\bar{\mathbf{f}}_4^{(3)} = \left[\begin{array}{c} \breve{q}_{3,1}^{(\mathcal{W})}\mathbf{x}_3^{(\mathcal{C})}, \breve{q}_{3,2}^{(\mathcal{W})}\mathbf{x}_3^{(\mathcal{C})}, \breve{q}_{3,3}^{(\mathcal{W})}\mathbf{x}_3^{(\mathcal{C})}, 1 \end{array}\right] \tag{B.8}$$

where

$$\breve{q}_{i,j}^{(\mathcal{W})} = \frac{q_{i,j}^{(\mathcal{W})}}{q_{i,1}^{(\mathcal{W})}\breve{p}_{i,1}^{(\mathcal{W})} + q_{i,2}^{(\mathcal{W})}\breve{p}_{i,2}^{(\mathcal{W})} + q_{i,3}^{(\mathcal{W})}\breve{p}_{i,3}^{(\mathcal{W})}}, \tag{B.9}$$

such that

$$\mathbf{N}\bar{\mathbf{f}}_4^{(1)} = \begin{bmatrix} \mathbf{0} \\ -\mathbf{m}_2^{(\mathcal{C})} \\ -\mathbf{m}_3^{(\mathcal{C})} \end{bmatrix}, \; \mathbf{N}\bar{\mathbf{f}}_4^{(2)} = \begin{bmatrix} -\mathbf{m}_1^{(\mathcal{C})} \\ \mathbf{0} \\ -\mathbf{m}_3^{(\mathcal{C})} \end{bmatrix}, \text{ and } \mathbf{N}\bar{\mathbf{f}}_4^{(3)} = \begin{bmatrix} -\mathbf{m}_1^{(\mathcal{C})} \\ -\mathbf{m}_2^{(\mathcal{C})} \\ \mathbf{0} \end{bmatrix}. \tag{B.10}$$

From this result and considering the last column of the matrix $\mathbf{N}$ – Equations (9.21) and (9.20), we define the last base for the null-space as $\mathbf{N}\mathbf{f}_4 = \mathbf{0}$, such that

$$\mathbf{f}_4 = \bar{\mathbf{f}}_4^{(1)} + \bar{\mathbf{f}}_4^{(2)} + \bar{\mathbf{f}}_4^{(3)} + [0, \ldots, 0, -2]^T. \tag{B.11}$$

Note that the tenth element of $\mathbf{f}_1$, $\mathbf{f}_2$ $\mathbf{f}_3$ and $\mathbf{f}_4$ are 0, 0, 0 and 1. As a result, the null-space for the matrix $\mathbf{N}$ such that $\xi_{10} = 1$, corresponds to Equation (9.23).

Appendix C

Minimal Solvers

C.1 Solver for the General Case

In the algorithms proposed in thesis, the solutions for the pose problem are obtained as a result of estimating the intersection points of three quadrics. From *Bézout*'s theorem [CLO04], we can conclude that we have up to eight solutions for the intersection points $(\alpha_1, \alpha_2, \alpha_3)$. According to Nister *et al.* at [NHS07], this problem requires a solution of a single variable polynomial equation with degree no less than eight. As shown by Guo at [Guo12], it is possible to derive an eight degree polynomial equation for the problem of Equation (9.29). It is also possible to use conventional Methods such as *Gröebner basis* [Ste05, KBP08] or *hidden variable technique* [HL12]. In this section we used a *polynomial eigenvalue* approach (Kúkelová *et al.* at [KBP12b]) and derived a simple and very intuitive solution for the intersection of the quadrics. We use this approach on the experiments. Note that the aim of the thesis is not to develop a minimal solver to the probem but a new problem formulatiom.

In this section we briefly describe the application of the *eigen decomposition* method to compute the intersection points between the three quadrics. Let us consider the system of three equations represented in Equation (9.29). If we choose α_1 as the *eigenvalue*, the *polynomial eigenvalue* as the form

$$\left(\alpha_1^2\mathbf{C}_2 + \alpha_1\mathbf{C}_1 + \mathbf{C}_0\right)\mathbf{v} = \mathbf{0}. \tag{C.1}$$

Polynomial eigenvalue solvers require that the matrices $\mathbf{C}_i$ have square shapes. Note that since the original problem has only three equations and the size of the vector $\mathbf{v}$ is six, we can not get square matrices $\mathbf{C}_i$. We have to generate new equations as a polynomial combination of the original ones (*resultant–based methods*), to ensure that the coefficient matrices $\mathbf{C}_i$ have square shape. We generate new constraints multiplying the constraints $g_{i,j}\left(\alpha_1, \alpha_2, \alpha_3\right) = 0$ by the monomials $\left\{\alpha_2^2, \alpha_3^2, \alpha_2\alpha_3, \alpha_2, \alpha_3, 1\right\}$.

After the generation of the new constraints, we get $\mathbf{v} \in \mathbb{R}^{15}$ such that

$$\mathbf{v} = \left[\alpha_2^4, \alpha_2^3\alpha_3, \alpha_2\alpha_3^3, \alpha_2^2\alpha_3^2, \alpha_3^4, \alpha_2^3, \alpha_2^2\alpha_3, \alpha_2\alpha_3^2, \alpha_3^3, \alpha_2^2, \alpha_2\alpha_3, \alpha_3^2, \alpha_2, \alpha_3, 1\right] \tag{C.2}$$

and, from the eighteen constraints, we can find fifteen that makes $\mathbf{C}_0, \mathbf{C}_1, \mathbf{C}_2 \in \mathbb{R}^{15\times15}$ such that matrix $\mathbf{C}_0$ will be non–singular. As a result, the *polynomial eigenvalue* problem of Equation (C.1) can be rewritten as a standard *eigen decomposition* problem $\mathbf{C}\breve{\mathbf{v}} = \breve{\alpha}_1\breve{\mathbf{v}}$, where $\breve{\alpha}_1 = \alpha_1^{-1}$ and matrix $\mathbf{C} \in \mathbb{R}^{30\times30}$ and vector $\breve{\mathbf{v}} \in \mathbb{R}^{30}$ are

$$\mathbf{C} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{C}_0^{-1}\mathbf{C}_2 & -\mathbf{C}_0^{-1}\mathbf{C}_1 \end{bmatrix}, \quad \breve{\mathbf{v}} = \begin{bmatrix} \mathbf{v} \\ \breve{\alpha}_1\mathbf{v} \end{bmatrix}. \tag{C.3}$$

To conclude, the solution for the unknowns $(\alpha_1, \alpha_2, \alpha_3)$ that verifies Equation (9.29) can be given by the *eigen decomposition* of matrix $\mathbf{C}$ where: α_1 is given by the inverse of the eigenvalue of $\mathbf{C}$ and the corresponding α_2 and α_3 are given by the thirteenth and fourteenth element of the corresponding eigenvector.

Note that from the size of matrix $\mathbf{C}$ there will be thirty *eigenvalues*. However, we can see that there will be sixteen columns of matrices $-\mathbf{C}_0^{-1}\mathbf{C}_2$ and $-\mathbf{C}_0^{-1}\mathbf{C}_1$ equal to zero. These columns will generate zero *eigenvalues*, which can not be used since the value of α_1 will be the inverse of each *eigenvalue*. Thus, we can be eliminate these columns, as well as the corresponding rows, resulting in an 14×14 matrix. Additional zero and complex *eigenvalues* are also generated and have to be eliminated, which will result in up to eight real solutions.

C.2 Solver for Central Camera Models

In Section 9.2.1, we derived an algorithm for the minimal pose problem for central cameras. The algorithm requires the estimation of the intersection points between three special quadrics. In this section we describe the closed-form method that we used to solve this problem.

Let us consider the problem defined by Equations (9.35) and (9.34). We can see that the three constraints $g_{i,j}\left(\widetilde{\alpha}_1, \widetilde{\alpha}_2, \widetilde{\alpha}_3\right) = 0$ have only one monomial with the unknown $\widetilde{\alpha}_3$, which is the monomial $\widetilde{\alpha}_3^2$.

We use the equation $g_{2,3}\left(\widetilde{\alpha}_1, \widetilde{\alpha}_2, \widetilde{\alpha}_3\right)$ to find the solution for $\widetilde{\alpha}_3$, such that

$$\widetilde{\alpha}_3 = \pm\Phi_1[\widetilde{\alpha}_1, \widetilde{\alpha}_2]^{1/2} \tag{C.4}$$

where $\Phi_1[\widetilde{\alpha}_1, \widetilde{\alpha}_2]$ represents a polynomial equation with unknowns $\widetilde{\alpha}_1$ and $\widetilde{\alpha}_2$ with degree two.

Now, we can substitute $\widetilde{\alpha}_3^2$ on constraints $g_{1,2}\left(\widetilde{\alpha}_1, \widetilde{\alpha}_2, \widetilde{\alpha}_3\right)$ and $g_{1,3}\left(\widetilde{\alpha}_1, \widetilde{\alpha}_2, \widetilde{\alpha}_3\right)$

using Equation (C.4). Thus, we derive two constraints of the form

$$\widetilde{g}_{1,2}\left(\widetilde{\alpha}_1,\widetilde{\alpha}_2\right)=\Phi_2[\widetilde{\alpha}_1,\widetilde{\alpha}_2] \text{ and } \widetilde{g}_{1,3}\left(\widetilde{\alpha}_1,\widetilde{\alpha}_2\right)=\Phi_3[\widetilde{\alpha}_1,\widetilde{\alpha}_2] \tag{C.5}$$

where both $\Phi_2[\widetilde{\alpha}_1,\widetilde{\alpha}_2]$ and $\Phi_3[\widetilde{\alpha}_1,\widetilde{\alpha}_2]$ have degree two.

Let us consider the constraint $\widetilde{g}_{1,3}\left(\widetilde{\alpha}_1,\widetilde{\alpha}_2\right)=0$. If we solve in terms of $\widetilde{\alpha}_2$, we get

$$\widetilde{\alpha}_2=\phi\left(\Phi_4[\widetilde{\alpha}_1]\pm\Gamma[\widetilde{\alpha}_1]^{1/2}\right) \tag{C.6}$$

where: ϕ is a constant; and $\Phi_4[\widetilde{\alpha}_1]$ and $\Gamma[\widetilde{\alpha}_1]$ are polynomial equations with degree one and two respectively.

To conclude, replacing $\widetilde{\alpha}_2$ defined at Equation (C.6) on the first constraint $\widetilde{g}_{1,2}\left(\widetilde{\alpha}_1,\widetilde{\alpha}_2\right)=0$, we derive the fllowing equations

$$\Phi_5[\widetilde{\alpha}_1]\pm\Phi_6[\widetilde{\alpha}_1]\Gamma[\widetilde{\alpha}_1]^{1/2}=0\Rightarrow\Phi_5[\widetilde{\alpha}_1]=\mp\Phi_6[\widetilde{\alpha}_1]\Gamma[\widetilde{\alpha}_1]^{1/2}. \tag{C.7}$$

The degrees of the polynomial equations $\Phi_5[\widetilde{\alpha}_1]$ and $\Phi_6[\widetilde{\alpha}_1]$ are respectively two and one. Squaring both sides of the Equation (C.7) we get

$$\Phi_5[\widetilde{\alpha}_1]^2=\Phi_6[\widetilde{\alpha}_1]^2\Gamma[\widetilde{\alpha}_1]\Rightarrow\Phi_7[\widetilde{\alpha}_1]=\Phi_5[\widetilde{\alpha}_1]^2-\Phi_6[\widetilde{\alpha}_1]^2\Gamma[\widetilde{\alpha}_1]=0 \tag{C.8}$$

where polynomial equation $\Phi_7[\widetilde{\alpha}_1]$ has degree four.

Thus, to find the solutions for the unknown $\widetilde{\alpha}_1$ that solve the problem defined in Equation (9.35), we just need to find the roots of the four degree polynomial equation $\Phi_7[\widetilde{\alpha}_1]$, which can be solved in closed-form (*eg.* using the Ferrari's technique for solving the general quartic roots). For each real solution for $\widetilde{\alpha}_1$ we get $\widetilde{\alpha}_2$ using Equation (C.6) – from the two possible solutions, we choose the one that verifies $\widetilde{g}_{1,2}\left(\widetilde{\alpha}_1,\widetilde{\alpha}_2\right)=\widetilde{g}_{1,3}\left(\widetilde{\alpha}_1,\widetilde{\alpha}_2\right)=0$. For the set of possible solutions for $\widetilde{\alpha}_1$ and $\widetilde{\alpha}_2$ compute $\widetilde{\alpha}_3$, using Equation (C.4). Note that for each set of $\widetilde{\alpha}_1$ and $\widetilde{\alpha}_2$ we have two possible solution for $\widetilde{\alpha}_3$, which means that we will have up to eight solutions for $\{\widetilde{\alpha}_1,\widetilde{\alpha}_2,\widetilde{\alpha}_3\}$.

Appendix D

Proof of Theorems and Propositions

In Sections 9.1 and 9.2, some theorems and proposition were presented without *proofs*. In this appendix we derive these *proofs*.

D.1 *proof* of the Theorem 9.1

The *proof* for this theorem is divided into three lemmas.

Lemma D.1. *For a given homography matrix,* $\mathbf{H} \in \mathcal{H}$ *(Proposition 9.1), and a known plane with coordinates* $\mathbf{\Pi}^{(\mathcal{W})}\mathbb{R} \doteq \left(-\zeta^{(\mathcal{W})}, \boldsymbol{\pi}^{(\mathcal{W})}\right) \in \mathbb{R}^4$ *(where, without loss of generality,* $\left|\boldsymbol{\pi}^{(\mathcal{W})}\right| = 1$*), if the singular values of* $\mathbf{H}$ *verify* $\sigma_1^2 > 1 > \sigma_3^2$*, its decomposition into a rotation and a translation,* $\mathbf{R} \in \mathcal{SO}(3)$ *and* $\mathbf{t} \in \mathbb{R}^3$*, has a unique solution.*

Proof. Let us consider the decomposition of the *homography* described in Equations (9.12) and (9.13), Section 9.1.

Since the plane coordinates $\zeta^{(\mathcal{W})}$ and $\boldsymbol{\pi}^{(\mathcal{W})}$ are known, the estimates $\breve{\boldsymbol{\pi}}_i^{(\mathcal{W})}$ can be compared against the real value for $\boldsymbol{\pi}^{(\mathcal{W})}$ and as a result the real solution selected. Therefore, to have more than one solution for $\mathbf{R}$ and $\mathbf{t}$, $\breve{\boldsymbol{\pi}}_i^{(\mathcal{W})} = \breve{\boldsymbol{\pi}}_j^{(\mathcal{W})}$ must exist for for some $i \neq j$.

We are dealing with vectors $\breve{\boldsymbol{\pi}}_i^{(\mathcal{W})}$ whose norms are equal to one. As a result, to have $\breve{\boldsymbol{\pi}}_i^{(\mathcal{W})} = \breve{\boldsymbol{\pi}}_j^{(\mathcal{W})}$, their inner product $\breve{\boldsymbol{\pi}}_i^{(\mathcal{W})} \cdot \breve{\boldsymbol{\pi}}_j^{(\mathcal{W})}$ must be equal to one. Using the *Binet–Cauchy* identity[1] and since $\mathbf{v}_2 \cdot \mathbf{u}_1 = \mathbf{v}_2 \cdot \mathbf{u}_2 = 0$, we can define the

[1] *Binet–Cauchy* identity – $(\mathbf{a} \times \mathbf{b}) \cdot (\mathbf{c} \times \mathbf{d}) = (\mathbf{a} \cdot \mathbf{c})(\mathbf{b} \cdot \mathbf{d}) - (\mathbf{a} \cdot \mathbf{d})(\mathbf{b} \cdot \mathbf{c})$ for $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d} \in \mathbb{R}^3$

several hypothesis $\breve{\boldsymbol{\pi}}_i^{(\mathcal{W})} \cdot \breve{\boldsymbol{\pi}}_j^{(\mathcal{W})}$ as

$$\breve{\boldsymbol{\pi}}_1^{(\mathcal{W})} \cdot \breve{\boldsymbol{\pi}}_2^{(\mathcal{W})} = \mathbf{u}_1 \cdot \mathbf{u}_2 \tag{D.1}$$
$$\breve{\boldsymbol{\pi}}_1^{(\mathcal{W})} \cdot \breve{\boldsymbol{\pi}}_3^{(\mathcal{W})} = -1 \tag{D.2}$$
$$\breve{\boldsymbol{\pi}}_1^{(\mathcal{W})} \cdot \breve{\boldsymbol{\pi}}_4^{(\mathcal{W})} = -\mathbf{u}_1 \cdot \mathbf{u}_2 \tag{D.3}$$
$$\breve{\boldsymbol{\pi}}_2^{(\mathcal{W})} \cdot \breve{\boldsymbol{\pi}}_3^{(\mathcal{W})} = -\mathbf{u}_1 \cdot \mathbf{u}_2 \tag{D.4}$$
$$\breve{\boldsymbol{\pi}}_2^{(\mathcal{W})} \cdot \breve{\boldsymbol{\pi}}_4^{(\mathcal{W})} = -1 \tag{D.5}$$
$$\breve{\boldsymbol{\pi}}_3^{(\mathcal{W})} \cdot \breve{\boldsymbol{\pi}}_4^{(\mathcal{W})} = \mathbf{u}_1 \cdot \mathbf{u}_2. \tag{D.6}$$

The vectors $\mathbf{u}_1$ and $\mathbf{u}_2$ were derived in Equations (9.6) and (9.7).

From Equations (D.1-D.6), we can eliminate the hypotheses $\breve{\boldsymbol{\pi}}_1^{(\mathcal{W})} \cdot \breve{\boldsymbol{\pi}}_3^{(\mathcal{W})}$ and $\breve{\boldsymbol{\pi}}_2^{(\mathcal{W})} \cdot \breve{\boldsymbol{\pi}}_4^{(\mathcal{W})}$ because their inner products are always equal to minus one. The other hypotheses are $\pm\mathbf{u}_1 \cdot \mathbf{u}_2$. Expanding this inner product by using Equations (9.6) and (9.7), we get

$$\pm\mathbf{u}_1 \cdot \mathbf{u}_2 = \pm\frac{2 - \sigma_3^2 - \sigma_1^2}{\sigma_1^2 - \sigma_3^2}. \tag{D.7}$$

From this result, one obtains:

$\mathbf{u}_1 \cdot \mathbf{u}_2 = 1$ implies $\sigma_1^2 = 1$ for any σ_3^2;

$\mathbf{u}_1 \cdot \mathbf{u}_2 = -1$ implies $\sigma_3^2 = 1$ for any σ_1^2.

If we consider $\sigma_1^2 > 1 > \sigma_3^2$, we have $\breve{\boldsymbol{\pi}}_i^{(\mathcal{W})} \cdot \breve{\boldsymbol{\pi}}_j^{(\mathcal{W})} \neq 1$ for any $i \neq j$ which proves the Lemma. □

From the previous lemma, we proved that for $\sigma_1^2 > 1 > \sigma_3^2$, there will be a single solution for the *homography* decomposition. However, there are other special cases that have to be considered, namely: when $\sigma_1^2 = 1$ and $\sigma_3^2 < 1$, $\sigma_1^2 > 1$ and $\sigma_3^2 = 1$ and $\sigma_1^2 = \sigma_3^2 = 1$.

Lemma D.2. *Given a homography matrix, $\mathbf{H} \in \mathcal{H}$, and known plane coordinates $\boldsymbol{\Pi}^{(\mathcal{W})}\mathbb{R} = \left(-\zeta^{(\mathcal{W})}, \boldsymbol{\pi}^{(\mathcal{W})}\right) \in \mathbb{R}^4$ where $\left|\boldsymbol{\pi}^{(\mathcal{W})}\right| = 1$, if the singular values of $\mathbf{H}$ verify $\sigma_1^2 = 1$ and $\sigma_3^2 < 1$ or $\sigma_1^2 > 1$ and $\sigma_3^2 = 1$, its decomposition into rotation and translation has a unique solution.*

Proof. Let us consider the decomposition presented in Lemma 9.1. From Equations (9.6) and (9.7), $\sigma_1^2 = 1$ and $\sigma_3^2 < 1$ implies $\mathbf{u}_1 = \mathbf{u}_2$, and as a result, from Equations (D.1–D.6) we could have two of pairs of solutions (**Solutions 1 and 2** or **Solutions 3 and 4**).

However, from Equations (9.8–9.11), for $\mathbf{u}_1 = \mathbf{u}_2$ we have $\mathbf{U}^{(1)} = \mathbf{U}^{(2)}$ and $\mathbf{W}^{(1)} = \mathbf{W}^{(2)}$ and from Equations (9.12) and (9.13), we conclude that the **Solution 1** will be equal to **Solution 2** and **Solution 3** will be equal to **Solution 4**. As a result, it is proven that for $\sigma_1^2 = 1$ and $\sigma_3^2 < 1$, we also have a single solution.

For $\sigma_3^2 = 1$ and $\sigma_1 > 1$, we will have $\mathbf{u}_1 = -\mathbf{u}_2$, which will imply that we could have pairs of solutions (**Solutions 1 and 4** and **Solutions 2 and 3**). However, in that case the product $\mathbf{W}^{(1)}\left(\mathbf{U}^{(1)}\right)^T$ will be equal to $\mathbf{W}^{(2)}\left(\mathbf{U}^{(2)}\right)^T$ and as a result $\mathbf{R}^{(1)} = \mathbf{R}^{(2)} = \mathbf{R}^{(3)} = \mathbf{R}^{(4)}$. From this result, we conclude that **Solution 1** is equal to solution **Solution 4** and **Solution 3** is equal to **Solution 2** which proves that we still have a single solution for the *homography* decomposition. □

Lemma D.3. *Given a homography matrix,* $\mathbf{H} \in \mathcal{H}$ *if the singular values of* $\mathbf{H}$ *verify* $\sigma_1^2 = \sigma_3^2 = 1$*, its decomposition into rotation and translation has a unique solution, which is* $\mathbf{R} = \mathbf{H}$ *and* $\mathbf{t} = \mathbf{0}$.

Proof. When $\sigma_1^2 = \sigma_3^2 = 1$ we have $\mathbf{\Sigma} = \mathbf{I}$ which means that the *homography* matrix corresponds to a pure rotation $\mathbf{H} \in \mathcal{SO}(3)$. As a result, we can define $\mathbf{R} = \mathbf{H}$ and $\mathbf{t} = \mathbf{0}$. □

D.2 *Proof* of the Theorem 9.2

Using the definition of the *kronecker* product – Section 3.1, and rearranging rows and columns, we can get a matrix $\mathbf{M}^{(2)}$ such that $\operatorname{rank}\left(\mathbf{M}^{(2)}\right) = \operatorname{rank}(\mathbf{M})$ (note that the permutation of rows and columns does not change the *dimension of the column–space*) and

$$\mathbf{M}^{(2)} = \begin{bmatrix} \mathbf{0} & -\mathbf{D}^{(3)}\mathbf{P} & \mathbf{D}^{(2)}\mathbf{P} \\ \mathbf{D}^{(3)}\mathbf{P} & \mathbf{0} & -\mathbf{D}^{(1)}\mathbf{P} \\ -\mathbf{D}^{(2)}\mathbf{P} & \mathbf{D}^{(1)}\mathbf{P} & \mathbf{0} \end{bmatrix} \tag{D.8}$$

where

$$\mathbf{D}^{(i)} = \begin{bmatrix} d_i^{(1)} & 0 & 0 \\ 0 & d_i^{(2)} & 0 \\ 0 & 0 & d_i^{(3)} \end{bmatrix} \text{ and } \mathbf{P} = \begin{bmatrix} \mathbf{p}_1^{(\mathcal{W})\,T} \\ \mathbf{p}_2^{(\mathcal{W})\,T} \\ \mathbf{p}_3^{(\mathcal{W})\,T} \end{bmatrix}, \tag{D.9}$$

and $d_i^{(j)}$ is the ith element of the vector $\mathbf{d}_j^{(\mathcal{C})}$. On the other hand, matrix $\mathbf{M}^{(2)}$ can be decomposed as

$$\mathbf{M}^{(2)} = \underbrace{\begin{bmatrix} \mathbf{0} & -\mathbf{D}^{(3)} & \mathbf{D}^{(2)} \\ \mathbf{D}^{(3)} & \mathbf{0} & -\mathbf{D}^{(1)} \\ -\mathbf{D}^{(2)} & \mathbf{D}^{(1)} & \mathbf{0} \end{bmatrix}}_{\mathbf{Q}^{(1)}} \underbrace{\begin{bmatrix} \mathbf{P} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{P} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{P} \end{bmatrix}}_{\mathbf{Q}^{(2)}}. \tag{D.10}$$

$$\mathbf{Q}^{(1)} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & -d_3^{(1)} & 0 & 0 & d_2^{(1)} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ d_3^{(1)} & 0 & 0 & 0 & 0 & 0 & -d_1^{(1)} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -d_2^{(1)} & 0 & 0 & d_1^{(1)} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{S}^{(1)}} + \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -d_3^{(2)} & 0 & 0 & d_2^{(2)} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & d_3^{(2)} & 0 & 0 & 0 & 0 & 0 & -d_1^{(2)} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -d_2^{(2)} & 0 & 0 & d_1^{(2)} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{S}^{(2)}} +$$

$$+ \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -d_3^{(3)} & 0 & 0 & -d_2^{(3)} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & d_3^{(3)} & 0 & 0 & 0 & 0 & 0 & -d_1^{(3)} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -d_2^{(3)} & 0 & 0 & d_1^{(3)} & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{S}^{(3)}} \tag{D.11}$$

Since we are considering that the points in the world $\left\{\mathbf{p}_i^{(\mathcal{W})}\right\}$ define a plane that does not pass through the origin, we conclude that $\text{rank}\,(\mathbf{P}) = 3$ and as a result $\text{rank}\left(\mathbf{Q}^{(2)}\right) = 9$. Since the *dimension of the column–space* of $\mathbf{Q}^{(2)}$ is equal to the number of rows, one can conclude that $\text{rank}\left(\mathbf{M}^{(2)}\right) = \text{rank}\left(\mathbf{Q}^{(1)}\right)$.

In addition, we can decompose $\mathbf{Q}^{(1)}$ into $\mathbf{Q}^{(1)} = \mathbf{S}^{(1)} + \mathbf{S}^{(2)} + \mathbf{S}^{(3)}$ as it is shown in Equation (D.11). Since matrices $\mathbf{S}^{(i)}$ (for $i = 1, 2, 3$) represent orthogonal subspaces [Str80], we can write $\mathbf{S}^{(i)} \cap \mathbf{S}^{(j)} = \{\}$ for $i \neq j$ and, as a result,

$$\text{rank}\left(\mathbf{Q}^{(1)}\right) = \sum_{i=1}^{3} \text{rank}\left(\mathbf{S}^{(i)}\right). \tag{D.12}$$

Moreover and eliminating zero rows and columns of each matrix $\mathbf{S}^{(i)}$, we can see that $\text{rank}\left(\mathbf{S}^{(i)}\right) = \text{rank}\left(\widehat{\mathbf{d}}_i^{(\mathcal{C})}\right)$, and as a result

$$\text{rank}\left(\mathbf{Q}^{(1)}\right) = \sum_{i=1}^{3} \text{rank}\left(\widehat{\mathbf{d}}_i^{(\mathcal{C})}\right). \tag{D.13}$$

Since $\mathbf{d}_i^{(\mathcal{C})}$ are non–zero vectors, we can conclude that $\text{rank}\left(\widehat{\mathbf{d}}_i^{(\mathcal{C})}\right) = 2$, for all i, and as a result $\text{rank}\left(\mathbf{Q}^{(1)}\right) = 6$.

To conclude, $\text{rank}\left(\mathbf{Q}^{(1)}\right) = 6$ implies $\text{rank}\left(\mathbf{M}^{(2)}\right) = 6$ and as a result $\text{rank}\left(\mathbf{M}\right) = 6$, which proves the theorem.

D.3 *Proof* of the Proposition 9.2

From the definition, a matrix $\breve{\mathbf{H}} \in \mathbb{R}^{3\times3}$ represents a rigid transformation for points in $\mathbf{\Pi}^{(\mathcal{W})}$ if it preserves the distance between points on that plane. As a result, $\breve{\mathbf{H}}$ only defines a rigid transformation if ${\mathbf{q}_{i,j}^{(\mathcal{W})}}^T \mathbf{q}_{i,j}^{(\mathcal{W})} = {\mathbf{q}_{i,j}^{(\mathcal{W})}}^T \breve{\mathbf{H}}^T\breve{\mathbf{H}}\mathbf{q}_{i,j}^{(\mathcal{W})}$ holds, where $\mathbf{p}_i^{(\mathcal{W})}, \mathbf{p}_j^{(\mathcal{W})}$ are any two points that belong to $\mathbf{\Pi}^{(\mathcal{W})}$. For any $\mathbf{p}_i^{(\mathcal{W})}, \mathbf{p}_j^{(\mathcal{W})} \in \mathbf{\Pi}^{(\mathcal{W})}$ and

$$\mathbf{q}_{i,j}^{(\mathcal{W})} \in \mathcal{S}^2 \doteq \mathbf{p}_i^{(\mathcal{W})} - \mathbf{p}_j^{(\mathcal{W})} \tag{D.14}$$

where $\mathcal{S}^2$ is a two dimensional vector subspace. Let us consider three points $\mathbf{p}_1^{(\mathcal{W})}$, $\mathbf{p}_2^{(\mathcal{W})}$ and $\mathbf{p}_3^{(\mathcal{W})}$ that define the plane $\mathbf{\Pi}^{(\mathcal{W})}$. Using these points, we can define $\mathbf{q}_{1,2}^{(\mathcal{W})}$ and $\mathbf{q}_{1,3}^{(\mathcal{W})}$ and since $\breve{\mathbf{H}}$ must represent a rigid transformation, the following conditions must hold

$${\mathbf{q}_{1,2}^{(\mathcal{W})}}^T \mathbf{q}_{1,2}^{(\mathcal{W})} = {\mathbf{q}_{1,2}^{(\mathcal{W})}}^T \breve{\mathbf{H}}^T\breve{\mathbf{H}}\mathbf{q}_{1,2}^{(\mathcal{W})} \tag{D.15}$$

$${\mathbf{q}_{1,3}^{(\mathcal{W})}}^T \mathbf{q}_{1,3}^{(\mathcal{W})} = {\mathbf{q}_{1,3}^{(\mathcal{W})}}^T \breve{\mathbf{H}}^T\breve{\mathbf{H}}\mathbf{q}_{1,3}^{(\mathcal{W})}. \tag{D.16}$$

In addition $\breve{\mathbf{H}}$ must define a rigid transformation for any vector $\mathbf{q}^{(\mathcal{W})} \in \mathcal{S}^2$ so that

$${\mathbf{q}^{(\mathcal{W})}}^T \mathbf{q}^{(\mathcal{W})} = {\mathbf{q}^{(\mathcal{W})}}^T \breve{\mathbf{H}}^T\breve{\mathbf{H}}\mathbf{q}^{(\mathcal{W})}. \tag{D.17}$$

If $\breve{\mathbf{H}}$ defines a rigid transformation for points that belong to the plane $\mathbf{\Pi}^{(\mathcal{W})}$, it also verifies the Proposition 1, which means $\breve{\mathbf{H}} \in \mathcal{H}$.

Let us consider that $\mathbf{q}_{1,2}^{(\mathcal{W})}, \mathbf{q}_{1,3}^{(\mathcal{W})} \in \mathcal{S}^2$ are linearly independent vectors (which they are necessarily). As a result, any $\mathbf{q}^{(\mathcal{W})} \in \mathcal{S}^2$ can be defined as a linear combination of these vectors, $\mathbf{q}^{(\mathcal{W})} \doteq \lambda\mathbf{q}_{1,2}^{(\mathcal{W})} + \beta\mathbf{q}_{1,3}^{(\mathcal{W})}$, for $\lambda, \beta \in \mathbb{R}$. Using this representation, we can rewrite Equation (D.17) as

$$\begin{aligned}\lambda^2\, {\mathbf{q}_{1,2}^{(\mathcal{W})}}^T \mathbf{q}_{1,2}^{(\mathcal{W})} + \beta^2\, {\mathbf{q}_{1,3}^{(\mathcal{W})}}^T \mathbf{q}_{1,3}^{(\mathcal{W})} + 2\lambda\beta\, {\mathbf{q}_{1,2}^{(\mathcal{W})}}^T \mathbf{q}_{1,3}^{(\mathcal{W})} = \\ \lambda^2\, {\mathbf{q}_{1,2}^{(\mathcal{W})}}^T \breve{\mathbf{H}}^T\breve{\mathbf{H}}\mathbf{q}_{1,2}^{(\mathcal{W})} + \beta^2\, {\mathbf{q}_{1,3}^{(\mathcal{W})}}^T \breve{\mathbf{H}}^T\breve{\mathbf{H}}\mathbf{q}_{1,3}^{(\mathcal{W})} + 2\lambda\beta\, {\mathbf{q}_{1,2}^{(\mathcal{W})}}^T \breve{\mathbf{H}}^T\breve{\mathbf{H}}\mathbf{q}_{1,3}^{(\mathcal{W})}\end{aligned} \tag{D.18}$$

and since the condition of Equations (D.15) and (D.16) must hold, we can rewrite this equation as

$${\mathbf{q}_{1,2}^{(\mathcal{W})}}^T \mathbf{q}_{1,3}^{(\mathcal{W})} = {\mathbf{q}_{1,2}^{(\mathcal{W})}}^T \breve{\mathbf{H}}^T\breve{\mathbf{H}}\mathbf{q}_{1,3}^{(\mathcal{W})}. \tag{D.19}$$

If this constraint is satisfied, Equation (D.17) holds and as a result, $\breve{\mathbf{H}}$ defines a rigid transformation for all points that belong to the plane $\mathbf{\Pi}^{(\mathcal{W})}$ which proves the proposition.

If we use $\mathbf{q}_{23}^{(\mathcal{W})}$, which is a linear combination of $\mathbf{q}_{1,2}^{(\mathcal{W})}$ and $\mathbf{q}_{1,3}^{(\mathcal{W})}$, constraint ${\mathbf{q}_{2,3}^{(\mathcal{W})}}^T \mathbf{q}_{23}^{(\mathcal{W})} = {\mathbf{q}_{2,3}^{(\mathcal{W})}}^T \breve{\mathbf{H}}^T \breve{\mathbf{H}} \mathbf{q}_{2,3}^{(\mathcal{W})}$ holds only if Equation (D.19) is satisfied. As a result, a matrix $\breve{\mathbf{H}} \in \mathbb{R}^{3\times 3}$ that satisfies ${\mathbf{q}_{i,j}^{(\mathcal{W})}}^T \mathbf{q}_{i,j}^{(\mathcal{W})} = {\mathbf{q}_{i,j}^{(\mathcal{W})}}^T \breve{\mathbf{H}}^T \breve{\mathbf{H}} \mathbf{q}_{i,j}^{(\mathcal{W})}$ for $(i,j) = \{(1,2),(1,3),(2,3)\}$ defines a rigid transformation for points belonging to plane $\mathbf{\Pi}^{(\mathcal{W})}$ which proves the proposition.

Bibliography

[ACB98] Helder Araujo, Rodrigo L. Carceroni, and Christopher M. Brown. A Fully Projective Formulation to Improve the Accuracy of Lowe's Pose–Estimation Algorithm. *Computer Vision and Image Understanding*, 1998.

[AD03] Adnan Ansar and Kostas Daniilidis. Linear Pose Estimation from Points or Lines. *IEEE Trans. Pattern Annalysis and Machine Intelligence*, 2003.

[AHB87] K. S. Arun, T. S. Huang, and S. D. Blostein. Least–Square Fitting of Two $3 - D$ Point Sets. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1987.

[ATR10] Amit Agrawal, Yuichi Taguchi, and Srikumar Ramalingam. Analytical Forward Projection for Axial Non–Central Dioptric & Catadioptric Cameras. *Proc. European Conf. Computer Vision (ECCV)*, 2010.

[ATR11] Amit Agrawal, Yuichi Taguchi, and Srikumar Ramalingam. Beyond Alhazen's Problem: Analytical Projection Model for Non-Central Catadioptric Cameras with Quadric Mirrors. *IEEE Proc. Computer Vision and Pattern Recognition (CVPR)*, 2011.

[BN99] Simon Baker and Shree K Nayar. A Theory of Single–Viewpoint Catadioptric Image Formation. *Int'l J. of Computer Vision*, 1999.

[Bou10] Jean-Yves Bouguet. Camera Calibration Toolbox for Matlab. *[Oline]*, 2010.

[BS05] Adrien Bartoli and Peter Sturm. Structure–From–Motion Using Lines: Representation, Triangulation and Bundle Adjustment. *Computer Vision and Image Understanding*, 2005.

[Buh03] Martin Buhmann. *Radial Basis Functions: Theory and Implementations*. Cambridge University Press, 2003.

[CC02] Chu-Song Chen and Wen-Yan Chang. Pose Estimation for Generalized Imaging Device via Solving Non–Prespective n Point Problem. *IEEE Proc. Int.l Conf. Robotics and Automation (ICRA)*, 2002.

[CC04a] Chu-Song Chen and Wen-Yan Chang. On Pose Recovery for Generalized Visual Sensers. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2004.

[CC04b] Chu-Song Chen and Wen-Yan Chang. On Pose Recovery for Generalized Visual Sensors. *IEEE Trans. Pattern Annalysis and Machine Intelligence*, 2004.

[CLO04] David A. Cox, John Little, and Donal O'Shea. *Using Algebraic Geometry*. Springer Science+Business, 2004.

[FB81] Martin A Fischler and Robert C Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applicatlons to Image Analysis and Automated Cartography. *Comm. of the ACM*, 1981.

[GH97] Rajiv Gupta and Richard I Hartley. Linear Pushbroom Cameras. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1997.

[GN99] Joshua Gluckman and Shree K. Nayar. Planar Catadioptric Stereo: Geometry and Calibration. *IEEE Proc. Computer Vision and Pattern Recognition (CVPR)*, 1999.

[GN01] M D Grossberg and S K Nayar. A General Imaging Model and a Method for Finding its Parameters. *IEE Proc. Int'l Conf. Computer Vision (ICCV)*, 2001.

[GN05] M D Grossberg and S K Nayar. The Raxel Imaging Model and Ray–Based Calibration. *Int'l J. of Computer Vision*, 2005.

[Gon08] Nuno Goncalves. *Noncentral Catadioptric Systems with Quadric Mirrors, Geometry and Calibration*. PhD thesis, University of Coimbra, October 2008.

[GSB09] Simone Gasparini, Peter Sturm, and J P Barreto. Plane–Based Calibration of Central Catadioptric Cameras. *IEE Proc. Int'l Conf. Computer Vision (ICCV)*, 2009.

[Guo12] Yang Guo. A Novel Solution to the P4P Problem for an Uncalibrated Camera. *J. Math. Imaging Vis*, 2012.

[GVL96] Gene H Golub and Charles F Van Loan. *Matrix Computations (3rd ed.)*. Johns Hopkins University Press, 1996.

[Har97] Richard Hartley. In Defense of the Eight–Point Algorithm. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1997.

[HF09] Shaoxing Hu and Xuyong Feng. Calibration of Panorama Based on Linear CCD and Fisheye Lens. *Int'l Congress Image and Signal Processing (CISP)*, 2009.

[HJ91] Roger A. Horn and Charles R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991.

[HK05] Richard Hartley and Sing Bing Kang. Parameter–free Radial Distortion Correction with Centre of Distortion Estimation. *IEE Proc. Int'l Conf. Computer Vision (ICCV)*, 2005.

[HL12] Richard Hartley and Hongdong Li. An Efficient Hidden Variable Approach to Minimal-Case Camera Motion Estimation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2012.

[HLON94] Robert Haralick, Chung-Nan Lee, Karsten Ottenberg, and Michael Nolle. Review and Analysis of Solutions of the Three Point Perspective Pose Estimation Problem. *Int'l J. Computer Vision*, 1994.

[HR11] Joel Hesch and Stergios Roumeliotis. A Direct Least-Squares (DLS) Method for PnP. *IEEE Proc. Int'l Conf. Computer Vision (ICCV)*, 2011.

[HZ00] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[KB06] Juho Kannala and Sami S Brandt. A Generic Camera Model and Calibration Method for Conventional, Wide–Angle, and Fish-Eye Lenses. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2006.

[KBP08] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. Automatic Generator of Minimal Problem Solvers. *Proc. European Conf. Computer Vision (ECCV)*, 2008.

[KBP12a] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. Polynomial Eigenvalue Solutions to Minimal Problems in Computer Vision. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2012.

[KBP12b] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. Polynomial Eigenvalue Solutions to Minimal Problems in Computer Vision. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2012.

[Kin92] Rudolf Kingslake. *Optics in Photography*. SPIE Optical Engineering Press, 1992.

[KS08] Kriakos N Kutulakos and Eron Steger. A Theory of Refractive and Specular 3D Shape by Light–Path Triangulation. *Int'l J. of Computer Vision*, 2008.

[KSS11] Laurent Kneip, Davide Scaramuzza, and Roland Siegwart. A Novel Parametrization of the Perspective–Three–Point Problem for a Direct Computation of Absolute Camera Position and Orientation. *IEEE Proc. Computer Vision and Patter Recognition (CVPR)*, 2011.

[LHM00] Chien-Ping Lu, Gregory D Hager, and Eric Mjolsness. Fast and Globally Convergent Pose Estimation from Video Images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2000.

[MA10] Pedro Miraldo and Helder Araujo. Improving the Resolution of the Generic Camera Model by Means of a Parametric Representation. *Controlo*, 2010.

[MA13] Pedro Miraldo and Helder Araujo. Calibration of Smooth Camera Models. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2013.

[MAQ11] Pedro Miraldo, Helder Araujo, and João Filipe Queiró. Point-based Calibration Using a Parametric Representation of the General Imaging Model. *IEE Proc. Int'l Conf. Computer Vision (ICCV)*, 2011.

[MNLF07] Francesc Moreno Noguer, Vicent Lepetit, and Pascal Fua. Accurate Non–Iteractive $O(n)$ Solution to the PnP Problem. *IEEE Proc. Int.l Conf. Computer Vision (ICCV)*, 2007.

[MP04a] Marc Menem and Tomas Pajdla. Constraints on perspective images and circular panoramas. *Proc. British Machine Vision Conf. (BMVC)*, 2004.

[MP04b] B Micusik and T Pajdla. Autocalibration & $3D$ Reconstruction with Non–Central Catadioptric Cameras. *IEEE Proc. Computer Vision and Pattern Recognition (CVPR)*, 2004.

[MSKS04] Yi Ma, Stefano Soatto, Jana Košecká, and S. Shankar Sastry. *An Invitation to 3D Vision: From Images to Geometry Models*. Springer Science+Business, 2004.

[MV07] Ezio Malis and Manuel Vargas. Deeper Understanding of the Homography Decomposition for Vision–Based Control. Technical report, Institut National de Recherche en Informatique et en Automatique (INRIA Sophia Antipolis), 2007.

[NHS07] David Nistér, Richard Hartley, and Henrik Stewénius. Using Galois Theory to Prove Structure from Motion Algorithms are Optimal. *Proc. Computer Vision and Patter Recognition (CVPR)*, 2007.

[Nis03] David Nistér. Preemptive RANSAC for Live Structure and Motion Estimation. *IEEE Proc. Int.l Conf. Computer Vision (ICCV)*, 2003.

[Nis04a] David Nistér. A Minimal Solution to the Generalized 3–Point Pose Problem. *IEEE Proc. Computer Vision and Patter Recognition (CVPR)*, 2004.

[Nis04b] David Nistér. An Efficient Solution to the Five–Point Relative Pose Problem. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2004.

[Nor09] Northern Digital Incorporated. Optotrak Certus Motion Capture System, 2009. `www.ndigital.com`.

[NS07] David Nistér and Henrik Stewénius. A Minimal Solution to the Generalized 3–Point Pose Problem. *In J. Math. Imaging Vis*, 2007.

[NSG05] David Nistér, Henrik Stewénius, and Etienne Grossmann. Non–Parametric Self–Calibration. *IEE Proc. Int'l Conf. Computer Vision (ICCV)*, 2005.

[Ple03] Robert Pless. Using Many Cameras as One. *IEEE Proc. Computer Vision and Pattern Recognition (CVPR)*, 2003.

[Pon09] J Ponce. What is a Camera? *IEEE Proc. Computer Vision and Pattern Recognition (CVPR)*, 2009.

[PW01] H Pottmann and J Wallner. *Computational Line Geometry*. Springer–Verlag, 2001.

[QSW93] E Quak, N Sivakumar, and J D Ward. Least Squares Approximation by Radial Functions. *SIAM Journal on Mathematical Analysis*, 1993.

[RLS06] Srikumar Ramalingam, Suresh K Lodha, and Peter Sturm. A Generic Structure–from–Motion Framework. *Computer Vision and Image Understanding*, 2006.

[RSL05] Srikumar Ramalingam, Peter Sturm, and S K Lodha. Towards Complete Generic Camera Calibration. *IEEE Proc. Computer Vision and Pattern Recognition (CVPR)*, 2005.

[RSLSK10] Srikumar Ramalingam, Peter Sturm, and Lodha Suresh K. Generic self-calibration of central cameras. *Computer Vision and Image Understanding*, 2010.

[Sch66] Peter Schönemann. A Generalized Solution of the Orthogonal Procrustes Problem. *Psychometrika*, 1966.

[SGN03] Rahul Swaminathan, Michael D Grossberg, and Shree K Nayar. A Prespective on Distortions. *IEEE Proc. Computer Vision and Pattern Recognition (CVPR)*, 2003.

[SGN06] R Swaminathan, M D Grossberg, and S K Nayar. Non–Single Viewpoint Catadioptric Cameras: Geometry and Analysis. *Int'l J. of Computer Vision*, 2006.

[SP08] Gerald Schweighofer and Axel Pinz. Globally Optimal $O(n)$ Solution to the PnP Problem for General Camera Models. *Proc. British Machine Vision Conf. (BMVC)*, 2008.

[SR04] Peter Sturm and Srikumar Ramalingam. A Generic Concept for Camera Calibration. *Proc. European Conf. Computer Vision (ECCV)*, 2004.

[SRT+11] Peter Sturm, Srikumar Ramalingam, Jean-Philippe Tardif, Simone Gasparini, and João Barreto. Camera Models and Fundamental Concepts Used in Geometric Computer Vision. *Foundations and Trends in Computer Graphics and Vision*, 2011.

[Ste05] Henrik Stewénius. *Gröbner Basis Methods for Minimal Problems in Computer Vision*. PhD Thesis, Lund University, 2005.

[Str80] Gilbert Strang. *Linear Algebra and its Applications*. Academic Press, 1980.

[Stu05] Peter Sturm. Multi-View Geometry for General Camera Models. *IEEE Proc. Computer Vision and Pattern Recognition (CVPR)*, 2005.

[SW93] N Sivakumar and J D Ward. On the least squares fit by radial functions to multidimensional scattered data. *Numerische Mathematik*, 1993.

[TM99] Seth Teller and Hohmeyer Michael. Determining the Lines Through Four Lines. *J. Graphics Tools*, 1999.

[TP12] SriRam Thirthala and Marc Pollefeys. Radial Multi-focal Tensors – Applications to Omnidirectional Camera Calibration. *Int'l J. of Computer Vision*, 2012.

[Tsa86] R Y Tsai. An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision. *IEEE Proc. Computer Vision and Pattern Recognition (CVPR)*, 1986.

[TSKS12] Tali Treibitz, Yoav Y. Schechner, Clayton Kunz, and Hanumant Singh. Flat Refractive Geometry. *IEEE Trans. Pattern Annalysis and Machine Intelligence*, 2012.

[TSS08] Tali Treibitz, Yoav Y Schechner, and Hanumant Singh. Flat Refractive Geometry. *IEEE Proc. Computer Vision and Pattern Recognition (CVPR)*, 2008.

[Ume91] Shinji Umeyama. Least–Square Estimation of Transformation Between Two Point Patterns. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1991.

[Wen05] Holger Wendland. *Scattered Data Approximation*. Cambridge University Press, 2005.

[YM04] Jingyi Yu and Leonard McMillan. General Linear Cameras. *Proc. European Conf. Computer Vision (ECCV)*, 2004.

[ZFPW03] Assaf Zomet, Doron Feldman, Shmuel Peleg, and Daphna Weinshall. Mosaicing New Views: The Crossed–Slits Projection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2003.

[Zha00] Zhengyou Zhang. A Flexible New Technique for Camera Calibration. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2000.