Departamento de
Informática e Ingeniería
de Sistemas
Universidad Zaragoza

Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza

Line-projections in omnidirectional vision:

Modelling, extraction and calibration in central and non-central cameras

Jesús Bermúdez Cameo

Ph.D. Dissertation

Advisors: Gonzalo López Nicolás and José Jesús Guerrero Campo

Departamento de Informática e Ingeniería de Sistemas
Escuela de Ingeniería y Arquitectura
Universidad de Zaragoza, Spain

December 2015

Acknowledgments

These lines are dedicated to everyone who was directly or indirectly involved in the development of this thesis. First of all, I would like to thank my supervisors, Josechu Guerrero and Gonzalo López first for giving to me the opportunity of working with them in the robotics group, and second for these years of advising and discussions. I would also like to thank Luis Puig, who guided me during my first research work and supervised me during my engineering final project and master thesis.

Also many thanks to the committee members: Carlos Sagüés Blázquiz, Arturo de la Escalera Hueso, Cédric Demonceaux, David Fofi, and Antonio M. López Peña and to the international reviewers: Jean Charles Bazin and Luis Puig, for spending their time evaluating my work.

During this time I have had the opportunity of knowing other research groups in Europe. I would like to thank Professor João P. Barreto for the inspiring discussions and the invaluable knowledge about geometry, complexes and congruences and also to all his team at the University of Coimbra: Rui Melo, Carolina Raposo, Vitor Pedro, Miguel Lourenço, Michel Antunes and I would like to send a special greeting to Francisco Vasconcellos who help me understanding how to solve systems of polynomials.

I would also like to thank Professor Marc Pollefeys for giving me the opportunity of being part of one of the top computer vision laboratories at the ETH Zurich. I would like to thank Olivier Saurer for the inspiring conversations, Federico Camposeco for welcoming me to his home, the staff of the CVG for helping me to find an accommodation, and all the people working at the CVG: Lionel Heng, Petri Tanskanen, Andrea Cohen, Dominik Honegger, Cristian Häne, Helen Oleynikova, Lorenz Meier, Tobias Nägeli, Dario and all the people I forget.

I would also like to thank Professor Cédric Demonceaux for the inspiring conversations about geometry and the PnP problem, and also for helping me with the accommodation, repairing my car and other technical problems. Of course, I would like to dedicate some words for the team at the IUT Le Creusot: Professor David Fofi, Fabrice Meriaudeau, Désiré Sidibé, Cansen Jiang for the inspiring questions, Joan Massich, Mojdeh Rastgoo, Guillaume Lemaître, Amad, Quinglin Lu, David Strubel, Priyanka Phutane, Muhammad Usman, Panagiotis Palantas and all the people I forget.

I would like to thank the Spanish Ministry of Education for the FPU scholarship (AP2010-3849). I would like to thank Alejandro Rituerto, Ricardo J. Rodríguez and Diego Castán as members of the prestigious breakfast commission, the PhD students of the DIIS (Alejandro Pérez, Rosario Aragües, Raúl Mur, Sara Ruiz, Marilú Rodríguez, Daniel Gutiérrez, Estíbaliz Fraca, Henri Carrillo, Yasir Latif, Ana Cambra, Leonardo Fermín, Alejo Concha, Javier Amingol, Dorian Gálvez, Alberto Badías, Antonio Agudo, Eduardo Montijano et al.) and all the members of the ROPERT group at the University of Zaragoza.

I would like to thanks my parents Fernando and Teresa, my sisters Rocío and Lucía and all my family for supporting me and for the wonderful education I have received. Finally I want to thank Marta for supporting and encouraging me all these years and for being so amazing person.

Resumen

La visión por ordenador tiene un interés creciente en las tecnologías emergentes. Uno de los retos de esta disciplina es el diseño y estudio de cámaras con un amplio campo de vista que permiten obtener más información del entorno a partir de una única imagen. En particular, la visión omnidireccional es útil para la estimación de la localización en robótica, en conducción autónoma y en vehículos aéreos autónomos, entre otras aplicaciones. El amplio campo de vista de las cámaras omnidireccionales permite aprovechar la descripción de la escena usando características basadas en líneas rectas. Por un lado, las características basadas en rectas representan referencias naturales en entornos fabricados por el hombre y coinciden con los bordes de los elementos constructivos. Además son muy informativas en entornos de interior con bajo nivel de textura. Por otro lado, los segmentos largos, que son especialmente útiles para reducir derivas de localización, suelen ser completamente visibles en la proyección omnidireccional. Sin embargo, en cámaras omnidireccionales, la proyección de una recta sufre una deformación debida a la distorsión y se convierte en una curva más compleja. Esta tesis se centra en la geometría de las proyecciones de rectas (o imágenes de recta) en sistemas omnidireccionales. El principal tema tratado en este trabajo es la extracción de proyecciones de recta en distintos tipos de cámaras centrales y no centrales. Sin embargo, debido a la naturaleza de la propia proyección en las cámaras omnidireccionales, también se ha investigado en calibración a partir de proyecciones de rectas y en el caso de cámaras no centrales se ha estudiado la reconstrucción 3D a partir de rectas en una sola imagen. En particular, se han analizado los siguientes problemas relacionados con las proyecciones de rectas en cámaras omnidireccionales centrales y no centrales:

- **Extracción de proyecciones de rectas en cámaras omnidireccionales centrales.** El primero de los objetivos abordados en esta tesis es la extracción de rectas en imágenes hiper-catadióptricas para la estimación de direcciones dominantes en escenarios bajo condiciones de Manhattan. Además, el enfoque de extracción utilizado en este primer trabajo se ha generalizado para el resto de sistemas centrales obteniendo un marco teórico para las ecuaciones de las curvas que describen proyecciones de rectas en los sistemas catadióptricos y dióptricos. Usando este marco teórico común se ha desarrollado la restricción *plumb-line* para cada uno de estos sistemas obteniendo soluciones analíticas que permiten la extracción simultánea de la proyección de recta y de la calibración de la cámara. Estas soluciones teóricas se han integrado en un método automático para extracción de proyecciones de rectas cuando no se conoce la calibración en cámaras omnidireccionales centrales. Esta propuesta utiliza posiciones de puntos y direcciones de gradientes en estos puntos de manera que se puede reducir la complejidad del esquema de extracción.

- **Extracción de proyecciones de rectas y ajuste 3D de rectas en cámaras omnidireccionales no centrales.**

  Las características de la proyección en sistemas no centrales permite la reconstrucción completa de una recta 3D a partir de de una sola proyección. Los sistemas no centrales estudiados en esta tesis son los sistemas catadióptricos cónico y esférico y el panorama no central. Generalizando la descripción utilizada anteriormente se han desarrollado las ecuaciones analíticas para los casos de sistemas catadióptricos cónicos y esféricos. En el caso cónico podemos estimar la geometría del espejo y la reconstrucción de la recta 3D a partir de cinco puntos de la proyección de la recta. El principal problema de reconstruir en 3D a partir de una única proyección de recta en sistemas no centrales es que la precisión de los resultados es muy sensible al ruido ya que hay involucrados cuatro grados de libertad. Por ello, la extracción en rectas en sistemas no centrales es un problema abierto de difícil solución. Para abordar este problema se proponen diferentes enfoques y soluciones. Para condicionar la selección de puntos en un esquema de estimación robusto y evaluar la precisión se introduce el concepto de *línea de base efectiva* entre un conjunto de rayos que depende fuertemente del tipo de sistema no central. Respecto a precisión en el ajuste 3D se ha observado que utilizando el sistema panorámico no central se obtienen los mejores resultados aunque la complejidad constructiva dificulta su uso en la práctica. Para resolver la extracción automática de proyecciones de recta en sistemas no centrales se propone un enfoque robusto que usa un paso de pre-evaluación similar al usado en PROSAC y diferentes tipos de distancias. En particular, se ha resuelto analíticamente la distancia Euclídea de punto a proyección de recta para los sistemas catadióptricos cónico y esférico y se propone una distancia basada en la proyección del punto más cercano en el espacio para el caso de panorama no-central.

- **Ajuste de líneas en sistemas no centrales mediante la imposición de restricciones geométricas.**

  Un planteamiento útil para incrementar la precisión y la estabilidad en la reconstrucción de rectas en sistemas no centrales es explotar restricciones geométricas adicionales utilizando información a priori. En esta tesis, se proponen nuevas soluciones mínimas para imponer restricciones entre pares de rectas en sistemas no centrales. En concreto, se han desarrollado los casos de rectas secantes ortogonales y rectas paralelas. Finalmente, también se presenta la solución mínima para el cálculo de recta paralela a un plano dado que permite explotar la dirección de la gravedad cuando se asumen condiciones de Manhattan en el escenario.

Abstract

Computer vision has an increasing interest in most fields of emerging technologies. A challenging topic in this field is to study how to enlarge the field of view of the camera systems to obtain more information of the environment in a single view. In particular omnidirectional vision can be useful in many applications such as estimating location in robotics, autonomous driving and unmanned aerial vehicles. The wide field of view (FOV) of omnidirectional cameras allows taking advantage of describing 3D scenarios using line features. On the one hand, line features represent natural landmarks in man-made environments, they are easy to understand, coincident with edges of constructive elements and often still present when having texture-less scenarios. On the other hand, long segments are especially useful for drift reduction because they are usually completely visible on the omnidirectional projection. However, in omnidirectional cameras line projections are distorted by the projection mapping becoming complex curves.

This thesis is focused on the geometry of line projections (line-images) in omnidirectional systems. Main addressed topic of this work is line-image extraction on different kinds of central and non-central omnidirectional images. However, due to nature of projection in omnidirectional cameras, other addressed topics are camera calibration and, in the case on non-central cameras, 3D reconstruction from single images.

In particular, we analyzed the following problems involving line projections in central and non-central omnidirectional cameras:

- **Line-image extraction in central omnidirectional cameras.** The presented research begins with line-image extraction from hypercatadioptric images which is used for estimating dominant directions in Manhattan scenarios. This extraction approach has been generalized to a framework for central imaging systems obtaining the equations of line-images curves for a set of different catadioptric and dioptric systems. Using this framework we have developed the plumb-line constraint for each of these systems and obtained analytical solutions that allow simultaneously recovering camera calibration and line-images. We have integrated these theoretical solutions in an automatic method for line-image extraction in central systems when the calibration is unknown. This proposal exploits location of image points and gradient direction on these points reducing the complexity of the extraction scheme.

- **Line-image extraction and 3D line fitting in non-central omnidirectional cameras.** The characteristics of the projection in non-central systems allow recovering the complete 3D information of a line from a single projection. The non-central systems addressed in this thesis are the conical and spherical catadioptric systems and the non-central circular panorama. Generalizing the description used for central systems we have developed the analytical line-image equations for conical and spherical catadioptric systems. The particularization for conical catadioptric systems allows us to recover the geometry of the mirror and the geometry of the 3D line from five points of the line-image. Main problem of recovering 3D from single projections in non-central systems is that results are very noise sensitive because four degrees of freedom are involved making line-image extraction a challenging unsolved task. To address this issue we propose different solutions and approaches. To select a set of rays in a robust framework and evaluate the accuracy we introduce the concept of effective baseline among rays which strongly depends on the class of non-central system. Regarding accuracy in line 3D fitting, the non-central panorama is the system obtaining the best results although its constructive difficulties usually prevents from a practical using. To solve the problem of automatic line-image extraction in non-central systems we propose a robust approach using a pre-evaluation step, similar to the one used in PROSAC, and using different kind of distances. In particular, we have analytically solved the Euclidean distance from a point to line-image for conical and spherical catadioptric systems and we propose a distance based on the projection of the closest points in the 3D space for the non-central circular panorama.

- **Minimal solutions for line-image fitting in non-central cameras by imposing geometric constraints.** Another way for increasing the accuracy in 3D line reconstruction is exploiting geometric constraints and using prior information. In this thesis, we propose a set of new minimal solutions for imposing additional constraints between pairs of lines in non-central systems. In particular, we have developed the cases of intersecting orthogonal lines and pairs of parallel lines. Finally we present the minimal solution for computing a line which is parallel to a given plane which can be exploited using the prior information of the vertical direction in a Manhattan scenario.

Contents

Chapter 1

Introduction

1.1 Computer vision

Computer vision is a multidisciplinary topic involving 3D geometry, statistics, artificial intelligence, optics and electronics. From the physical point of view a camera is composed by a light sensor at a given range of frequency with an optical system which enforces a particular projection forming a two-dimensional distribution of measures called image. From the geometrical point of view a camera is a projection system mapping a 3D world on a 2D surface.

The goal of artificial intelligence in computer vision is recognizing abstract elements on the image such as objects, categories or features which are related with the elements of the scene via the projection geometry. In this context, image processing is used for detecting invariant features which describe the image using a smaller but discriminant amount of information (LBP, HAAR [Viola and Jones, 2001], SIFT [Lowe, 2004], HOG [Dalal and Triggs, 2005] ,SURF [Bay et al., 2006], ORB [Rublee et al., 2011]). When the categorization and abstraction of the image is solved (for example by human intervention) the projection geometry allows using the camera as photogrammetric sensor. Geometric measures on the projected image are characterized by stochastic distributions which encode the uncertainty inherent to any abstraction when taking a measure.

In the projection mapping one of the three dimensions of the space is lost, meaning that it is not possible to recover a 3D measure directly from a single image. However, it is indirectly feasible when having a correspondence of elements from different images or when making some assumptions about the structure of the scene (for example: assuming the scene is composed by planes along main directions following the Manhattan World assumption [López-Nicolás et al., 2014]). Multiple views of a projected element can be achieved by moving a camera or using multiple cameras. The correspondence of features or matching is estimated using the abstraction level tackled by

artificial intelligence.

1.1.1 Motivation

Classical applications of computer vision in industry (machine vision) have been focused on visual inspection for quality control and robotic arm guidance. Controlled illumination conditions and environment knowledge are used for guaranteeing the required performance. Another extended application for industry is 3D scanning of pieces using structured light where the matching problem is solved by projecting light features. Identification of products labels for logistics can be achieved using optical character recognition (OCR) although in practice, for reducing the complexity and increasing the precision in recognition the using of bar codes or QR codes is usual.

Nowadays, computer vision has an increasing importance in many other branches. Emerging information technology companies, have made intensive use of visual recognition for image searching in photographs where the environment and illumination conditions are unknown. Other application taking advantage of visual recognition is face recognition. The proposals adopted by the Computer Vision community for visual recognition during the last ten years have been based in supervised machine learning approaches, where a classifier is trained using a set of labelled examples. Two things have been determinant for reaching the impressive results of the last times. First is the using of huge amounts of training examples [Torralba et al., 2008] following the Big Data paradigm. Second is the irruption of Deep Learning in computer vision in [Krizhevsky et al., 2012]. In particular the exploiting of Convolutional Neural Networks beating the state of the art that was based on support vector machines(SVM) classifiers in combination with Big Data has completely changed the paradigm in visual recognition and computer vision community.

In mobile robotics 3D vision is used for localizing the robot improving the odometry, making maps and recognizing objects for avoiding obstacles. In particular, simultaneous localization and mapping (SLAM) takes advantage of Bayesian filters [Davison et al., 2007] by propagating uncertainties of landmarks in sequences of images simultaneously reconstructing a map of landmarks and the location of the robot. The estimated map can be improved using loop closing [Kümmerle et al., 2011] and bundle adjustment. Instead of discrete landmarks the map can densely describe the environments using surfaces [Newcombe et al., 2011]. Similar approaches [Klein and Murray, 2007] using discrete landmarks or even human made markers are used for estimating the camera location in augmented reality applications. Autonomous aerial vehicles [Mahony et al., 2012] also can take advantage of computer vision [Meier et al., 2012] and in particular of SLAM approaches by integrating the dynamic model of the vehicle in the Bayesian filter [Achtelik et al., 2011].

Inspired by mobile robotics, computer vision is getting more popular in the automobile industry. Current cars already include a set of sensors and front cameras for

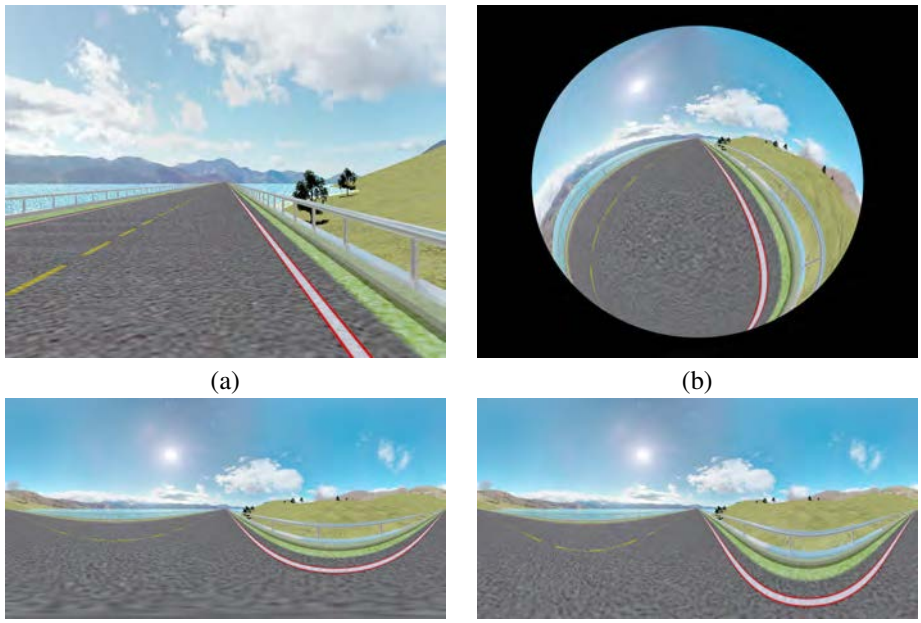
driving assistance. Complementing the cruise control, new cars[1] include a vision based lane detection system allowing the car autonomously follow the lane under human supervision. Other features like collision avoidance are performed using radar although the state of the art in pedestrian detection and car detection also allows using computer vision [Ros et al., 2015]. Autonomous driving[2] involves more computer vision challenges including vehicle, pedestrian, traffic light and signal detection [Carrasco et al., 2012]. The work presented by Daimler in [Ziegler et al., 2014] describes a complete autonomous driving combining GPS, radar and computer vision following the historical route of one of the first automobiles of the history. The challenge was possible using a detailed map of the route involving location of signals and traffic lights. The map for autonomous driving can be generated simultaneously when exploring the environment but the system usually takes advantage of more detailed maps acquired under human supervision. For example, Google has deployed dedicated cars[3] in routes and cities along the word for its maps street-view project. These cars carry Laser Imaging Detection and Ranging (LIDAR) systems for 3D acquisitions but the main sensor is the omnidirectional camera which allows taking panoramic images around a spherical view.

In this context, the using of omnidirectional cameras takes advantage of the high field of view for most of computer vision topics (see Fig. 1.1 and Fig. 1.2). Since the complete scene is observed, most features of a pair of images taken in the same position are visible in both, which is particularly useful for place recognition. The visibility of the whole scene is also valuable for visual odometry where features are seen for longer in sequences of omnidirectional images. In particular, occlusions are reduced when tracking longer features improving the pose estimation.

These advantages are even more relevant when dealing with line based features. Lines represent natural landmarks in man-made environments. They define the edges of constructive elements being particularly useful when having texture-less scenarios. For example, straight lanes in roads and walls in most indoor scenarios can be described by lines. Besides, because of a single line segment is defined along the 3D space of the scenario, the persistence of a line feature along the sequence is longer than a point feature, especially if the camera moves along the direction defined by the line. In addition, occlusion due to lack of field of view is reduced in omnidirectional images. More visibility of line features in omnidirectional images and persistence of lines along sequences allows reducing drift in 3D estimation.

Nevertheless, due to high distortions induced by omnidirectional imaging systems, lines are projected on curves making line related issues more difficult to solve. Open problems related with line features in omnidirectional vision are in modelling, extraction, calibration and 3D reconstruction and most of them are related with understanding the underlying geometry behind the line projection.

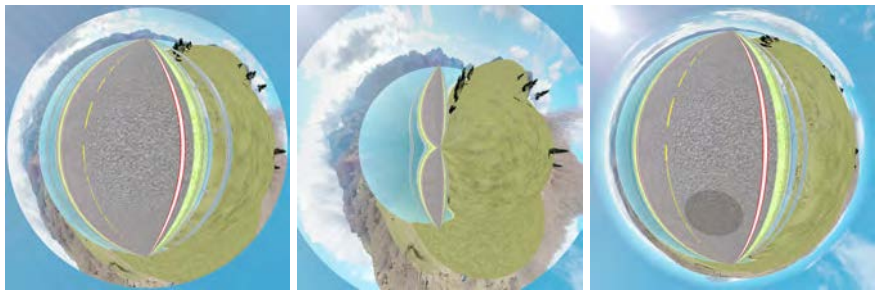
[1] $http://www.teslamotors.com/blog/your-autopilot-has-arrived$
[2] $http://www.nytimes.com/2010/10/10/science/10google.html$
[3] $http://www.google.com/maps/streetview/understand/$



(a) (b)

(c) (d)

Figure 1.1: Lane detection in different kinds of omnidirectional images. (a) Central perspective image. (b) Central fisheye. (c) Central spherical panorama. (d) Non-central circular panorama.



(a) (b) (c)

Figure 1.2: Lane detection in different kinds of catadioptric images. (a) Central hyper-catadioptric. (b) Non-Central conical-catadioptric. (c) Non-central spherical-catadioptric.

Main drawbacks in omnidirectional vision are because the imaging projection induces non-linearity in computer vision algorithms increasing the complexity, the locality of linear approximations and the computational time. However, as we show in this thesis, when dealing with line projections in omnidirectional cameras some of these drawbacks can be turned into advantages.

For example, distortion on curves representing the line projections makes extraction harder, on the other hand it embeds the calibration of the camera that can be recovered by fitting line-images. This transformation from drawback into advantage is even more evident with non-central imaging systems. In these systems, projecting rays do not pass through a unique optical center. That means that sometimes there is not analytical expression for the forward projection model and each projecting ray is defined by two points and some of the axioms assumed in computer vision must be reconsidered. In particular, from a single projection of a line on a single image it is possible to recover the complete geometry of the original 3D line including the depth and scale because the projection surface is not a plane and the projected curve embed the four degrees of freedom of the 3D line.

This property is very promising for single-image 3D reconstruction. However, because of its complexity, line-extraction in non-central images is an unsolved challenging problem.

1.1.2 Unconventional cameras

Conventional cameras have imitated the human vision which is almost perspective, central and binocular. However, in last decades new kind of different visual systems have been developed trying to exploit new ways of understanding the environment. One example among these non-conventional cameras is the plenoptic camera [Lumsdaine and Georgiev, 2009,Ng et al., 2005,Adelson and Wang, 1992] which is able to measure not only the amount of light received by each pixel but the light distribution depending on the incident direction. Other example is the event camera [Serrano-Gotarredona et al., 2009] which only measures asynchronous variations in local intensity performing faster measures. Finally, the set of different omnidirectional cameras which are optical instruments designed to acquire images with a wide field of view. This advantage allows them to include the whole scene in a single view. Some applications of these kinds of systems are surveillance, tracking, localization, visual navigation, structure from motion or SLAM. Omnidirectional cameras can be classified following different criteria:

Central and non-central systems There exist several types of omnidirectional cameras which can be classified as central and non-central. The central systems are those which satisfy that all the projecting rays intersect a common viewpoint. By contrast, in non-central systems the projecting rays do not intersect a common viewpoint. In this case, the locus of the viewpoint is in general tangent to a caustic which is an envelope

surface of the projecting rays.

Constructive classification Depending on the constructive set-up omnidirectional cameras can be classified in several classes:

- Catadioptric systems combine a conventional camera with mirrors and lenses. In [Baker and Nayar, 1999] is proven that the elliptic, parabolic and hyperbolic mirrors, combined with conventional cameras, are the only ones ensuring the single viewpoint property when the mirror is appropriately placed with respect to the camera.
- Dioptric systems use wide-angle lenses such as fisheye lenses combined with conventional cameras. Fisheyes are non-central by definition, but they can be considered as central systems due to the tangent to the caustic being close to a point.
- In moving cameras the resulting images are composed from the streaming of line cameras. Two examples are the rotating camera and the pushbroom camera. These kind of systems are non-central by definition because the optical center is not unique in the image composition.
- Multi-camera systems stitch images obtained from a set of cameras forming a panoramic image. Depending on the disposition of the cameras the system can be considered central or non-central.

1.1.3 Line-images

The interest of the geometry behind the projections of 3D lines is that line-image features represent natural landmarks of the environment. In man-made environments they define the edges of constructive elements and they are particularly useful when having texture-less scenarios. Main disadvantages of line-images as features are the lack of robust descriptors for matching in comparison with point features and the fact that line extraction is unstable because the supporting points of a given line-image can vary a lot from one image to another.

A straight 3D line is a geometric arrange of 3D points having four degrees of freedom (DOFs) which is defined by the span of two 3D points or the intersection of two planes. Considering projective geometry for describing the geometry of a scene, the set of straight 3D lines forms a four-dimensional manifold induced by the underlying geometry of the projective space $\mathbb{P}^3$ (See Appendix A). The projection of a straight 3D line on an image is a curve called line-image or line projection. The surface defined by the set of projecting rays passing through the straight 3D line and the line-image is known as projection surface.

The projection surface of a 3D line in central systems is a plane passing through the 3D line and the single viewpoint of the camera. In this projection, part of the

information of the 3D line is lost and only two of the four degrees of freedom (DOF) of the 3D line are preserved. All lines lying on this plane share the same line-image or in other words, a 3D line occludes any other line located behind. In perspective central systems the line-image is a straight 2D line on the image plane. When the central system is not perspective, the line-image is a curve with more than two DOFs due to the non-linearity of the projection. The additional DOFs of these curves are deeply related with the parameters of the projection model and the calibration of the imaging system.

The corollaries of these properties for central systems are the following:

- The projection plane in a central projection can be recovered from two projecting rays intersecting the line-image. When the calibration of a central system is known these two rays can be obtained from two points of the line-image meaning that the line-image can be fitted from at least two points.

- If the system is not calibrated we still know that the rays intersecting the line image must form a plane and this constraint can be used for estimating the projection plane and the calibration of the system using additional points lying on the line-image. This constraint is known in the literature as *plumbline constraint* [Devernay and Faugeras, 2001].

- From a single line-image the unique 3D structure we can recover is the projection plane, hence the only ways to recover the direction and the depth up to scale of the original 3D line are:

  - knowing the correspondence between different line-images from different views taken from different locations,
  - imposing additional constraints like dominant directions or perpendicularity.

In contrast with central systems, the projection surface of a straight 3D line in a non-central system is in general a surface containing the 4 DOFs of the original 3D line. Therefore, when the system is calibrated, the geometry of a straight 3D line can be recovered from a single line-image. In particular, four points on a line-image (except some degenerate cases) define four projecting rays providing four independent constraints for computing the complete geometry of the line. This can be intuitively understood as it is not possible to occlude a 3D line with other 3D line when the projection surface is not a plane or a regulus. The quality of the reconstructed 3D line depends on a magnitude called *effective-baseline of a set of rays* which is related with the distances between the four defining skew rays.

In non-central systems a line-image has in general more than 4 DOFs. As in central systems, the additional DOFs encode the parameters of the projection model and can be used for estimating the calibration of the system.

1.2 State of the art

Main challenge of working with omnidirectional cameras is that the high distorted images require additional geometry than the well known linear geometry used in perspective and conventional cameras. Historically, the community has researched for omnidirectional systems having the single-view property. Main reason is the similarity with conventional cameras which allows to use most of the classical 3D algorithms just embedding the corresponding projection model. Another reason is that having centrality guarantees a closed form for the forward-projection model and finally, often central images are easier to understand for the human cognition.

Projection models in central systems. The geometry of central catadioptric systems are extensively studied in [Baker and Nayar, 1999] proving that systems using elliptic, parabolic and hyperbolic mirrors fulfil the single viewpoint property when some configuration restriction are satisfied. Several projection models are proposed for dealing with central catadioptric cameras (See [Sturm et al., 2011]). [Svoboda and Pajdla, 2002] proposes models for different mirrors and study the epipolar geometry. The physical reflection on the mirror can be directly modelled like in [Strelow et al., 2001]. The projection model of a paracatadioptric system using a parabolic mirror and an orthographic camera follows the classical stereographic projection which maps a sphere onto a plane. Extending this idea, a unified model which is equivalent to the physical reflection but considering projections on the unitary sphere instead is proposed by [Geyer and Daniilidis, 2000] using the same parameters for different classes of catadioptric systems. This model was extended by [Barreto and Araujo, 2001] and [Ying and Hu, 2004b].

Fisheye cameras are intrinsically non-central. However, fisheye lenses are manufactured for reducing the effective baseline between rays hence in practice they have a central behaviour [Sturm et al., 2011]. Actually, fisheye lenses are constructed in order to satisfy any of the classical fisheye models: the equiangular-fisheye model (also known as equidistant projection), the equisolid-fisheye model, the stereographic-fisheye model and the orthographic fisheye model [Kingslake, 1989, Stevenson and Fleck, 1996, Ray, 2002]. A validation of these geometric models for real fisheyes is presented in [Schneider et al., 2009].

Assuming symmetry of revolution, these models are encoded using a function h that relates the elevation angle ϕ of the projection ray with the radius of an image point r. The function $r = h(\phi)$ can be an explicit expression [Kingslake, 1989, Ray, 2002] or can be defined as a series-based expression [Tardif et al., 2006, Kannala and Brandt, 2006, Kannala et al., 2008b]. There exist other empirical models valid for any central system that instead of defining series using the radius, use the elevation angle [Scaramuzza et al., 2006].

Other approach for modelling the central behaviour of a fisheye lens is using radial distortion models based on a transformation of the radius of an image point after a

linear projection. Some of these models are the *even order polynomial model* used in [Swaminathan and Nayar, 2000, Devernay and Faugeras, 2001, Thormählen et al., 2003, Rosten and Loveland, 2011] and the *division model* proposed by Fitzgibbon in [Fitzgibbon, 2001] which is extensively used in [Strand and Hayman, 2005, Wang et al., 2009, Bukhari and Dailey, 2013, Melo et al., 2013]. These models are usually used to describe radial distortion on conventional cameras and they can also be used to fit catadioptric and dioptric systems.

Line extraction in central systems. Most approaches to extract lines from omnidirectional central images use camera calibration to back project the image points to normalized rays lying on a unitary sphere around the viewpoint [Ying and Hu, 2004c, Bazin et al., 2010]. In this space the representation of the line projection becomes linear and classical approaches can be used. In the particular case of central catadioptric systems the curve defining a line-image is a conic. Some approaches extract these conics directly on the images. In [Barreto and Araujo, 2005] conics are computed using classical conic fitting approaches, hence 5 DOFs are required in this fitting which is noise sensitive. In [Ying and Zha, 2005, Cucchiara et al., 2003] a Hough transform approach is used to simultaneously extract lines and calibration parameters from uncalibrated omnidirectional images.

As explained before, in central systems the projection plane provides a constraint which can be exploited to estimate the calibration whenever more than two independent constraints are available [Devernay and Faugeras, 2001, Alvarez et al., 2009, Brown, 1971]. This constraint is known as plumb-line constraint [Sturm et al., 2011] and it is one of the approaches used for self-calibration in central distorted images. The location of additional edge points lying on the line-image provides additional independent constraints allowing fitting the line-image and obtaining the projection plane and the calibration.

Apart from the plumb-line approach there exist other methods for calibrating omnidirectional central cameras. A survey of these methods can be found in [Puig et al., 2012]. We want to remark that, in spite of existing a considerable number of methods for calibrating catadioptric systems [Wu and Hu, 2005, Barreto and Araujo, 2005, Gasparini et al., 2009, Puig et al., 2011] there are few methods for fisheye systems. A general method for central systems that can be also used for fisheyes is presented in [Scaramuzza et al., 2006]. In [Mei and Rives, 2007] the sphere model is used to approximate the fisheye model. A calibration method specific for fisheyes is presented in [Kannala and Brandt, 2006, Kannala et al., 2008a].This multi-view approach uses a series-based description of the radially symmetric model. The radially symmetric series-based model is also used in a plumb-line calibration method for central systems in [Tardif et al., 2006]. More recently, in [Alemán-Flores et al., 2014] line projections are detected in uncalibrated distorted images using a Hough transform approach.

Projection models in non-central systems. The geometry of non-central systems has previously been studied for most of catadioptric and non catadioptric systems. The back projection of different non-central catadioptric systems is introduced in [Swaminathan et al., 2006]. Contrary to the back projection case, the forward projection of non-central systems does not necessarily have a closed form solution. In [Agrawal et al., 2010] Agrawal et al. present a polynomial solution for the forward projection in axial non-central catadioptric systems based on a quadric mirror which is extended for the non axial case in [Agrawal et al., 2011]. These approaches are used in a calibration method for multi-axial non-central systems in [Agrawal and Ramalingam, 2013]. In [López-Nicolás and Sagüés, 2014] the epipolar geometry of conical catadioptric systems and its corresponding calibration are presented.

Another way to obtain non-central images is using images generated from moving cameras without common viewpoint. The epipolar geometry is studied for the case of linear pushbroom camera in [Gupta and Hartley, 1997] and for the case of non-central circular panorama in [Menem and Pajdla, 2004]. A multi-camera system can be also tackled as particular case of a non-central system. In [Lee et al., 2015] Lee et al. propose a new method to solve the pose estimation in a multi-camera system represented by a set of ray bundles in a non-central description using Plücker coordinates.

Line-extraction in non-central systems. Some previous approaches have tried to extract 3D lines from a single image in non-central catadioptric systems. The fundamentals of this approach are exposed in [Teller and Hohmeyer, 1999, Kneebone et al., 1998, Griffiths and Harris, 2011] where is shown that two lines are the intersection of four generic lines. In [Caglioti and Gasparini, 2005, Caglioti et al., 2007a, Gasparini and Caglioti, 2011] this approach is exploited to compute 3D lines from 4 rays in non-central catadioptric systems. Different computation methods are considered and the degeneracies and singular configurations are studied. In [Lanman et al., 2006] the approach is used with spherical catadioptric mirrors, and in addition two non-central systems are used for reconstruction. Work in [Swaminathan et al., 2008] extends the approach to planar curves. To improve the accuracy in reconstruction using catadioptric systems some approaches have been proposed: considering only horizontal lines [Pinciroli et al., 2005, Chen et al., 2011] , exploiting cross-ratio properties [Perdigoto and Araujo, 2012] . Using off-axis systems [Caglioti et al., 2007b] allows avoiding the degeneracies caused by the revolution symmetry. As application, the pose of non-central catadioptric systems is estimated in an image sequence [Miraldo and Araujo, 2014, Miraldo et al., 2015] using known 3D lines.

Line projections have been also used to estimate the calibration of non-central systems in a generalization of the plumb-line approach. In [Caglioti et al., 2007b] non-central calibration using lines is studied. They exploit the fact that there exists less ambiguity when the system is off-axis with impressive results. However, this work does not exploit the particular geometric description of the line-images. In [Agrawal and Ramalingam, 2013] they exploit particular geometric properties of spherical mir-

rors for computing extrinsic calibration parameters but they do not use lines.

1.3 Contributions

In this thesis we focus on the geometry of line projections or line-images in different kinds of central and non-central omnidirectional cameras. Main topics covered during the development of this thesis cover: projection models of line-images, line-extraction, calibration and 3D reconstruction. The most relevant contributions of the present thesis can be summarized as follows:

Line extraction in central and calibrated cameras. We propose a new method for extracting line-images in hypercatadioptric systems which is extended to different classes of central catadioptric and dioptric systems. For the particular case of the hypercatadioptric systems the description of the vanishing points on the image are presented and used for estimating dominant directions in Manhattan scenarios and rectifying the original image.

Associated publications:

- [Bermudez-Cameo et al., 2012b] J. Bermudez-Cameo, L. Puig, J.J. Guerrero, "Hypercatadioptric Line Images for 3D Orientation and Image Rectification", Robotics and Autonomous Systems, 60, pp: 755-768, 2012.
- [Bermudez-Cameo et al., 2012a] J. Bermudez-Cameo, G. Lopez-Nicolas and J. J. Guerrero, "A Unified Framework for Line Extraction in Dioptric and Catadioptric Cameras", 11th Asian Conference on Computer Vision, ACCV, Daejeon, Korea, Nov. 2012. In Computer Vision - ACCV 2012, volume 7727 of Lecture Notes in Computer Science, pp: 627-639.
- [Bermudez et al., 2010] J. Bermudez, L. Puig, J.J. Guerrero, "Line extraction in central hyper-catadioptric systems", In 10th OMNIVIS Omnidirectional Vision, Camera Networks and Non-classical Cameras, with RSS 2010, pp:8, 1-7, June 2010.

Uncalibrated line extraction in central cameras. We present a new proposal for the automatic simultaneous extraction of line-images and the main calibration parameter from uncalibrated omnidirectional images. The method is particularized for different catadioptric and dioptric systems obtaining closed solutions for most of them. The proposal also study the using of gradient information to reduce the complexity of the robust extraction. The proposal is compiled in a fully functional toolbox [4] which has been released as open source for research purposes.

Associated publications:

[4] webdiis.unizar.es/%7Ebermudez/toolbox.html

- [Bermudez-Cameo et al., 2015c] J. Bermudez-Cameo, G. Lopez-Nicolas and J.J. Guerrero, "Automatic Line Extraction in Uncalibrated Omnidirectional Cameras with Revolution Symmetry", International Journal of Computer Vision. 114(1) pp:16-37, 2015.

- [Bermudez-Cameo et al., 2013] J. Bermudez-Cameo, G. Lopez-Nicolas and J. J. Guerrero, "Line extraction in uncalibrated central images with revolution symmetry", BMVC 2013 - British Machine Vision Conference , Bristol, United Kindom, 9-13 Sept 2013.

Line extraction in non-central cameras. The geometric development of line projections in different classes of non-central imaging systems. In particular, we present the expression of the line-image for the conical catadioptric system, the spherical catadioptric system and the non-central circular panorama. Using the proposed models we present an automatic method for line-image extraction from non-central images obtaining the complete 3D geometry of the lines from a single projection on a single image. Up to our knowledge this is the first work addressing automatic line-extraction in non-central images. We also present the concept of *effective baseline of a set of rays* describing the suitability of a 3D line fitting from a set of rays, and a feature for measuring this *effective baseline*. For achieving the non-central extraction we propose a RANSAC based method using the effective baseline measure for filtering the candidates. We also define a set of distances from point to line developed from the proposed line-image descriptions. These distances are: an algebraic distance measured in pixels based on the general proposal for line-images when having revolution symmetry, an Euclidean distance developed from the quartic description of a line-image in conical and spherical catadioptric systems and finally, a distance based on projecting the closest point from ray to line. The later proposed distance has been used in non-central circular panoramas.

Associated publications:

- [Bermudez-Cameo et al., 2014b] J. Bermudez-Cameo, G. Lopez-Nicolas and J. J. Guerrero, "Line-images in Cone Mirror Catadioptric Systems", ICPR 2014 - 22nd International Conference on Pattern Recognition, pp: 2083-2088, Stockholm, Sweden 24-28 August 2014.

- [Bermudez-Cameo et al., 2015a] J. Bermudez-Cameo, G. Lopez-Nicolas and J. J. Guerrero, "Fitting Line Projections in Non-Central Catadioptric Omnidirectional Systems" (Submitted as a journal paper), 2015.

- [Bermudez-Cameo et al., 2015d] J. Bermudez-Cameo, O. Saurer, G. Lopez-Nicolas, J. J. Guerrero and M. Pollefeys, "Exploiting line metric reconstruction from non-central circular panoramas" (Technical Report), 2015.

Minimal solutions for line-fitting imposing constraints in non-central cameras. We present analytical solutions in the form of roots of polynomials for solving line-image fitting in non-central systems when imposing additional geometric constraints. In particular we present the analytical solutions for: two parallel lines when having three points lying on each line-image, two intersecting orthogonal lines when having three points lying on each line-image, and the line parallel to a plane with known orientation when having three points lying on the line-image. Last constraint has been integrated in a line-image extraction method for non-central circular panorama systems exploiting Manhattan conditions and assuming the gravity direction is a prior information obtained from other sensors like an IMU.

Associated publications:

- [Bermudez-Cameo et al., 2014a] J. Bermudez-Cameo, J.P. Barreto , G. Lopez-Nicolas and J. J. Guerrero, "Minimal solution for computing pairs of lines in non-central cameras", ACCV 2014 - The 12th Asian Conference in Computer Vision, Singapore 1-5 November 2014.

- [Bermudez-Cameo et al., 2015b] J. Bermudez-Cameo, C. Demonceaux, G. Lopez-Nicolas and J. J. Guerrero, "Minimal solution for line parallel to a plane in non-central systems" (Technical Report), 2015 .

I also collaborated in a survey of techniques for calibrating omnidirectional cameras, with the following publications associated:

- [Puig et al., 2012] L. Puig , J Bermudez, P. Sturm, J.J. Guerrero, "Calibration of Omnidirectional Cameras in Practice: A Comparison of Methods", Computer Vision and Image Understanding, 116, pp:120-137, 2012.

- [Puig et al., 2010] L. Puig, J. Bermudez, J.J. Guerrero, "Self-orientation of a hand-held catadioptric system in man-made environments", 2010 IEEE International Conference on Robotics and Automation, ICRA, pp:2549-2555, Anchorage, Alaska May 2010.

1.4 Outline

The thesis is organized in two parts, the first part is dedicated to central imaging systems (Chapters 2 and 3) and the second focus on non-central imaging systems.

In Chapter 2 we focus on line-image extraction from hypercatadioptric images and its use for computing vanishing points and main directions in man-made environments. In Chapter 3 we present a framework for describing line-images in central systems with symmetry of revolution and integrate it in a method for automatically extracting line-images and calibration exploiting the plumbline constraint.

Second part of the thesis deals with line-images in non-central systems where it is possible to recover the 3D information of a line from a single image. Chapter 4

introduces representations of line-images for conical and spherical catadioptric systems and handles with line-image extraction in non-central catadioptric systems. In Chapter 5 we address line-image extraction from non-central panorama images and we explore different ways for acquiring these panoramas. Chapter 6 is dedicated to the minimal solutions for fitting pair of lines in non-central systems. The cases of two parallel lines and two intersecting orthogonal lines are studied. Finally, in Chapter 7 we exploit a minimal solution solving a line parallel to a plane to include prior information of gravity direction in Manhattan scenarios.

Along this thesis, for describing straight 3D lines we use Plücker coordinates (See Appendix A). In central systems the projection surface of a straight 3D line $\mathbf{L} = \left(\mathbf{l}^\mathsf{T}, \bar{\mathbf{l}}^\mathsf{T}\right)^\mathsf{T}$ is a plane passing through the 3D line and the single viewpoint and it is described by the normal $\mathbf{n}$ which is coincident with the moment of the straight 3D line $\bar{\mathbf{l}}$. When dealing with central systems (Chapters 2 and 3) the projection surface it is a plane and only the projection plane can be recovered from a single projection, therefore we use the normal $\mathbf{n}$ for describing this plane and omit the straight 3D line notation. By contrast, in Chapters 4-7 we address non-central systems where the concept of projection plane does not make sense (the projection surface is not a plane), hence we use the notation based on Plücker coordinates for the recovered straight 3D line. In general, the notation of each chapter is independent and self-explained.

Part I

Line-Images in central omnidirectional cameras

Chapter 2

Hypercatadioptric line-images for 3D orientation and rectification

First part of this thesis is dedicated to line-images in central systems. In this chapter we focus on central catadioptric systems where 3D lines are projected into conics. In particular, we present a new approach to extract these conics in raw hypercatadioptric images which correspond to projected straight lines in the scene. Using the internal calibration and two image points we are able to compute analytically these conics which we name hypercatadioptric line-images. We obtain the error propagation from the image points to the 3D line projection depending on the calibration parameters. We also perform an exhaustive analysis on the elements that can affect the conic extraction accuracy. Besides that, we exploit the presence of parallel lines in man-made environments to compute the dominant vanishing points (VPs) in the omnidirectional image. In order to obtain the intersection of two of these conics we analyze the self-polar triangle common to this pair. With the information contained in the vanishing points we are able to obtain the 3D orientation of the catadioptric system. This method can be used either in a vertical stabilization system required by autonomous navigation or to rectify images required in applications where the vertical orientation of the catadioptric system is assumed. We use synthetic and real images to test the proposed method. We evaluate the 3D orientation accuracy with a ground truth given by a goniometer and with an inertial measurement unit (IMU). We also test our approach performing vertical and full rectifications in sequences of real images.

2.1 Introduction

Omnidirectional cameras are devices designed to capture images with a wide field of view. This characteristic introduces a new approach in computer vision minimizing the possibility of fatal occlusions and helping the tracking of features. Among these cameras we find the catadioptric systems, which are a combination of a mirror and a camera. Some of these systems have the single viewpoint constraint. In [Baker and Nayar, 1999] an analysis of central catadioptric systems is presented describing which systems have the single viewpoint property. Among these we have the hypercatadioptric system which is composed of a hyperbolic mirror and a perspective camera. In mobile robotics when a catadioptric system is used it is commonly observed that it has a vertical orientation. This is because most robotic platforms used are wheel-based. Under this configuration planar-motion is assumed simplifying the localization algorithms. In applications where line tracking or line matching is performed [Mezouar et al., 2004, Guerrero et al., 2008, Scaramuzza et al., 2009] this assumption is useful since vertical lines are coplanar with the optical axis becoming straight radial lines in the catadioptric image. When this assumption is not satisfied, vertical lines are projected as conics in the catadioptric image. This situation requires the development of new algorithms capable to deal with these conic projections. Another advantage of the wide field of view in catadioptric systems is that visible segments of projected lines are very large. When we use catadioptric systems in man-made environments we can observe sets of parallel and orthogonal lines. These sets of lines encapsulate geometrical information which we exploit in this work. In particular, vanishing points contain the orientation of the camera with respect to the reference coordinate system defined by the main directions of the environment. However, dealing with line projections in central catadioptric images requires to solve the problem of extraction of conics. In general, five points are required to determine uniquely a conic. When the internal calibration of the central catadioptric system is known, only two points are needed to compute these conics, which we particularly call Hypercatadioptric Line Images (HLIs).

Line extraction in omnidirectional catadioptric systems has been studied previously. In [Barreto and Araujo, 2006a] an exhaustive analysis of line images in paracatadioptric cameras is presented for calibration purposes. In [Vasseur and Mouaddib, 2004], the space of the equivalent sphere which is the unified domain of central catadioptric sensors combined with the Hough transform is used. They also perform tests showing the influence of the calibration parameters in their approach. In [Ying and Hu, 2004c] Ying and Hu also use the Hough transform and two parameters on the unitary sphere to detect the image lines. The accuracy on the detection of these two approaches depends on the resolution of the Hough transform. The higher the accuracy, the more difficult it is to compute the lines. In [Mei and Malis, 2006] the randomized Hough transform is used to overcome the singularity present in [Vasseur and Mouaddib, 2004, Ying and Hu, 2004c], which speeds up the extraction of conics. This scheme is compared in converge mapping to a RANSAC approach. In [Bazin et al.,

2007, Bazin et al., 2010] an scheme of split and merge is proposed to extract the image lines present in a connected component. The connected components are computed in two steps. The first step consists of detecting the edges using the Canny operator. The second step is a process of chaining which builds the connected components.

A vanishing point (VP) is the point on the image where the projections of two or more parallel 3D lines intersect. This point is the projection of a virtual 3D point located at infinity. That means that coordinates of a VP on the image only changes when the camera is rotated. This property is exploited to obtain the orientation of the camera with respect to a global reference system. The relationship between the orientation of the camera and the vanishing points of perspective images has been deeply studied before [Hartley and Zisserman, 2000]. Due to the wide field of view of omnidirectional cameras VPs are commonly located inside the catadioptric image. For this reason VP-based techniques are interesting with omnidirectional cameras. In [Bazin et al., 2010] VPs are used with omnidirectional images to estimate the camera orientation. In [Bosse et al., 2002] vanishing points are used in omnidirectional sequences as orientation feature in an Extended Kalman Filter. A method of expectation maximization is used in [Antone and Teller, 2000] to classify and detect VPs from edges on the Gaussian Sphere.

In this chapter we present a method to extract HLIs in the raw image in contrast to the approaches presented in [Vasseur and Mouaddib, 2004, Ying and Hu, 2004c, Mei and Malis, 2006, Bazin et al., 2007] which execute the fitting on the unitary sphere. Our approach does not use the Hough transform so the computing of the HLI is made directly from two image points. Extracting the HLIs directly on the image, means that thresholds and fitting parameters are expressed in pixel units. An example of this is the metric used to decide if a point belongs to a conic. Instead of using an algebraic distance we propose an approximation to the geometrical distance on the image. A RANSAC voting scheme is used to extract all the HLIs present in a single connected component. Contrary to [Bazin et al., 2007], the extraction of one HLI is performed in one step. We also perform an exhaustive analysis on the elements that can affect the HLIs extraction accuracy. Since we work on the image it allows measuring the influence and behaviour caused by calibration errors in pixels. This study is applicable to single HLIs in opposition to the study performed in [Vasseur and Mouaddib, 2004] which shows the influence of calibration parameters in a collection of projected lines. The way the results are presented in [Vasseur and Mouaddib, 2004] is conditioned by the Hough transform approach. To complete the analysis we present the propagation error formulas from calibration parameters to the spatial representation of the 3D line projection. Analogous formulation and analysis is performed with respect to the HLI defining points on the image.

Once lines in the catadioptric images have been extracted, vanishing points are estimated by computing the intersection of parallel lines. We propose a modification to the computation of the common self-polar triangle [Barreto, 2003] to compute the intersection between a pair of HLIs. The intersection of two HLIs is a particular case in

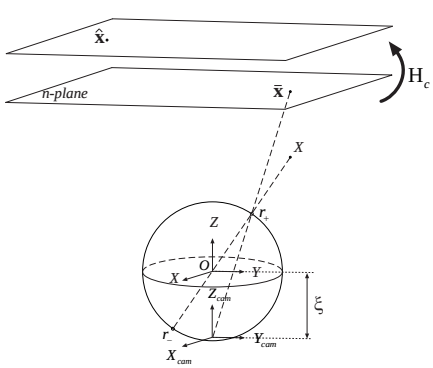


Figure 2.1: Sphere camera model.

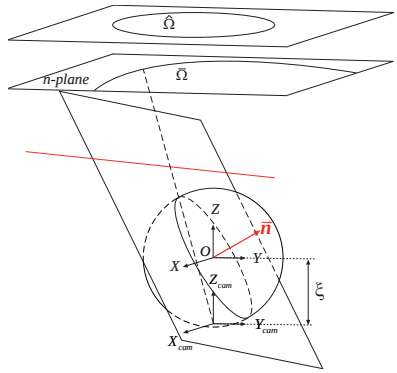


Figure 2.2: Projection of a line under the sphere camera model.

which the number of intersections between conics is reduced from four to two points. When this intersection corresponds to images of parallel lines these points are the vanishing points. We compute all the intersections between the HLIs present in the image. Then with a voting approach we robustly determine which ones are the vanishing points. The Vertical Vanishing Point (VVP) allows us to determine two of the three orientation angles with respect to an absolute reference. Using an analogous process we compute the horizontal vanishing points (HVP) and as consequence the full orientation of the catadioptric system. This orientation can be used to perform a rectification of the omnidirectional image.

The rest of the chapter is distributed as follows. In section 2.2 we describe the unified theory to project points and lines in catadioptric systems [Geyer and Daniilidis, 2000]. In section 2.3 we present a new approach to describe conics which are projections of lines with two points and we show an algorithm to extract them from a hypercatadioptric image. Section 2.4 presents an analysis of the influence of calibration on the extraction procedure and an analysis of the uncertainty propagation in the HLI computation. Section 2.5 describes how to use these conics to estimate vanishing points assuming dominant directions, in order to compute the orientation of the camera. Some experiments are presented in section 2.6 to validate the proposed method. Finally, we present the discussion in section 2.7.

2.2 Projections in central catadioptric systems

Under the sphere camera model [Geyer and Daniilidis, 2000] all central catadioptric systems can be modelled by a projection to the unitary sphere followed by a perspective projection. The projection of a 3D point $\mathbf{X} = (X\ Y\ Z\ 1)^{\mathsf{T}}$ into an omnidirectional

image point $\hat{\mathbf{x}}$ can be performed as follows (Fig. 2.1). First, the 3D point is associated with a ray $\mathbf{x}$ in the mirror reference system. This is done by P, a conventional projection matrix, $\mathbf{x} = \mathsf{P}\mathbf{X}$. We assume the world reference system and the mirror reference system are the same $\mathsf{P} = [\mathtt{I}|\mathbf{0}]$. Second, the 3D ray is projected onto the sphere passing through its center and intersecting in two points $\mathbf{r}_{\pm}$. These points are then projected into an intermediate perspective plane with focal length equal to one, giving the points $\bar{\mathbf{x}}_{\pm}$, one of which is physically true. This step is encoded in the function $\hbar$ (2.1). The last step is the projection of these points into the omnidirectional image, which is performed by a collineation H_c [Barreto and Araujo, 2005]. Matrix H_c is the combination of the intrinsic parameters of the perspective camera K_c, the rotation between the camera and the mirror R_c assumed the identity $\mathtt{I}$ and the shape of the mirror M_c. This model considers all central catadioptric cameras, encoded by ξ, which is the distance between the center of the perspective projection and the center of the sphere. $\xi = 0$ for perspective, $\xi = 1$ for paracatadioptric and $0 < \xi < 1$ for hypercatadioptric.

$$\hbar(\mathbf{x}) = \begin{pmatrix} x \\ y \\ z + \xi\sqrt{x^2+y^2+z^2} \end{pmatrix} \tag{2.1}$$

$$\mathsf{H}_c = \underbrace{\begin{pmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix}}_{\mathsf{K}_c} \underbrace{\begin{pmatrix} \psi-\xi & 0 & 0 \\ 0 & \xi-\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\mathsf{M}_c} \tag{2.2}$$

Let $\Pi = (n_x, n_y, n_z, 0)^{\mathsf{T}}$ be a plane defined by a 3D line and the effective viewpoint in the sphere camera model $\mathbf{O}$ (see Fig. 2.2). The projected line $\mathbf{n}$ associated to the 3D line by P can be represented as $\mathbf{n} = (n_x, n_y, n_z)^{\mathsf{T}}$. Then, the points $\mathbf{X}$ lying in the 3D line are projected to points $\mathbf{x}$. These points satisfy $\mathbf{n}^{\mathsf{T}}\mathbf{x} = 0$ and $\mathbf{x} = \hbar^{-1}(\bar{\mathbf{x}})$, so $\mathbf{n}^{\mathsf{T}}\hbar^{-1}(\bar{\mathbf{x}}) = 0$. As in [Barreto and Araujo, 2005], this equality can be written as

$$\bar{\mathbf{x}}^{\mathsf{T}}\bar{\Omega}\bar{\mathbf{x}} = 0 \tag{2.3}$$

where the image conic is

$$\bar{\Omega} = \begin{pmatrix} n_x^2(1-\xi^2) - n_z^2\xi^2 & n_x n_y(1-\xi^2) & n_x n_z \\ n_x n_y(1-\xi^2) & n_y^2(1-\xi^2) - n_z^2\xi^2 & n_y n_z \\ n_x n_z & n_y n_z & n_z^2 \end{pmatrix} \tag{2.4}$$

and the image of the conic in the catadioptric image is

$$\hat{\Omega} = \mathsf{H}_c^{-\mathsf{T}}\bar{\Omega}\mathsf{H}_c^{-1}. \tag{2.5}$$

Notice that $\bar{\Omega}$ is a degenerate conic when the 3D line is coplanar with the optical axis [Barreto and Araujo, 2005].

2.3 Computation of hypercatadioptric line-images

In this section we present the process used to extract HLIs contained in hypercatadioptric images. First we show the analytical process, which requires only two image points and the mirror parameter. Then we introduce the proposed point-to-conic metric used to decide if a point lies on a conic. Then the whole extraction process is depicted as pseudo-code. Finally we show a comparison of the five-point algorithm with our approach, which distinguishes between conics coming from straight lines and from other projected conics.

2.3.1 Conic definition using two points

Here we explain the method used to extract the HLIs from two image points. As mentioned before, in the case of uncalibrated systems we require five points to compute a conic. If these points are not well distributed in the whole conic, the estimation is difficult and usually inaccurate. Another disadvantage of a 5-point approach is the number of parameters. When a robust technique like RANSAC is used, this is quite important, because the number of iterations required considerably increases with respect to the number of parameters of the model. Our approach overcomes these problems requiring only two points and the calibration parameters. First we transform the points to the normalized plane $\bar{\mathbf{x}} = (\bar{x}\ \bar{y}\ \bar{z})^{\mathsf{T}}$ using the inverse of matrix H_c

$$\bar{\mathbf{x}} = \mathsf{H}_c^{-1}\hat{\mathbf{x}}. \tag{2.6}$$

Developing (2.3) and after some algebraic manipulation we obtain

$$\left(1-\xi^2\right)\left(n_x\bar{x}+n_y\bar{y}\right)^2 + 2n_z\bar{z}\left(n_x\bar{x}+n_y\bar{y}\right) + n_z^2\left(\bar{z}^2 - \xi^2\left(\bar{x}^2+\bar{y}^2\right)\right) = 0 \tag{2.7}$$

simplifying

$$(1-\xi^2)\alpha^2 + 2\alpha + \left(\bar{z}^2 - \xi^2\left(\bar{x}^2+\bar{y}^2\right)\right) = 0 \tag{2.8}$$

where a change of variable to $\alpha = \dfrac{n_x\bar{x}+n_y\bar{y}}{n_z}$ is performed.

We can compute α by solving the quadratic equation

$$\alpha = -\frac{\bar{z}}{1-\xi^2} \pm \frac{\xi}{1-\xi^2}\sqrt{\bar{z}^2 + \left(\bar{x}^2+\bar{y}^2\right)\left(1-\xi^2\right)} \tag{2.9}$$

Once we have solved this quadratic equation we can compute the normal $\mathbf{n}$. Consider two points in the normalized plane $\bar{\mathbf{x}}_1 = (\bar{x}_1, \bar{y}_1, \bar{z}_1)^{\mathsf{T}}$ and $\bar{\mathbf{x}}_2 = (\bar{x}_2, \bar{y}_2, \bar{z}_2)^{\mathsf{T}}$ lying on the conic we want to extract. From (2.9) we compute the corresponding α_1 and α_2.

Notice that there exist two solutions for α and just one has a physical meaning[1]. Using these parameters we obtain the linear system

$$\begin{pmatrix} \bar{x}_1 & \bar{y}_1 & -\alpha_1 \\ \bar{x}_2 & \bar{y}_2 & -\alpha_2 \end{pmatrix} \begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{2.10}$$

Solving for n_x, n_y and n_z we have

$$n_x = \bar{y}_1\alpha_2 - \bar{y}_2\alpha_1 \qquad n_y = \bar{x}_2\alpha_1 - \bar{x}_1\alpha_2 \quad \text{and} \quad n_z = \bar{x}_2\bar{y}_1 - \bar{x}_1\bar{y}_2. \tag{2.11}$$

Notice that we have analytically obtained the normal $\mathbf{n}$ that defines the projection plane of the 3D line, therefore the conic $\bar{\Omega}$ described in (2.4).

2.3.2 Distance from a point to a conic

In contrast to previous works [Vasseur and Mouaddib, 2004, Ying and Hu, 2004c, Mei and Malis, 2006, Bazin et al., 2007] that deal with points in the unitary sphere, we work in the image plane where conics are computed. In order to know if a point $\mathbf{x}$ lies on a conic Ω, represented as a 6-vector $\mathbf{c} = (c_1, c_2, c_3, c_4, c_5, c_6)^\mathsf{T}$, we need to compute the distance from a point to a conic. Two distances are commonly used to this purpose. An algebraic distance defined by (2.12) which just gives a value without clear geometrical meaning

$$d_{alg} = c_1x^2 + c_2xy + c_3y^2 + c_4x + c_5y + c_6. \tag{2.12}$$

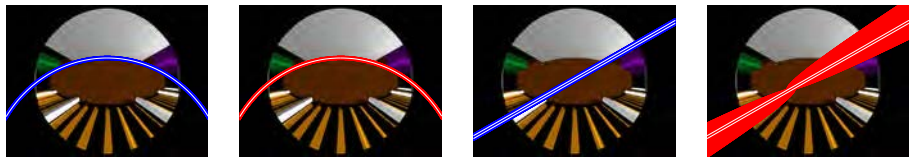
The other gives the geometric distance from this point to the closest point on the conic. The geometric distance is calculated by solving a 4th order polynomial. This is time consuming and does not allow analytical derivation [Sturm and Gargallo, 2007]. We propose an approximation to this distance replacing the point-to-conic distance by a point-to-point distance.

Given a conic point $\mathbf{x}_c$ lying on a conic described by the matrix Ω the tangent line to the conic passing through $\mathbf{x}_c$ is the polar line of the point $\mathbf{x}_c$ with respect to the conic Ω.

$$s = \Omega\mathbf{x}_c = \begin{pmatrix} c_1 & \frac{c_2}{2} & \frac{c_4}{2} \\ \frac{c_2}{2} & c_3 & \frac{c_5}{2} \\ \frac{c_4}{2} & \frac{c_5}{2} & c_6 \end{pmatrix} \begin{pmatrix} x_c \\ y_c \\ 1 \end{pmatrix} = \begin{pmatrix} c_1x_c + \frac{c_2}{2}y_c + \frac{c_4}{2} \\ \frac{c_2}{2}x_c + c_3y_c + \frac{c_5}{2} \\ \frac{c_4}{2}x_c + \frac{c_5}{2}y_c + c_6 \end{pmatrix} \tag{2.13}$$

The perpendicular line to the conic Ω passing through $\mathbf{x}_c$ is defined as

[1] We have observed that the positive solution is the correct one.



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

Figure 2.3: Region defined by a metric threshold of 20 pixels surrounding a conic. (a) Ellipse case using metric distance. (b) Ellipse case using Sampson distance. (c) Hyperbola case using metric distance. (d) Hyperbola case using Sampson distance.

$$l_{\perp}(\mathbf{x}_c) = \begin{pmatrix} s_2 \\ -s_1 \\ -s_2 x_c + s_1 y_c \end{pmatrix} \tag{2.14}$$

When a point $\mathbf{x}_o$ does not lie on the conic we can compute an estimation to its corresponding perpendicular line using the property that $l_{\perp}(\mathbf{x}_c) = l_{\perp}(\mathbf{x}_0 + \Delta\mathbf{x}) \approx l_{\perp}(\mathbf{x}_0)$.

$$l_{\perp}(\mathbf{x}_0) = \begin{pmatrix} s_2(\mathbf{x}_0) \\ -s_1(\mathbf{x}_0) \\ -s_2(\mathbf{x}_0) x_0 + s_1(\mathbf{x}_0) y_0 \end{pmatrix} \tag{2.15}$$

With the approximated perpendicular line we compute the intersection between the line and the conic [Barreto, 2003] using the expression,

$$p^{\pm} = \left(\pm \sqrt{-l'_{\perp} \mathsf{C}^* l_{\perp}} \mathsf{I}_{3x3} + l_{\times} \mathsf{C} \right) \mathsf{I}_s l_{\perp} \tag{2.16}$$

where

$$\mathsf{I}_s = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 0 & -1 \\ 1 & 1 & 0 \end{pmatrix}, \; \mathsf{C}^* = \det(\mathsf{C})\, \mathsf{C}^{-1}, \; l_{\times} = \begin{pmatrix} 0 & s_2 x_0 - s_1 y_0 & -s_1 \\ s_1 y_0 - s_2 x_0 & 0 & -s_2 \\ s_1 & s_2 & 0 \end{pmatrix}.$$

The approximated distance is the minor distance from both intersections.

Given a conic, a metric threshold with the proposed distance defines a uniform area surrounding the conic (Fig. 2.3 (a) and Fig. 2.3 (c)). When using other approaches like Sampson's distance this property is not guaranteed. In Fig. 2.3 (b,d) we show the area defined by a threshold based on Sampson's distance. We observe that the area is uniform when the conic is an ellipse; however when the conic is a hyperbola the thickness of the area using Sampson's distance is not uniform.

(a) (b) (c)

Figure 2.4: Extraction of image lines (HLIs). (a) Canny edge detector result, (b) connected components and (c) Extracted HLIs.

2.3.3 Hypercatadioptric line-images extraction

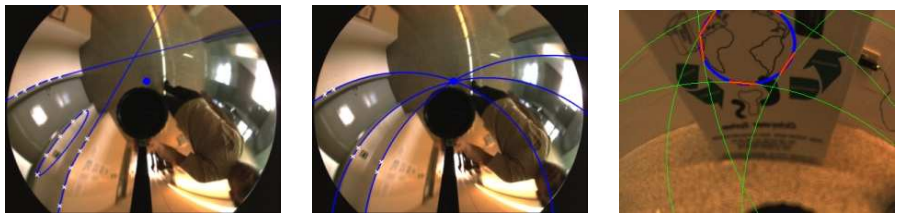
Our line extraction proposal can be explained as follows. First we detect the edges using the Canny algorithm. Then the connected pixels are stored in components. For each component we perform a RANSAC approach to detect the conics present into this component. Two points from the connected component are selected randomly and the corresponding HLI is computed. The distances from the rest of the points to this conic are computed. The points with a distance smaller than a threshold vote for this HLI. The process stops when the number of points that has not voted for any conic and the number of points in the component are smaller than a threshold. In Fig. 2.4 we can observe the three main steps to extract HLIs. A pseudo-code version is presented in Algorithm 1.

Five points vs. two points

For illustration purposes we show in Fig. 2.5 (a-b) the extraction of three conics corresponding to vertical lines. We use conic fitting with five points and the extraction of the same HLIs using our 2-point approach. We also show the VVP which is used as a measure of quality. All conics corresponding to vertical lines must intersect this point. When the generic five point approach is used, the shape of the HLIs changes depending on the points used to compute it. It can be either a hyperbola or an ellipse and it can easily change from one to the other. The result is a conic which fits the five selected points but it does not describe the projection of the 3D line. For example, using conics showed in Fig. 2.5(a) we can not compute the vertical vanishing point by intersection. In the case of the proposed 2-point algorithm the HLI cover with accuracy the points used to compute it, and describe the edge from which the points were extracted. Therefore the three conics intersect in the vanishing point. Another advantage of the 2-point approach is the number of iterations performed inside voting approach. For instance, using a probability $p = 99\%$ of not failing in the random search and 50% of outliers (ε) just 17 iterations are needed to get a result using the proposed approach (2-point)

Algorithm 1 Two-point HLI extraction algorithm

Require: Image
Ensure: Ω_{array}
  $edges = Canny(image)$
  $boundaries = extractboundaries(edges)$
  for $k = 1$ to $nBoundaries$ **do**
    $\mathbf{x} = boundaries(k)$
    $j = 0$
    while $remaining\_points/total\_points > T$ **do**
      $\mathbf{x}_{nplane} = \mathsf{H}_c^{-1}\mathbf{x}$
      for $i = 1$ to $nAttemps$ **do**
        $\mathbf{x}_{random} = rand(\mathbf{x}_{nPlane}, 2)$
        $\Omega_{img}(i) = twoPoints2Conic(\mathbf{x}_{random})$
        $d = dist2conic(\Omega_{img}(i), \mathbf{x})$
        $\mathbf{x}_{vote}(i) = inliers(\mathbf{x}, d, Threshold)$
        $votes(i) = size(\mathbf{x}_{inliers}(i))$
      end for
      $ind_{max} = MaximumVoted(\Omega_{img}, votes)$
      $\Omega = \Omega_{img}(ind_{max})$
      $\mathbf{x}_{inliers} = \mathbf{x}_{vote}(ind_{max})$
      $j = j + 1$
      $\Omega_{array}(j) = \Omega$
      $\mathbf{x} = UpdateRemainingBoundary(\mathbf{x}, \mathbf{x}_{inliers})$
    end while
  end for



(a) (b) (c)

Figure 2.5: Computing a HLI with (a) using the five point approach. (b) using our approach with only two close points. The central blue point corresponds to the vertical vanishing point. (c) Extraction of HLIs over a circular contour. Blue points give the connected component. Red points correspond to the HLIs detected by RANSAC. The corresponding conics to each segment HLI are shown in green.

and 146 using the general five point approach. The number of iterations n_r is given by $n_r = \frac{\log(1-p)}{\log(1-(1-\varepsilon)^k)}$.

Discrimination of HLIs from other conics

Since lines present in the scene become conics in hypercatadioptric images, one may think that these conics may represent projections of circles or other conic shapes. Our approach is able to distinguish between conics which are projections of 3D straight lines and the rest of the image conics. To illustrate this, we show in Fig. 2.5 (c) the behavior of our approach extracting HLIs in hypercatadioptric images. We observe a circular contour whose boundary points are given manually. We apply our extraction algorithm on this connected component assuming the contour is composed of HLIs. We can observe how the algorithm is not able to fit the whole circle with a single conic. With our approach we can directly test if a set of image points corresponds to a HLI. The definition of a HLI presented in section 2.3.1 is not limited to two points. From (2.10) we observe that each point gives a row of a linear homogeneous system. For n points we have

$$\mathsf{M} = \begin{pmatrix} \bar{x}_1 & \bar{y}_1 & -\alpha_1 \\ \bar{x}_2 & \bar{y}_2 & -\alpha_2 \\ \vdots & \vdots & \vdots \\ \bar{x}_n & \bar{y}_n & -\alpha_n \end{pmatrix} \tag{2.17}$$

This situation allows fitting a HLI using more than two points. This is very useful in practice when we have more than two pixels defining a conic. At the same time it permits to distinguish HLIs from other conics. The rank of the homogeneous matrix M is two when it represents a HLI.

2.4 Influence of calibration and error propagation in the extraction of HLIs

In this section we present an exhaustive analysis of the influence of the calibration parameters of a hypercatadioptric system on the extraction of HLIs. First, we present an analysis of the error propagation from the calibration parameters and the two defining points of a HLI to the normal **n** describing the projection plane of the 3D line. Then we show an empirical study of the influence of calibration parameters on the extraction of HLIs. Finally we analyze the influence of the length of the observed conic segment.

2.4.1 Analytical study of errors propagation

We have performed an analytical study of the propagation errors in the calibration parameters to the normal defining the projection of the line. We also propagate the error in pixels from a pair of defining points of a HLI on the image to its normal in the space. To deal with this uncertainty we use a first order approximation of the formulation.

We first analyze the uncertainty from the calibration parameters to the normal **n**. For this formulation the calibration parameters of the system are represented in the vector $\kappa = (u_0, v_0, f_x, f_y, s_{skew}, \xi, \psi)^\mathsf{T}$. The corresponding Jacobian is

$$\frac{d\mathbf{n}}{d\kappa} = \Psi(\Lambda\Upsilon + \Gamma) \tag{2.18}$$

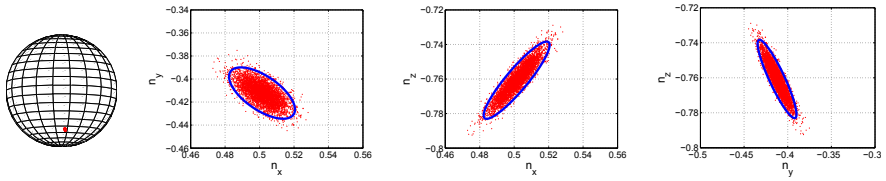
with

$$\Psi = \frac{1}{\left(n_x^2 + n_y^2 + n_z^2\right)^{\frac{3}{2}}} \begin{pmatrix} n_y^2 + n_z^2 & -n_x n_y & -n_x n_z \\ -n_x n_y & n_x^2 + n_z^2 & -n_y n_z \\ -n_x n_z & -n_y n_z & n_x^2 + n_y^2 \end{pmatrix}, \tag{2.19}$$

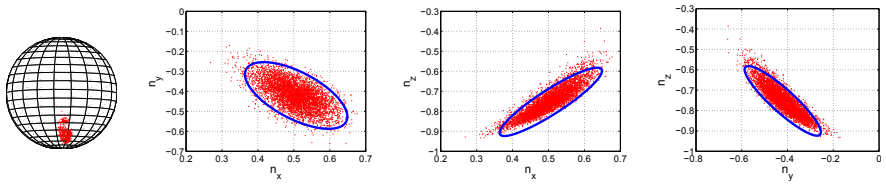
$$\Gamma = \frac{1}{(1-\xi^2)^2} \begin{pmatrix} 0_{3\times5} & \begin{matrix} -\frac{y_2(\lambda_1 - \xi z_1)}{\lambda_1} + \frac{y_1(\lambda_2 - \xi z_2)}{\lambda_2} \\ \frac{x_2(\lambda_1 - \xi z_1)}{\lambda_1} - \frac{x_1(\lambda_2 - \xi z_2)}{\lambda_2} \\ 0 \end{matrix} & 0_{3\times1} \end{pmatrix} \tag{2.20}$$

and

$$\Upsilon = \frac{1}{(\xi - \psi)} \begin{pmatrix} \frac{\hat{z}_1}{f_x} & -\frac{s\hat{z}_1}{f_x f_y} & \frac{f_y \Delta x_1 - s\Delta y_1}{f_y f_x^2} & -\frac{s\Delta y_1}{f_y^2 f_x} & \frac{\Delta y_1}{f_y f_x} & \frac{f_y \Delta x_1 - s\Delta y_1}{f_y f_x (\xi - \psi)} & \frac{-f_y \Delta x_1 + s\Delta y_1}{f_y f_x (\xi - \psi)} \\ 0 & -\frac{\hat{z}_1}{f_y} & 0 & \frac{\Delta y_1}{f_y^2} & 0 & -\frac{\Delta y_1}{f_y (\xi - \psi)} & \frac{\Delta y_1}{f_y (\xi - \psi)} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\hat{z}_2}{f_x} & -\frac{s\hat{z}_2}{f_x f_y} & \frac{f_y \Delta x_2 - s\Delta y_2}{f_y f_x^2} & -\frac{s\Delta y_2}{f_y^2 f_x} & \frac{\Delta y_2}{f_y f_x} & \frac{f_y \Delta x_2 - s\Delta y_2}{f_y f_x (\xi - \psi)} & \frac{-f_y \Delta x_2 + s\Delta y_2}{f_y f_x (\xi - \psi)} \\ 0 & -\frac{\hat{z}_2}{f_y} & 0 & -\frac{\Delta y_2}{f_y^2} & 0 & -\frac{\Delta y_2}{f_y (\xi - \psi)} & \frac{\Delta y_2}{f_y (\xi - \psi)} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{2.21}$$



(a)



(b)

Figure 2.6: Uncertainty from defining points to normal vector **n**. Gaussian randomized points around the defining points are used to compute a cloud of normals (red points). The 95% confidence region of the normal is represented from this propagation with an ellipse (in blue). (a) Standard Deviation of 2 pixels. (b) Standard Deviation of 15 pixels.

where $\Delta x_1 = \hat{x}_1 - \hat{z}_1 u_0$, $\Delta x_2 = \hat{x}_2 - \hat{z}_2 u_0$, $\Delta y_1 = \hat{y}_1 - \hat{z}_1 u_0$ and $\Delta y_2 = \hat{y}_2 - \hat{z}_2 u_0$.

We have also obtained the error propagation formulas from the two defining points on the image to the normal **n**. Let be $\hat{\chi} = (\hat{x}_1, \hat{y}_1, \hat{z}_1, \hat{x}_2, \hat{y}_2, \hat{z}_2)^\mathsf{T}$ the corresponding Jacobian is described by the following matrix product

$$\frac{d\mathbf{n}}{d\hat{\chi}} = \Psi\Lambda \begin{pmatrix} \mathsf{H}_c^{-1} & 0_{3\times3} \\ 0_{3\times3} & \mathsf{H}_c^{-1} \end{pmatrix} \tag{2.22}$$

in which

$$\Lambda = \begin{pmatrix} \frac{-\bar{y}_2\bar{x}_1\xi}{\lambda_1} & \frac{-\bar{y}_2\bar{y}_1\xi}{\lambda_1} + \alpha_2 & \frac{-\bar{y}_2(-1+\xi\bar{z}_1)}{\lambda_1(1-\xi^2)} & \frac{\bar{y}_1\bar{x}_2\xi}{\lambda_2} & \frac{\bar{y}_2\bar{y}_1\xi}{\lambda_2} - \alpha_1 & \frac{\bar{y}_1(-1+\xi\bar{z}_2)}{\lambda_2(1-\xi^2)} \\ \frac{\bar{x}_2\bar{x}_1\xi}{\lambda_1} - \alpha_2 & \frac{\bar{y}_1\bar{x}_2\xi}{\lambda_1} & \frac{\bar{x}_2(-1+\xi\bar{z}_1)}{\lambda_1(1-\xi^2)} & \frac{-\bar{x}_2\bar{x}_1\xi}{\lambda_2} + \alpha_1 & \frac{-\bar{y}_2\bar{x}_1\xi}{\lambda_2} & \frac{-\bar{x}_2(-1+\xi\bar{z}_2)}{\lambda_2(1-\xi^2)} \\ -\bar{y}_2 & \bar{x}_2 & 0 & \bar{y}_1 & -\bar{x}_1 & 0 \end{pmatrix}, \tag{2.23}$$

where $\lambda_i = \sqrt{\bar{z}_i^2 + (\bar{x}_i^2 + \bar{y}_i^2)(1-\xi^2)}$.

This approximation allows propagating the error from the two defining points in the image plane to the normal **n**. To evaluate the expressions, in fig. 2.6 we show the uncertainties of two points defining the HLI. We generated 5000 random values of $\hat{\chi}$ on the image adding a Gaussian variance of two pixels and fifteen pixels (Fig. 2.6(a) and 2.6(b), respectively). These points have been mapped to normal vectors using (2.11).

We compare these points with the 95% confidence region we obtain propagating the covariance matrix of $\hat{\chi}$.

2.4.2 Empirical study of the influence of calibration

In this experiment we consider the following calibration parameters, focal lengths (f_x, f_y), principal point (u_0, v_0) and mirror parameter ξ. We modify each parameter independently inside a certain range from which we select 1000 samples. Then, we compute a specific HLI for each value using its two extreme points. We perform this experiment for two different types of HLIs, one of which is nearly perpendicular to the optical axis and the other which is nearly parallel. To illustrate the results two types of graphics are showed. The first one represents the mean error in pixels of the conic fitting as a function of the calibration parameter. In the second type the horizontal axis represents the variation of the calibration parameter and the vertical axis the angle between the reference normal vector $\mathbf{n}_{ref}$ and the normal vector $\mathbf{n}$ computed with the modified calibration.

- *Focal length:* In Fig. 2.7(a),(b) we show the corresponding conics when the focal lengths (f_x, f_y) are modified. We observe that these parameters affect their corresponding coordinate of influence x and y. The plot shows that the effect is the inverse for both axis. However, the magnitude is bigger in the case of f_y. Lines with a more horizontal component are more affected by the change on the focal length.

- *Principal point:* In Fig. 2.7(c),(d) we observe the influence of the principal point (u_0, v_0) on the computation of the HLIs. When these two parameters are modified we observe a displacement proportional to the distortion added to the corresponding parameter on the corresponding axis.

- *Mirror parameter:* In Fig. 2.7(e) we observe the effect of the mirror parameter ξ on the computation of HLIs. We observe that both lines approximate a straight line when the value of ξ approximates to zero. This is explained because $\xi = 0$ represents the pin-hole model which projects lines to lines. Therefore, the closer a HLI is to a straight line, the less the mirror parameter ξ has an influence.

We observe that the fitting error is bigger for the perpendicular line than for the parallel line, except for errors in v_0. This is explained since the perpendicular lines are transformed in conics with a higher curvature than the parallel ones, which are mapped to straighter lines.

2.4.3 Influence of observed length of the HLI

In the extraction process of the HLIs using RANSAC, two points in a connected component are randomly selected and a HLI is computed. Then it is used to obtain all the

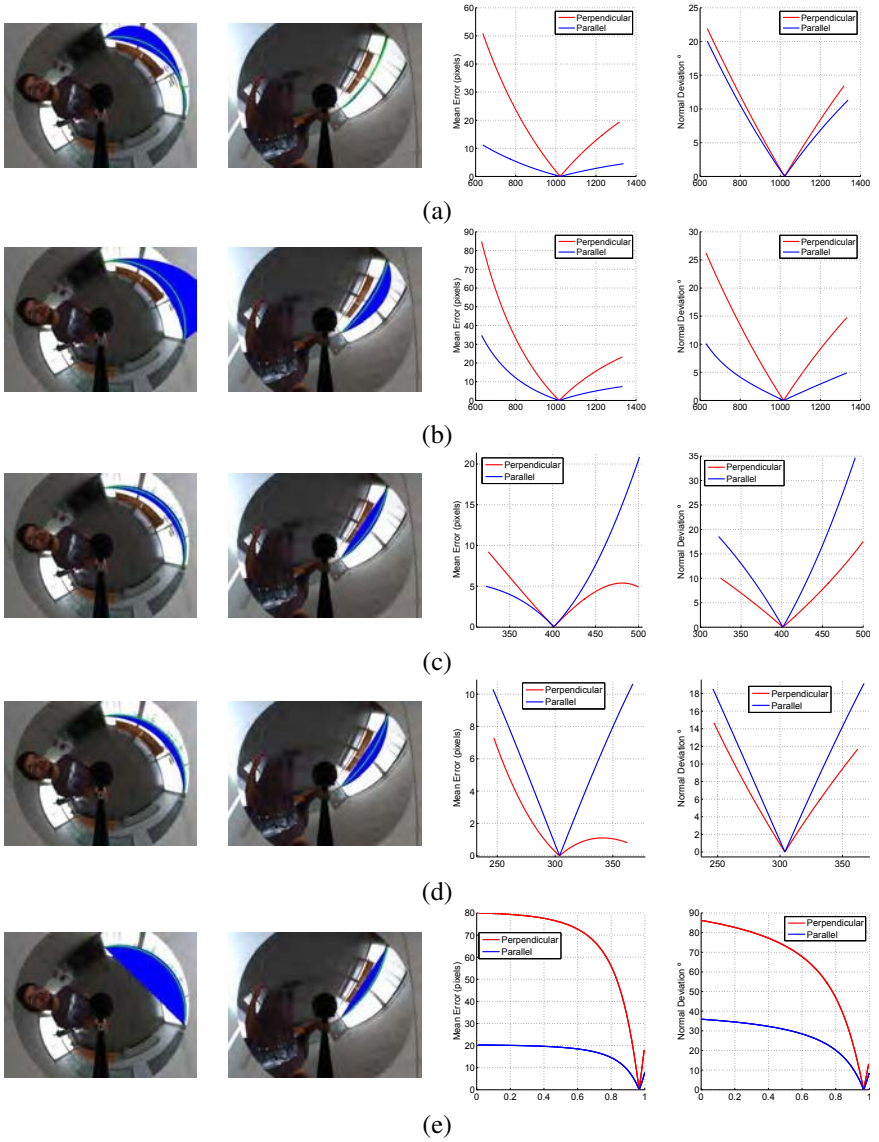
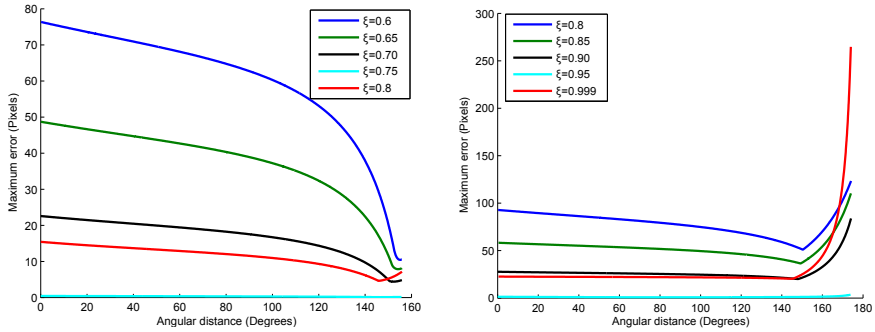


Figure 2.7: Mean error in pixels of the conic fitting varying the calibration parameters. Angle between reference normal and normal when varying the calibration parameters. (a,b) Focal length, f_x, f_y. (c,d) Principal point, u_0, v_0. (e) Mirror parameter ξ.



(a) (b)

Figure 2.8: Maximum error in pixels as a function of the extension (in degrees) of the observed HLI for mirror parameters (a) $\xi = 0.75$ and (b) $\xi = 0.95$.

points that belong to such HLI. In this experiment we want to observe the influence of the distance between these two defining points on the accuracy of the extracted HLI. As in previous experiment we observe that the influence of calibration errors is stronger in perpendicular lines to the optical axis. In this experiment we perform simulations using this type of line. We try different line lengths, from a few degrees to the longest theoretical line which has an observed extension close to 180°. We also modify the mirror parameter to observe its behavior. As we expected, we observe that the longer the line is, the better the extraction of the HLI is. In Fig. 2.8(a) we observe such behavior with a mirror parameter of $\xi = 0.75$. We also observe that when we approach the maximum observed length of the line the error starts increasing, showing the limit of the longest line that should be used to extract the corresponding HLI. In Fig. 2.8(b) we repeat the experiment, this time using a mirror parameter of $\xi = 0.95$ close to a parabolic mirror ($\xi = 1$). We observe that the error increases a lot when we are closer to both the longest line and the parabolic mirror. The para-catadioptric system case has been studied in [Barreto and Araujo, 2005] and it is simpler than the hypercatadioptric analyzed here.

With the information obtained from this experiment we can infer that we should select the two more distant points to extract their corresponding HLI. However, there exist an upper limit that depends on the length of the HLI and the mirror parameter. In practice it is very unlikely to observe a line of such length. Since we use a RANSAC approach, whose principle is to search in the space of solutions minimizing the error, this problematic situation is automatically avoided in practice.

2.5 Vanishing points and image rectification

The vanishing points indicate the intersection of image lines corresponding to parallel lines in the scene. In vertical aligned catadioptric systems, vertical lines are radial lines in the catadioptric image. Their intersection point, the vertical vanishing point (VVP), is located close to the image center. When the camera is not vertically aligned, the vertical lines become conic curves as we explained before. One consequence is that the VVP displaces from the image center. Its new location contains important information about the orientation of the camera with respect to the scene.

2.5.1 Intersection of two HLIs using the common self-polar triangle

In a general configuration, two conics intersect in four points (Fig. 2.9 (a)). The union of couples of these points define three distinct pair of lines. The intersection of these lines represent the vertices of the self-polar triangle common to a pair of conics [Barreto, 2003]. We have studied the particular case where two HLIs intersect, which is a degenerate configuration, since they intersect in two points. As we observe in Fig. 2.9 (b), there exist a line l that intersects these two points and the origin of the normalized plane. Our goal is to compute this line and from it to extract the two intersections of the conics that correspond to the two points $\mathbf{P}^+$ and $\mathbf{P}^-$.

Let $\mathbf{n}_1 = (n_{x_1}, n_{y_1}, n_{z_1})^\mathsf{T}$ and $\mathbf{n}_2 = (n_{x_2}, n_{y_2}, n_{z_2})^\mathsf{T}$ be two normal vectors representing the projection of two lines in the scene and $\bar{\Omega}_1$ and $\bar{\Omega}_2$ two conics representing the image lines in the normalized plane. The vertices of the self-polar triangle associated to the pencil $\bar{\Omega}(\lambda) = \bar{\Omega}_1 + \lambda\bar{\Omega}_2$ satisfy the constraint

$$\det(\bar{\Omega}_1 + \lambda\bar{\Omega}_2) = 0. \tag{2.24}$$

If we develop this constraint we obtain a third order polynomial where just one of the solutions is real and it corresponds to $\lambda_1 = -n_{z1}^2/n_{z2}^2$. So, the null-space of $\bar{\Omega}(\lambda_1) = \bar{\Omega}_1 + \lambda_1\bar{\Omega}_2$ is the line l, expressed in a parametric way as

$$l = \mu \cdot \mathbf{v} = \mu \begin{pmatrix} v_x \\ v_y \end{pmatrix} = \mu \begin{pmatrix} n_{z_2}^2 n_{y_1} n_{z_1} - n_{z_1}^2 n_{y_2} n_{z_2} \\ n_{z_1}^2 n_{x_2} n_{z_2} - n_{z_2}^2 n_{x_1} n_{z_1} \end{pmatrix}. \tag{2.25}$$

The intersection of this line to both $\bar{\Omega}_1$ and $\bar{\Omega}_2$ gives the two points $\mathbf{P}^+$ and $\mathbf{P}^-$. To obtain them we solve for μ in the following equation

$$\mu^2(c_1 v_x^2 + c_2 v_x v_y + c_3 v_y^2) + \mu(c_4 v_x + c_5 v_y) + c_6 = 0 \tag{2.26}$$

and substitute in (2.25).

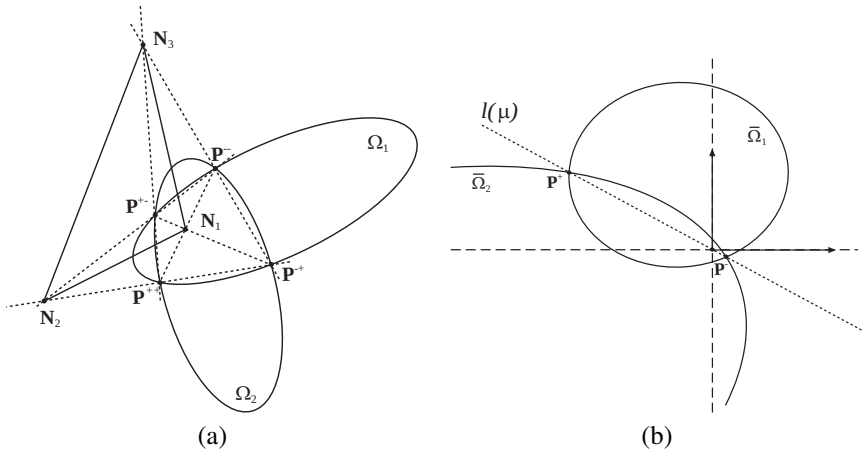


Figure 2.9: (a) Intersection of two generic conics. (b) Intersection of two conics in the normalized plane.

2.5.2 Vertical vanishing point (VVP)

We use a classic algorithm to detect the VVP. Let m be the number of putative vertical HLIs detected in the omnidirectional image and let $\mathbf{n}_i$ their corresponding representation in the normalized plane. For every pair of HLIs (there is a total of $m(m-1)/2$ pairs), we compute their intersection as explained above. Then for each line $\mathbf{n}_i$ we compute the distance to these points. If the line is parallel to that pair of HLIs the distance is smaller than a threshold and then that line votes that possible VVP. The most voted point is considered the VVP. A refinement of the estimation can be performed using the p lines that voted for the VVP. This refinement can be performed using singular value decomposition to solve a linear system, followed by an optimization process to improve the accuracy. As these steps also increase the computational cost, we decide to avoid them in our final implementation.

2.5.3 Horizontal vanishing point (HVP)

Once the VVP is extracted we can exploit several properties to compute the horizontal vanishing point (HVP). The VVP $\bar{\mathbf{x}}_{VVP} = \left(\bar{x}_{v_x}, \bar{x}_{v_y}\right)$ defines a separation between vertical lines and potential horizontal lines. The VVP defines the plane of the horizon,which is projected into the horizon conic, defined as:

$$\Omega_v = \begin{pmatrix} \zeta^2\bar{x}_{v_x}^2\left(1-\xi^2\right)-(\zeta-\xi)^2\xi^2 & \zeta^2\bar{x}_{v_x}\bar{x}_{v_y}\left(1-\xi^2\right) & \zeta\bar{x}_{v_x}(\zeta-\xi) \\ \zeta^2\bar{x}_{v_x}\bar{x}_{v_y}\left(1-\xi^2\right) & \zeta^2\bar{x}_{v_y}^2\left(1-\xi^2\right)-(\zeta-\xi)^2\xi^2 & \zeta\bar{x}_{v_y}(\zeta-\xi) \\ \zeta\bar{x}_{v_x}(\zeta-\xi) & \zeta\bar{x}_{v_y}(\zeta-\xi) & (\zeta-\xi)^2 \end{pmatrix}, \quad (2.27)$$

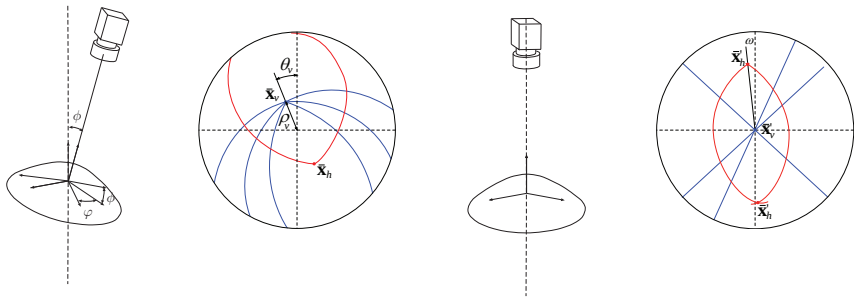


Figure 2.10: (a) Camera Reference: The VVP location is related with orientation angles ϕ and φ. (b) Vertical Reference: After rotating the camera around the x-axis an angle $-\phi$ the VVP is in the principal point. In this reference the HVP defines the orientation angle ω. Each vertical 3D line is projected on a radial straight line.

where $\zeta = \frac{\xi+\sqrt{1+\left(1-\xi^2\right)\left(\bar{x}_{vx}^2+\bar{x}_{vy}^2\right)}}{\bar{x}_{vx}^2+\bar{x}_{vy}^2+1}$. Therefore the HVP must lie on this conic.

By introducing these constraints we reduce the search space and we impose the main directions to be perpendicular. Notice that using a single image without any assumptions it is impossible to distinguish between the VVP and the HVP. However, some prior knowledge is easy to have in practice.

2.5.4 Image rectification

Here we explain the relation between the VVP computed in the normalized plane and the whole orientation of the catadioptric system. Having an absolute reference system defined by the main directions, the orientation of the camera with respect to this system is described by three angles (ϕ,φ,ω). Two of them are obtained from the VVP. Writing the VVP in polar coordinates $\bar{\mathbf{x}}_{vp} = (\rho_v, \theta_v)^\mathsf{T}$ (Fig. 2.10(a)) we observe that there exist a relation between the angle θ_v and the angle φ representing the rotation of the catadioptric system around the z-axis (2.28). The negative angle is produced by the mirror effect which inverts the catadioptric image.

$$\varphi = -\theta_v \tag{2.28}$$

We observe that the component ρ_v is intrinsically related to the angle ϕ and the mirror parameter ξ of the catadioptric system. Since angles ϕ and φ are independent, we consider the case where $\varphi = 0$ (Fig. 2.10(b)). Using (2.25) and (2.26) with a pair

of parallel HLIs in polar coordinates we compute the following relation

$$\rho_v = -\frac{\sin\phi}{\cos\phi \pm \xi}, \tag{2.29}$$

selecting geometrically compatible solutions, ϕ can be isolated resulting in:

$$\phi = \text{atan2}\,(1,\rho_v) + \arccos\left(\frac{-\rho_v\xi}{\sqrt{\rho_v^2+1}}\right). \tag{2.30}$$

With ϕ and φ angles any image can be transformed to a vertical reference in which VVP lies on the image center. We perform the vertical rectification in two steps. The first step undoes a horizontal rotation according to angle φ ($-\theta_v$ in the image) to an arbitrary central reference (see Fig. 2.10(b)). The second step consists of translating the VVP to the image center through a rotation around the vertical axis by the angle ϕ, which is computed (2.30) from ρ_v (in the image) and the mirror parameter ξ. This procedure is performed by the following equation

$$\bar{\mathbf{x}}'_{\mathbf{h}} = \hbar\left(rot_z(\varphi)\, rot_x(-\phi)\, rot_z(-\varphi)\, \hbar^{-1}(\bar{\mathbf{x}}_{\mathbf{h}})\right) \tag{2.31}$$

Once we have rectified the HVP to this reference we can compute the full orientation and perform the full rectification of the catadioptric image. We compute its polar coordinates from which we only require the angle component ω. With this angle we translate it to a central position. Since a horizontal rotation was performed in the previous step, the position of the HVP has been modified and we have to undo such transformation. From these angles we construct a rotation matrix R which relates the absolute reference system with the camera reference system

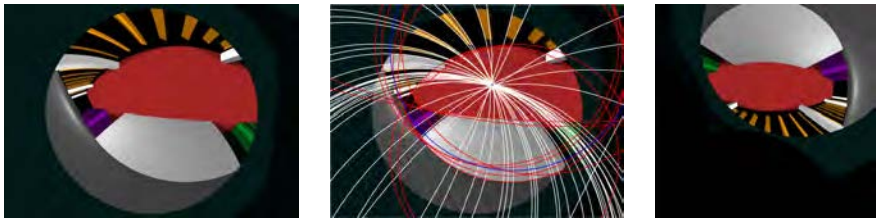
$$\mathbf{X}_{cam} = \mathsf{R}\mathbf{X}_{abs}, \quad \text{where} \qquad \mathsf{R} = rot_z(\varphi) rot_x(\phi) rot_z(\omega - \varphi) \tag{2.32}$$

2.6 Experiments

In this section we present experiments focused on testing the accuracy of the proposed algorithm. We use synthetic and real images as well as image sequences acquired with a hand-held hypercatadioptric system and a hypercatadioptric system mounted on a helmet.

2.6.1 Orientation accuracy

This group of experiments is designed to evaluate the accuracy of the orientation of the camera computed from vanishing points. Initially, we use synthetic images to avoid setting and calibration errors. We also use real images which are acquired with a hypercatadioptric system. We accurately move this system with a precision head composed of a goniometer, from which we obtain the ground truth orientation of the system.

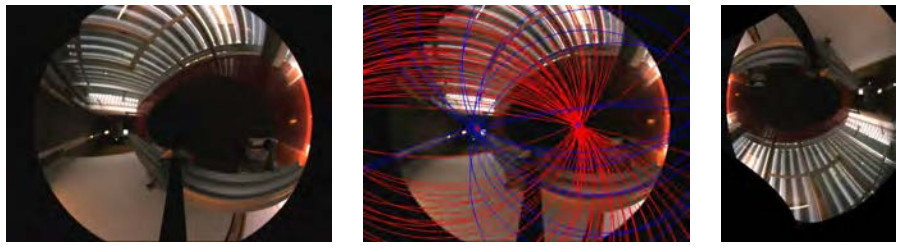


(a) (b) (c)

Figure 2.11: Orientation accuracy test with synthetic images: (a) Original Image, (b) VP extraction, (c) Rectified image using the orientation.

Reference	Our Method	Bazin's Method
0°	0.45°	0.12°
5°	5.27°	5.10°
10°	10.11°	9.80°
15°	15.07°	14.56°
20°	19.90°	19.77°
25°	24.58°	24.76°
30°	30.09°	29.79°
35°	35.54°	34.73°
40°	40.41°	39.77°
45°	45.35°	45.02°
50°	50.06°	49.71°
55°	55.01°	54.74°
60°	59.97°	59.75°

Table 2.1: ϕ angle (in degrees) extraction accuracy test on simulated scenario.



(a) (b) (c)

Figure 2.12: Orientation accuracy test with real images in scenario A: (a) Original Image, (b) VP extraction, (c) Rectified image.

Synthetic images

The goal of this experiment is to estimate the orientation errors with the proposed method avoiding the calibration influence. We use a synthetic environment based on a real scenario composed of elements like doors and columns which define main directions. The catadioptric system is modelled by a perspective camera and a hyperbolic mirror. To test the proposed method thirteen images have been generated from 0^o to 60^o using a ray-tracing software[2] introducing a consecutive variation of five degrees in ϕ angle. In Table 2.1 we show the results obtained with the proposed method and the method presented by Bazin et al. in [Bazin et al., 2010]. The obtained errors are in average smaller than 0.25 degrees. There is no significant difference between our results and Bazin's method. Fig. 2.11 shows an example of the vanishing points extraction and full image rectification from a synthetic image generated from an angle $\phi = 40^o$.

Real images

In this experiment we test the accuracy of the algorithm in a real environment. The camera is attached to a precision head[3] which provides the ground truth. Images have been acquired with a hypercatadioptric system[4]. To evaluate the influence of the lines density and quality, the experiment has been repeated in two different scenarios. Scenario A is a corridor where main directions are well defined, the number of vertical lines is high, floor and walls have different colors and without reflections. Scenario B is a small hall in which the number of vertical lines is low, the color of the constructive elements is uniform and there exist reflections on the floor.

[2] http://www.povray.org

[3] UTR 80, Manufactured by Newport with goniometer of 1/60 ^{o} resolution

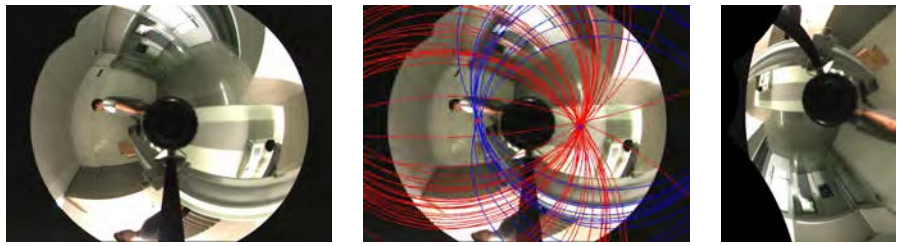
[4] http://neovision.cz

Reference	Scenario A mean	Scenario A std. dev.	Scenario B mean	Scenario B std. dev.
0°	0.63°	0.153°	−0.32°	0.226°
5°	4.76°	0.227°	5.22°	0.162°
10°	10.87°	0.417°	11.28°	0.167°
15°	15.30°	0.200°	15.83°	0.086°
20°	20.15°	0.038°	21.57°	0.300°
25°	25.21°	0.021°	25.45°	0.038°
30°	30.06°	0.045°	29.75°	0.382°
35°	34.88°	0.043°	35.20°	0.468°
40°	39.97°	0.132°	39.30°	0.185°
45°	44.93°	0.083°	44.38°	0.386°
50°	49.93°	0.031°	49.28°	0.345°
55°	54.42°	0.036°	53.62°	0.249°
60°	59.22°	0.008°	59.13°	1.115°

Table 2.2: ϕ angle (in degrees) extraction accuracy test on real stages.

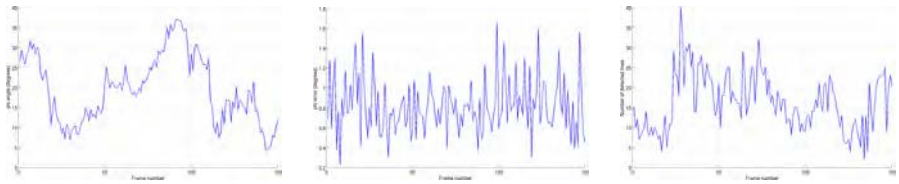
We also use thirteen images from 0 to 60 deg which have been acquired introducing an increment of five degrees in ϕ angle and the algorithm has been executed five times for each image. In Table 2.2 we show the result from the different two scenarios. The first column is the reference angle given by the goniometer. Next columns are the mean and standard deviation of the orientation calculation for each scenario. Since we have not a perfectly aligned system with respect to the vertical we assume an offset setup error for the complete experiment.

Mean errors are 0.31 degrees for well-conditioned scenario A and 0.80 degrees for scenario B. Maximum errors are 0.87 degrees and 1.57 degrees respectively. These results are better than maximum error of 2 degrees presented by Magnier et al. in [Magnier et al., 2010]. Results obtained in scenario A, which have more and longer lines, are considerably more accurate than results from scenario B. Therefore as expected, the number of HLIs and their quality have influence in angle accuracy. Spurious HLIs extracted from non-line objects, like people, also introduce uncertainty into the extraction process. We observe that the inclusion of one parameter radial distortion in the projection model improve the accuracy. We also observe the high influence of radial distortion and principal point in angle ϕ accuracy. The relation between angle ϕ and the distance from principal point to the VVP, which can be seen in (2.29), explains this tendency. In Fig. 2.12 and 2.13 we show an example of the vanishing points extraction from two real images taken in the two considered scenarios. The system has an inclination of 40 deg. Full orientation extracted from vertical and horizontal vanishing point is used to rectify the original image.



(a) (b) (c)

Figure 2.13: Orientation accuracy test with real images scenario B: (a) Original Image, (b) VP extraction, (c) Rectified image.



(a) (b) (c)

Figure 2.14: (a) Elevation angle ϕ of the catadioptric system. Frames 1-150. (b) Angular deviation of the vertical lines after rectification, in degrees. (c) Number of vertical lines present in each frame of sequence 1.

2.6.2 Rectification of image sequences

In order to evaluate the system in real environments the proposed method has been tested with image sequences. This application consists of a 3D orientation computation in which the extracted orientation is used to rectify the image sequence. The result is an oriented image sequence.

As commented in previous sections, it is necessary to introduce additional constraints to distinguish between VVP and HVP. When we deal with image sequences it is possible to track the vanishing points. The vanishing points detected in the first frame are defined as reference. The tracking is performed with previous VPs inside a confidence region. This region is described as a covariance matrix which depends on the hypothesized angular velocity. Three sequences have been acquired using two different catadioptric systems. Sequences 1 and 2 have been acquired with a hand-held hypercatadioptric system[5]. Sequence 3 has been acquired with a hypercatadioptric system mounted on a helmet (see Fig. 2.17(a)).

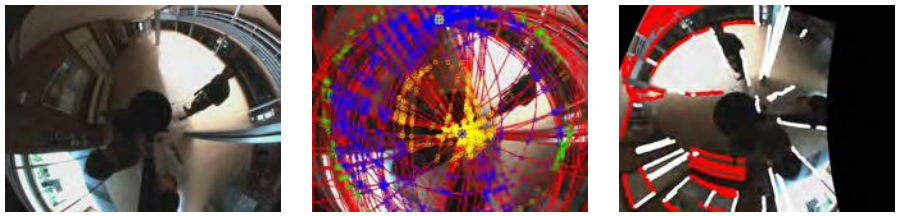
Image sequence 1

In this experiment we only perform the vertical rectification of a sequence. In Fig.2.14 (a) we show the ϕ angle computed for the first 150 frames of sequence 1. To measure the accuracy of our approach we compute the verticality of the lines present in every frame. This process consists of building the corresponding panoramic image from the omnidirectional image. Then all points belonging to a vertical line are used to compute the line equation. Finally we measure the angle deviation of this computed line with a true vertical line. The average error computed was 0.72° with a maximum error of 1.63° and a standard deviation of 0.70°. In Fig.2.14 (b) we show the average error of the angle deviation of all the vertical lines present in each frame of sequence 1 after rectification. We observe that this error is related to the number of vertical HLIs present in the catadioptric image (Fig. 2.14 (c)). The more the number of vertical HLIs present in the frame the better the estimation of the vanishing points, and consequently a better rectification of the image.

Image sequence 2

In this experiment we compute the full rectification of the second sequence. In Fig. 2.15 we can observe how the two vanishing points are computed. Once the rectification is computed we align the images to the reference system given by the vanishing points, i.e., the scene reference system. This allows to see that the motion performed in the sequence is pure translation (Fig. 2.16).

[5] http://neovision.cz

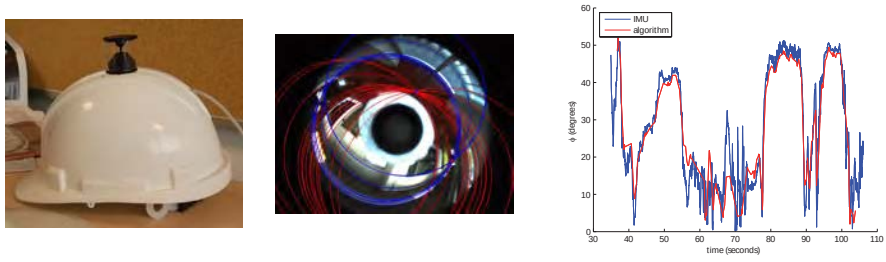


(a) (b) (c)

Figure 2.15: Example of full image rectification. (a) Frame 1 of the sequence 2. (b) Putative vertical and horizontal vanishing points. The yellow circles represent the putative vertical vanishing points. The blue ones the putative horizontal vanishing points and the green ones are the intersections points that cannot be consider either vertical or horizontal vanishing points. The white square is the estimated HVP and the black one is the VVP. (c) Full-rectified image. The vertical HLIs are shown in white and the horizontal ones in red.

Frame 107 Frame 242

Figure 2.16: Panoramic representation of several full-rectified frames. Vertical lines are shown in white and horizontal ones in red. The horizontal vanishing point is aligned to the image center.



(a) (b) (c)

Figure 2.17: Image sequence acquired with the omni-camera helmet: (a) Acquisition helmet, (b) Vanishing Point Extraction, (c) Comparison between angle ϕ given by the accelerometer and the vision based algorithm.

Helmet camera and IMU comparison

In this experiment we measure the orientation of a head during a human walk. An inertial measurement unit (IMU) has been coupled to the helmet to compare the computed angle with the angle given by the IMU. In Fig. 2.17(b) we show an image of the sequence and many conics lying on the vanishing points. Fig. 2.17(c) shows the comparison between the angle given by the IMU and the algorithm during the image sequence.

2.7 Summary

We have presented a new method for the extraction of conics which are projections of 3D lines in omnidirectional images generated by a calibrated hypercatadioptric system. We propose an approach that requires only two points and the calibration of the system to extract them and use it with a metric distance from point to conic in an extraction algorithm. We analyze the uncertainty propagation using the proposed approach showing the influence of each particular calibration parameter and the two defining points. We develop the common self-polar triangle approach to the particular case of HLIs intersection. With a voting approach we select the VPs from the intersections of the HLIs. Orientation of the hypercatadioptric system is estimated relating the location of the vanishing points on the image with the orientation angles in an man-made structured environment. To show the effectiveness of this method we perform experiments with synthetic and real images comparing our proposal to similar approaches. In particular we test the accuracy of the orientation estimation by using static images, obtaining mean orientation errors of less than 1 degree. We also show the behaviour of the method dealing with image sequences. We compare our algorithm with the orientation data extracted from an IMU. Using the orientation we perform the vertical rectification of the sequence in order to create images where applications that require the vertical constraint can be used.

Chapter 3

Line extraction in uncalibrated axial cameras

In this chapter we address a general representation for line-images in central systems assuming symmetry of revolution. This is a realistic assumption when modelling the majority of catadioptric and dioptric cameras. When extending line projection from central catadioptric to general central systems line projections are not conics. Instead, straight lines are projected on curves having more than two degrees of freedom whose shape strongly depends on the particular camera configuration. Therefore, the existing line-extraction methods for this kind of omnidirectional cameras require the camera calibration by contrast with the perspective case where the calibration is not involved in the shape of the projected line. However, this drawback can be considered an advantage since the shape of the line-images can be used for self-calibration. In this chapter we present a novel method to extract line-images in uncalibrated omnidirectional images, which is valid for radially symmetric central systems. In this method we propose using the plumb-line constraint to find closed form solutions for different types of camera systems, dioptric or catadioptric. The inputs of the proposed method are points belonging to the line-images and their intensity gradient. The gradient information allows to reduce the number of points needed in the minimal solution improving the result and the robustness of the estimation. The scheme is used in a line-image extraction algorithm to obtain lines from uncalibrated omnidirectional images without any assumption about the scene. The algorithm is evaluated with synthetic and real images showing good performance. The results of this work have been implemented in an open source Matlab toolbox for evaluation and research purposes.

3.1 Introduction

The projection of a straight line on an image is in general a curve called line-image. In conventional perspective cameras, a straight line is projected on the image via a projection plane and the resulting line-image is a 2D straight line which can be recovered from 2 image points. Notice that both the line-image and the projection plane have 2 degrees of freedom (DOF). As a consequence, part of the geometry of the 3D line, which has 4 DOFs, is lost in the projection. In non perspective central systems the projection surface of a straight 3D line is also a plane with 2 DOFs, however the line-image is a curve with more than 2 DOFs due to the non-linearity of the projection model. The additional DOFs of these curves are deeply related with the calibration of the system which is needed to define one of these curves from 2 points. Moreover, the projection plane provides a constraint which can be exploited to estimate the calibration whenever more than two independent constraints are available [Devernay and Faugeras, 2001, Alvarez et al., 2009, Brown, 1971]. Consequently, with the appropriate approach, line-images can be extracted without calibration.

Most approaches to extract lines from omnidirectional central images use camera calibration to back project the image points to normalized rays lying on a unitary sphere around the viewpoint [Ying and Hu, 2004c, Bazin et al., 2010]. In this space the representation of the line projection becomes linear and classical approaches can be used. In the particular case of central catadioptric systems the curve defining a line-image is a conic. Some approaches extract these conics directly on the images. In [Barreto and Araujo, 2005] conics are computed using classical conic fitting approaches, hence 5 DOFs are required in this fitting which is noise sensitive. In [Ying and Zha, 2005, Cucchiara et al., 2003] a Hough transform approach is used to simultaneously extract lines and calibration parameters from uncalibrated omnidirectional images.

In the case of non linear projection with unknown calibration, the location of additional edge points lying on the curve provides additional independent constraints. This kind of constraint is known as plumb-line constraint [Sturm et al., 2011] and it is one of the approaches used for self-calibration in central distorted images. Most of the plumb-line based approaches solve radial distortion models based on a transformation of the radius of an image point after a linear projection. Some of these models are the "even order polynomial model" used in [Swaminathan and Nayar, 2000, Devernay and Faugeras, 2001, Thormählen et al., 2003, Rosten and Loveland, 2011] and the " division model" proposed by Fitzgibbon in [Fitzgibbon, 2001] and extensively used in [Strand and Hayman, 2005, Wang et al., 2009, Bukhari and Dailey, 2013, Melo et al., 2013]. These models are usually used to describe radial distortion on conventional cameras and they can also be used to fit catadioptric and dioptric systems.

Other kind of projection models try to describe the physical phenomenon and the behaviour of the system, e.g. the sphere camera model [Baker and Nayar, 1999, Geyer and Daniilidis, 2000, Geyer and Daniilidis, 2001] for catadioptric systems (modelling the reflection of the light on the mirror) or the classical fisheye models: the

equiangular-fisheye model (also known as equidistant projection), the equisolid-fisheye model, the stereographic-fisheye model and the orthographic fisheye model [Kingslake, 1989, Stevenson and Fleck, 1996, Ray, 2002]. Actually, fisheye lenses are constructed in order to satisfy such models [Sturm et al., 2011]. A validation of these geometric models for real fisheyes is presented in [Schneider et al., 2009]. Assuming symmetry of revolution, these models are encoded using a function h that relates the elevation angle ϕ of the projection ray with the radius of an image point r. The function $r = h(\phi)$ can be an explicit expression [Kingslake, 1989, Ray, 2002] or can be defined as a series-based expression [Tardif et al., 2006, Kannala and Brandt, 2006, Kannala et al., 2008b].

Apart from the plumb-line approach there exist other methods for calibrating omnidirectional central cameras. A survey of these methods can be found in [Puig et al., 2012]. We want to remark that, in spite of existing a considerable number of methods for calibrating catadioptric systems [Wu and Hu, 2005, Barreto and Araujo, 2005, Gasparini et al., 2009, Puig et al., 2011] there are few methods for fisheye systems. A general method for central systems that can be also used for fisheyes is presented in [Scaramuzza et al., 2006]. In [Mei and Rives, 2007] the sphere model is used to approximate the fisheye model. A calibration method specific for fisheyes is presented in [Kannala and Brandt, 2006, Kannala et al., 2008a].This multi-view approach uses a series-based description of the radially symmetric model. The radially symmetric series-based model is also used in a plumb-line calibration method for central systems in [Tardif et al., 2006]. More recently, in [Alemán-Flores et al., 2014] line projections are detected in uncalibrated distorted images using a Hough transform approach.

In this chapter we present an automatic method for line extraction in uncalibrated images which is valid for central dioptric and catadioptric cameras with symmetry of revolution. This extraction allows us to segment the collection of edges corresponding to line-images automatically. For this, we exploit two different line-image constraints: point's location (which is the constraint typically used in the plum-line approach) and point's gradient constraint, using the gradients of the intensity field. Constraints based on gradient are particularly useful when the number of DOFs of the curve is high. From each extracted line-image we obtain the corresponding projecting plane since main calibration information is implicitly computed. In our method we do not assume any restriction in the orientation of 3D lines. The input of the method is a single image containing projections of lines. The output is a set of line-images and their supporting edges, the corresponding projection planes and the main calibration parameter of the system. The main contributions of this work are the following:

- **An explicit closed form solution of the constraints developed for the catadioptric sphere camera model and for different fisheye projection models.** By contrast with [Tardif et al., 2006] each projection model is treated independently obtaining a closed solution and reducing the number of parameters to encode the distortion.

- **A gradient-based method to reduce the minimal solution**. The proposed line-

image constraint is a representation of the plumb-line constraint of the image space. Working in the image space allows us to exploit the parallelism between the gradient of the intensity field and the gradient of the curve as additional independent constraint reducing the number of points needed to define a line projection when the calibration is unknown.

- **A unified parametrization for any central system with revolution symmetry**. We deal with different representations of central projections used for catadioptric and dioptric systems. Instead of calibrating all the parameters of these models we focus in a single calibration parameter $\hat{r}_{vl}$ which represents the radius of the vanishing line and it is common to all these models. Besides, we relate this parameter with the existing models and explain how to compute this relation whenever any central projection model can be expressed in the form $r = h(\phi)$.

- **A practical and robust method for automatic line-image extraction from uncalibrated omnidirectional images with revolution symmetry** . The proposed method has been implemented in a Matlab toolbox for evaluation and research purposes. This toolbox allows to extract the lines projections and the calibration of the system from a single-image.

The rest of the chapter is distributed as follows. In Section 3.2 we describe the covered catadioptric and fisheye projection models. In Section 3.3 we present the unified description to represent line-images when the system has revolution symmetry. In section 3.4 we present the constraints based on point's location and the constraints based on brightness gradient. Section 3.5 describes the algorithm to extract line-images from omnidirectional cameras without calibration. In section 3.6 we show the results of the experiments used to validate the method. Finally, in section 3.7 we briefly summarize the chapter.

3.2 Projection models for central systems with revolution symmetry

In this section we introduce the projection models covered by this work. When a projection system is central the projected rays lie on a common fixed point called viewpoint **O**. Assuming the system has revolution symmetry, the reference system of the camera has the origin in the viewpoint **O** and the Z-axis is aligned with the axis of revolution. Let **X** be a 3D point in homogeneous coordinates $\mathbf{X} = (X\ Y\ Z\ 1)^{\mathsf{T}}$. The point is projected onto a unitary sphere around the viewpoint **O** of the system. It is defined with two spherical coordinates ϕ and φ as, $\mathbf{x} = (\sin\phi\cos\varphi, \sin\phi\sin\varphi, \cos\phi)^{\mathsf{T}}$ (see Fig. 3.1).

Depending on the projection model this point is mapped on the image using different expressions. Notice that any point lying on the revolution axis is projected on an

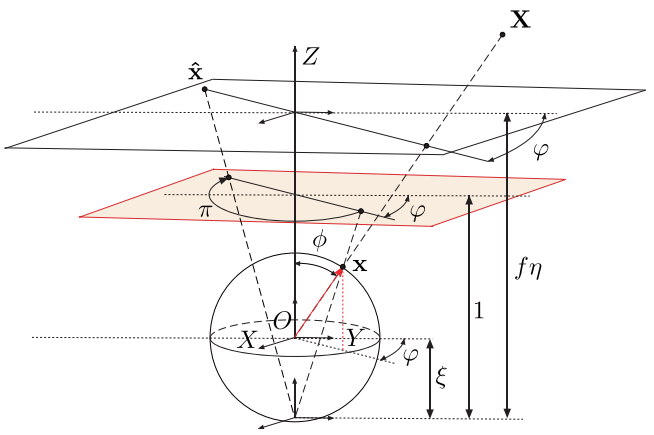
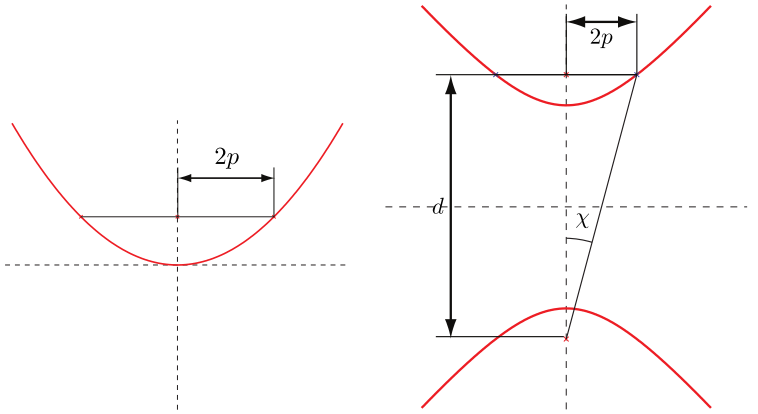


Figure 3.1: Catadioptric sphere camera model: The 3D point $\mathbf{X}$ is projected onto the sphere. Then this point is backprojected to a normalized plane through a virtual optic center located a distance ξ from the effective viewpoint. This point is transformed to the point $\hat{\mathbf{x}}$ on the centred image plane using the collineation $\mathsf{H_c} = \mathsf{diag}\,(f\eta, -f\eta, 1)$.



Parabolic Hyperbolic

	ξ	η
Parabolic:	1	$2p$
Hyperbolic:	$\cos\chi = \frac{d}{\sqrt{d^2+4p^2}}$	$\sin\chi = \frac{2p}{\sqrt{d^2+4p^2}}$
Planar:	0	1

Figure 3.2: Parameters of the unified sphere model for catadioptric systems.

image point called principal point. Consider the polar coordinates (r,θ) of an image point taking as reference the principal point. If the camera is correctly aligned with the axis of revolution we can observe that the coordinate θ is related with the spherical coordinate φ via the pixel aspect ratio k_{par}, as $\tan\theta = \pm k_{par}\tan\varphi$ (the sign in this expression is used to model reflections in catadioptric systems). Catadioptric and dioptric systems are projection systems that conserve the revolution symmetry. This means that the radius of a pixel $\hat{r}$ can be expressed in terms of the elevation angle ϕ. Depending on the device a different expression for $\hat{r} = h(\phi)$ is used.

In the following descriptions we assume that image points are expressed in a reference centred in the principal point. We also assume that pixel aspect ratio is equal to one which is valid in digital imagery. A point in this reference system is denominated with the notation $\hat{\mathbf{x}}$. The transformation from this reference to the final image coordinate system is the following,

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & s & u_0 \\ 0 & k_{par} & v_0 \\ 0 & 0 & 1 \end{pmatrix} \hat{\mathbf{x}} \,. \tag{3.1}$$

3.2.1 Projection models for catadioptric systems

Under the sphere camera model [Geyer and Daniilidis, 2000, Baker and Nayar, 1999, Geyer and Daniilidis, 2001] all central catadioptric systems can be modelled by a projection to the unitary sphere followed by a perspective projection via a virtual viewpoint located a distance ξ from the effective viewpoint (see Fig. 3.1). Let $\hat{\mathbf{x}} = (\hat{x},\hat{y},1)^T$ be a point on an image referred to the principal point and given the spherical coordinates ϕ and φ of the corresponding point on the unitary sphere then,

$$\hat{x} = \frac{f\eta\sin\phi\cos\varphi}{\cos\phi+\xi} \quad \text{and} \quad \hat{y} = -\frac{f\eta\sin\phi\sin\varphi}{\cos\phi+\xi} \,. \tag{3.2}$$

In polar coordinates the point is described by $\hat{\theta} = -\varphi$ and

$$\hat{r} = \frac{f\eta\sin\phi}{\cos\phi+\xi} = \frac{f\eta\tan\phi}{1+\xi\sqrt{\tan^2\phi+1}} \,. \tag{3.3}$$

The geometry of the projection system is described by parameters ξ and η which have a different definition depending on the system type (see Fig. 3.2).

3.2.2 Fisheye models

Several models are used to describe point projection in dioptric systems depending on the manufacturing procedure of the lens [Kingslake, 1989, Stevenson and Fleck, 1996, Ray, 2002]. Assuming square pixel, these models are expressed in polar coordinates $(\hat{r},\hat{\theta})$. For all these models $\hat{\theta} = \varphi$ and the radius changes depending on the camera

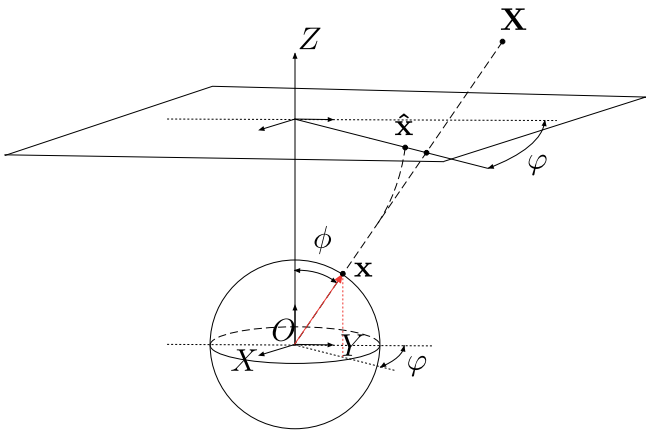


Figure 3.3: Fisheye camera models: The radius of the point on the image is distorted by a function $\hat{r} = h(\phi)$.

Equiangular Fisheye	Stereographic Fisheye	Orthogonal Fisheye	Equisolid Fisheye
$f\phi$	$2f\tan\left(\frac{\phi}{2}\right)$	$f\sin(\phi)$	$2f\sin\left(\frac{\phi}{2}\right)$

Table 3.1: Function $\hat{r} = h(\phi)$ describing fisheye projection models.

type (see Table 3.1). The parameter f of these models does not represent the focal distance of the camera like in the case of the parameter f of the sphere camera model presented in the previous section.

Notice that the stereographic projection is equivalent to the projection in a paracatadioptric system. The stereographic projection is also equivalent to the "division model" model of Fitzgibbon [Fitzgibbon, 2001] when using a single parameter ($r' = k_1 \frac{r}{1-k_2^2}$).

3.3 Unified description for line projection in central systems with revolution symmetry

In this section we present a unified description for line-images in dioptric and catadioptric imagery, which is valid for central systems with revolution symmetry.

Let $\Pi = (n_x, n_y, n_z, 0)^\mathsf{T}$ be a plane defined by a 3D line and the viewpoint of the system $\mathbf{O}$. The projected line associated to the 3D line can be represented by $\mathbf{n} = (n_x, n_y, n_z)^\mathsf{T}$. Then, the points $\mathbf{X}$ lying in the 3D line are projected to points $\mathbf{x}$. These points satisfy $\mathbf{n}^\mathsf{T}\mathbf{x} = 0$. Using the spherical representation and assuming that $\hat{\theta} = \pm\varphi$

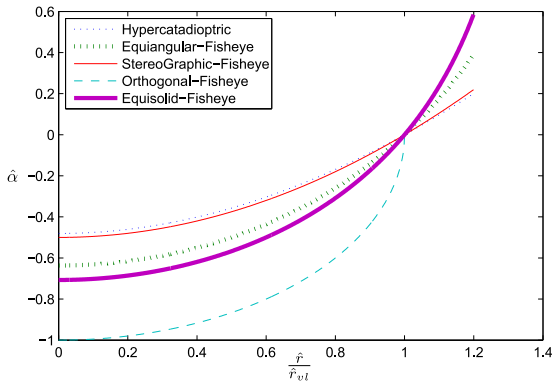


Figure 3.4: Representation of $\hat{\alpha}$ depending on $\hat{r}$ for different central projection systems with revolution symmetry. $\hat{r}$ is normalized with respect to the radius of the projection of the vanishing line $\hat{r}_{vl}$.

(square pixel) this equality could be expressed as

$$\sin\phi\,(n_x\hat{x} \pm n_y\hat{y}) + n_z\hat{r}\cos\phi = 0\ . \tag{3.4}$$

With the change of variable $\hat{\alpha} = \dfrac{n_x\hat{x} \pm n_y\hat{y}}{n_z}$ we can isolate the model parameters from the normal describing the line, obtaining the expression,

$$\hat{\alpha} = -\hat{r}\cot\phi\ . \tag{3.5}$$

Notice that $\hat{\alpha} = \hat{\alpha}(\hat{r})$, as a result of $\phi = h^{-1}(\hat{r})$ when we have symmetry of revolution and square pixel. Therefore, the constraint for points on the line projection in image coordinates for systems with symmetry of revolution is

$$n_x\hat{x} \pm n_y\hat{y} - n_z\hat{\alpha}(\hat{r}) = 0\ , \tag{3.6}$$

where $\hat{\alpha}$ is a different expression for each camera model depending on the radius and the model parameters (see Table 3.2).

3.3.1 Line-image curve representation and unified main calibration parameter

Equation (3.6) is the homogeneous representation of the line projection on the image. There exist two particular cases common to all the projection models showed above. First we have the case in which 3D lines are coplanar to the revolution axis. In this case

	$\hat{\alpha}$	$\hat{r}_{vl}$	polynomial
Perspective	f	∞	straight
Para catadioptric	$\frac{\hat{r}^2}{4fp} - fp$	$2fp$	conic
Hyper catadioptric	$\frac{-f+\cos\chi\sqrt{\hat{r}^2+f^2}}{\sin\chi}$	$f\tan\chi$	conic
Equiangular-Fisheye	$-\hat{r}\cot\frac{\hat{r}}{f}$	$f\frac{\pi}{2}$	non polynomial
Stereographic-Fisheye	$\frac{\hat{r}^2}{4f} - f$	$2f$	conic
Orthogonal-Fisheye	$-\sqrt{f^2-\hat{r}^2}$	f	conic
Equisolid-Fisheye	$\frac{2\hat{r}^2-f^2}{2\sqrt{f^2-\hat{r}^2}}$	$f\frac{\sqrt{2}}{2}$	quartic

Table 3.2: Parameters for different central projection systems with revolution symmetry.

$n_z = 0$ and the resulting line-image is a radial straight line passing through the principal point, described as

$$n_x\hat{x} \pm n_y\hat{y} = 0\ . \tag{3.7}$$

The second particular case happens when $\mathbf{n} = (0,0,1)^\mathsf{T}$. In this case the line-image is the projection of the vanishing line. This projection is a circle centred at principal point and with radius $\hat{r}_{vl}$. This radius depends only on the system geometry (see Table 3.2) and can be used as a main calibration parameter independently of the camera system. In this case, the line-image equation has the form,

$$\hat{\alpha}\,(\hat{r}_{vl}) = 0\ . \tag{3.8}$$

In Fig. 3.4 we show a comparison among different $\hat{\alpha}\,(\hat{r})$ functions for different central projection systems. The radius $\hat{r}$ has been normalized with respect to $\hat{r}_{vl}$. For the case of the orthogonal system, $\hat{\alpha}$ only makes sense when $\hat{r} < \hat{r}_{vl}$.

The general form for a line-image is a curve. In most cases, these curves can be also expressed as polynomials. The catadioptric case has been deeply studied in [Barreto and Araujo, 2005], and it has been proven that the line-image is a conic. The stereographic case is equivalent to the paracatadioptric projection therefore the corresponding line-image is a conic encoded with the parameters of the sphere model. The

orthogonal-fisheye line-image is also a conic but it is not encoded with the parametrization of the sphere model. The equisolid line-image is a quartic (see Section 3.3.5). For other cases in general the curve is not a polynomial.

Each line-image is the projection of any 3D line contained in a projection plane which is described by the normal vector $\mathbf{n}$. Consider the orientation of the central system is aligned with the vertical axis Z (see Fig. 3.5 (a)) and the parametrization of this normal is $\mathbf{n} = (\cos\Phi\cos\Theta, \cos\Phi\sin\Theta, \sin\Phi)^{\mathsf{T}}$ with the elevation angle Φ, and the azimuth angle Θ (do not confuse with the angles of the projecting ray of a point ϕ and φ). Given a fixed value of Φ the relation between two different line-images with different azimuth angle Θ is a rotation around the principal point. However, given a fixed value of Θ a variation in the elevation angle Φ implies a change in the curvature of the line-image. In Fig. 3.5, we show a parametric representation of line-images for different central projection systems with revolution symmetry. We can imagine the projected line-image, as the projection of the intersection of the projecting plane with any sphere centred on the reference system. We represent different line-images with a fixed value of Θ but increasing the elevation angle Φ from 0 to $\frac{\pi}{2}$. (see Fig. 3.5 (a)). To compare the different systems each image has been simulated for a different model but with the same $\hat{r}_{vl}$. The outside region of this circle corresponds to a FOV greater than 180 deg and the inside with a FOV lesser than 180 deg. As in the example the system is oriented from bottom to top the inside of the circle corresponds to points with $Z > 0$ and the outside the circle corresponds with points with $Z < 0$. Notice that any line-image pass through both regions, as the projection plane intersects the both regions in the 3D space. This parametric representation corresponds to the projection plane covering 360 deg around the axis of revolution. An infinity 3D segment only fulfils a part of the curve covering 180 deg.

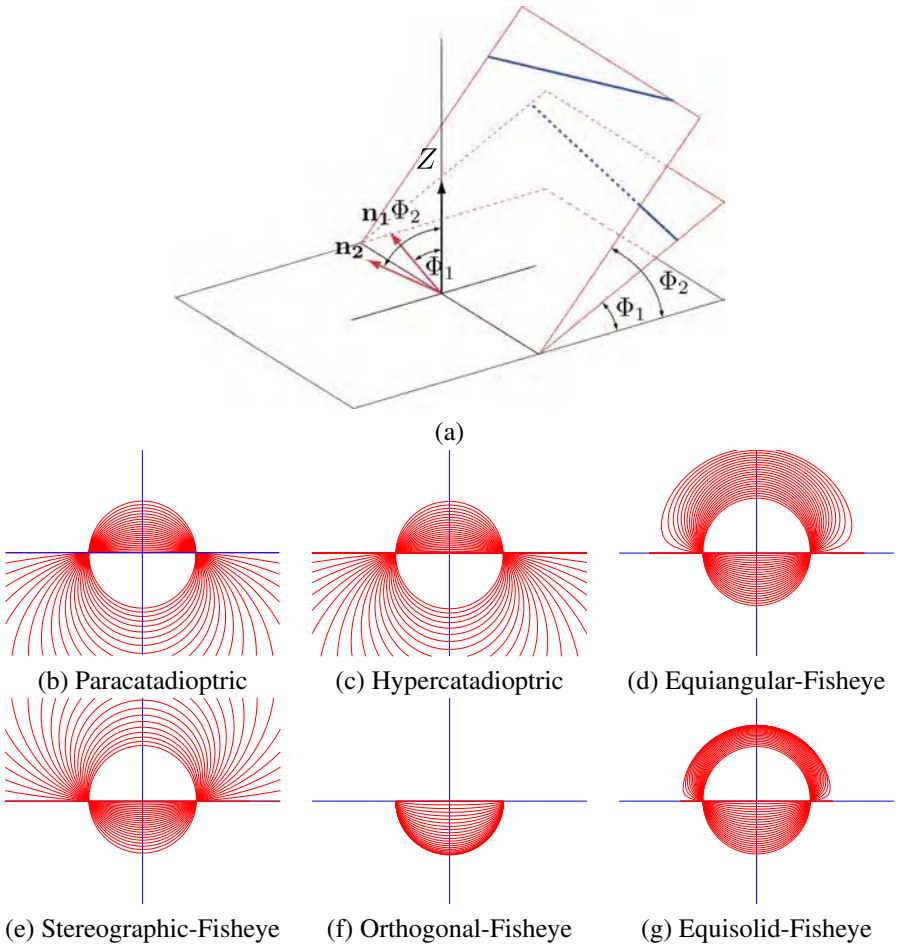
3.3.2 The sphere catadioptric model as fisheye model

Some authors have used the catadioptric sphere model to calibrate fisheye models [Ying and Hu, 2004a, Courbon et al., 2007]. The stereographic projection is equivalent to the paracatadioptric projection and it can be directly encoded using the sphere model. From the definition of Table 3.1 and using the half-angle formula,

$$\hat{r} = 2f\tan\left(\frac{\phi}{2}\right) = 2f\frac{\sin\phi}{\cos\phi + 1} \tag{3.9}$$

which is the equation (3.3) with $\xi = 1$ and $\eta = 2$. For other cases [Ying and Hu, 2004a,Courbon et al., 2007] propose to use the sphere model with $\xi > 1$ (in catadioptric systems where $0 < \xi < 1$).

In Fig. 3.6 we show a comparison of the line-images of an equiangular-fisheye system using the equiangular-fisheye projection model and the catadioptric sphere model showing the limitations of using the catadioptric sphere model for fisheyes. In order to illustrate these limitations in more detail, in Fig. 3.7 we represent the line projection of

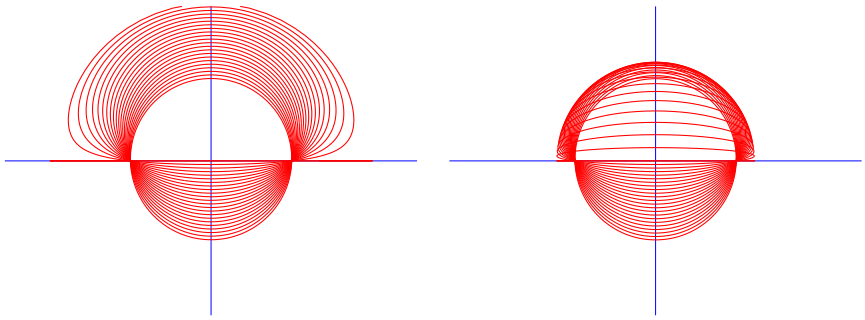


(a)

(b) Paracatadioptric (c) Hypercatadioptric (d) Equiangular-Fisheye

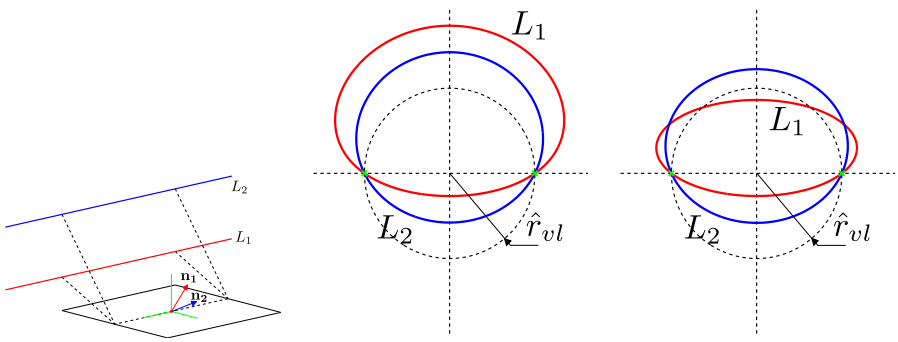
(e) Stereographic-Fisheye (f) Orthogonal-Fisheye (g) Equisolid-Fisheye

Figure 3.5: Representation of line-images on the image plane depending on the projection model and with different values of the elevation angle Φ of the normal $\mathbf{n}$ representing the projection plane. (a) Elevation of the projection plane. (b-g) Line-images when increasing the elevation angle Φ for different projection models.



(a) (b)

Figure 3.6: Comparison of line-images on the image plane with an equiangular-fisheye system using (a) the equiangular-fisheye projection model and (b) the catadioptric sphere model with $\xi > 1$. Each line-image corresponds to a different value of the elevation of the normal $\mathbf{n}$.



(a) (b) (c)

Figure 3.7: (a) Isometric view of two parallel lines. Comparison of line-images of two parallel lines with an equiangular-fisheye system using the equiangular-fisheye projection model (b) and using the catadioptric sphere model (c). In green the vanishing points of direction X.

two parallel lines. In Fig. 3.7 (b) we show the line projection of these lines using the equiangular-fisheye projection model. The dotted circle represents the projection of the vanishing line. The region inside this circle corresponds to a FOV lesser than 180 deg and the projections of 3D points with $Z > 0$. The region outside this circle corresponds to a FOV greater than 180 deg and the projections of 3D points with $Z < 0$. Consider the line-image L_1. The part of L_1 located inside the circle corresponds with the part of the plane with $Z > 0$ and the part of L_1 located outside the circle corresponds with the part of the plane with $Z < 0$. As the 3D lines are parallel the line-images intersects in two points corresponding with the vanishing points (in green) of the direction of the lines. One of the points corresponds to the positive direction and the other with the negative. In Fig. 3.7 (c) we show the projection of these lines using the catadioptric sphere model with $\xi > 1$ proposed in [Ying and Hu, 2004a, Courbon et al., 2007]. Consider the line-image L_1. The part of the plane with $Z > 0$ is projected inside the circle, however the part of the plane with $Z < 0$ is projected in both regions: inside and outside the circle. That means that the line-image does not correctly fit the line projection when the FOV is greater than 180 deg. In addition, the line-images intersect in four points instead of two giving a sense of non-geometric coherence: two of the intersections are the vanishing points and two are points without geometric sense.

3.3.3 The homogeneous line-image equation as a measure of distance

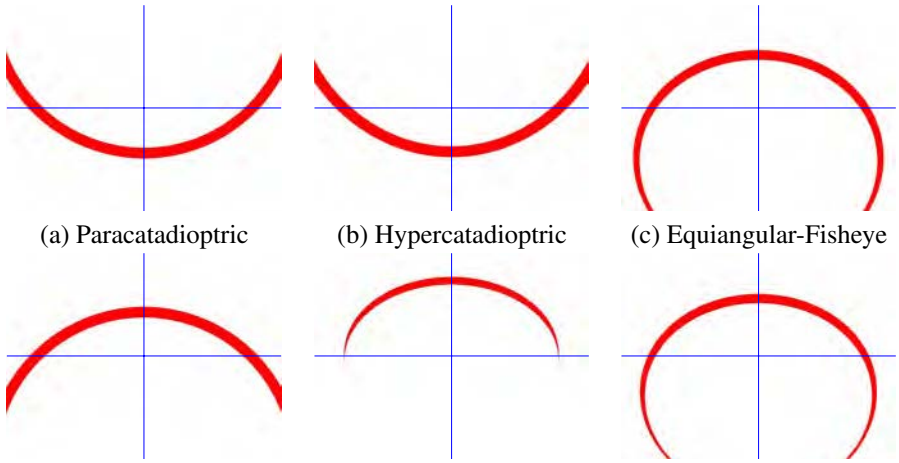
The homogeneous expression of the line-image (3.6) defines a family of curves located to an algebraic distance from the original curve.

$$d(\hat{x},\hat{y}) = n_x\hat{x} \pm n_y\hat{y} - n_z\hat{\alpha} \; . \tag{3.10}$$

This algebraic distance is an approximation of the metric distance from a point to the line-image and is defined in pixels. In Fig. 3.8 we show the region defined by a fixed threshold of 20 pixels around a given line-image for different systems. The thickness of this region is homogeneous for catadioptric and stereographic systems but it is not completely homogeneous for equiangular, orthogonal and equisolid systems. However, in general this distance can be used to discriminate if a point belongs to a line-image.

Consider for example the algebraic distance based on polynomials (e.g. for hypercatadioptric systems $d = \sqrt{\mathbf{x}^\mathsf{T}\Omega_{cata}\mathbf{x}}$, see Section 3.3.5). Given a fixed threshold, the region around the conic has a different thickness depending on the elevation angle of the vector $\mathbf{n}$. With our proposal the distance is a good approximation in regions close to the line-image.

In Fig. 3.9 (a) we show a comparison between the distance of a point to the line-image (blue dotted) and the proposed algebraic distance (red) for hypercatadioptric images. The algebraic distance approximates the real distance in regions which are



(a) Paracatadioptric (b) Hypercatadioptric (c) Equiangular-Fisheye

(d) Stereographic-Fisheye (e) Orthogonal-Fisheye (f) Equisolid-Fisheye

Figure 3.8: Region around a line-image defined by a fixed distance of 20 pixels.

close to the line-image, therefore can be used to discriminate if a point lies on a line-image or not. In Fig. 3.9 (b) we show the same comparison but using the algebraic distance defined by the expression of a conic on the image ($d = \sqrt{\mathbf{x}^\mathsf{T}\Omega_{cata}\mathbf{x}}$). We can see how this distance does not approximate well the metric distance in regions close to the curve. We also show that this distance is lower than the metric distance in vertical lines but higher when the lines are horizontal. In practice that means that the thickness of a region defined by a threshold varies considerably if elevation of $\mathbf{n}$ changes.

3.3.4 Line-image definition from two points

Given at least two points lying on a line-image described by (3.6) we can obtain the normal $\mathbf{n}$ by solving the homogeneous linear system

$$\begin{pmatrix} \hat{x}_i & \pm\hat{y}_i & -\hat{\alpha}_i \end{pmatrix}\mathbf{n} = 0 \quad \text{for} \quad i = 1,...,n \quad \text{with} \quad n \geq 2. \tag{3.11}$$

where $\hat{\alpha}$ is a different expression for each camera system (see Table 3.2) and the sign '$\pm$' is positive for dioptric systems and negative for catadioptrics. The system is solved by using a Singular Value Decomposition. In particular for the minimal case of two points and solving for $\mathbf{n}$ we have

$$\mathbf{n} = \begin{pmatrix} \hat{y}_1\hat{\alpha}_2 - \hat{y}_2\hat{\alpha}_1 \\ \pm(\hat{x}_2\hat{\alpha}_1 - \hat{x}_1\hat{\alpha}_2) \\ \hat{x}_2\hat{y}_1 - \hat{x}_1\hat{y}_2 \end{pmatrix}. \tag{3.12}$$

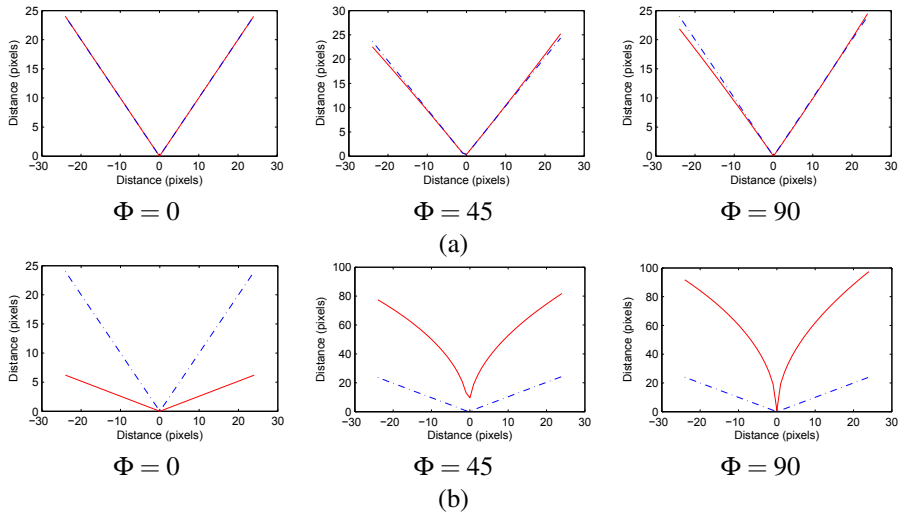


Figure 3.9: Comparison between metric distance (blue dotted) and algebraic distances (red solid): (a) Our proposal (3.10). (b) Conic based algebraic distance.

3.3.5 Polynomials describing line-images

Some of the line-images in central systems with revolution symmetry can be expressed as polynomials. In this Section we show the description of these line images using polynomials for Catadioptric systems, Equisolid-fisheye, Stereographic-fisheye, Orthogonal-fisheye and Equisolid-fisheye systems.

Catadioptric and Stereographic-fisheye

$$\begin{pmatrix} \hat{x}^2 & \hat{x}\hat{y} & \hat{y}^2 & \hat{x} & \hat{y} & 1 \end{pmatrix} \Omega_{cata} = 0 \tag{3.13}$$

where

$$\Omega_{cata} = \begin{pmatrix} n_x^2 \sin\chi^2 - n_z^2 \cos\chi^2 \\ 2n_x n_y \sin\chi^2 \\ n_y^2 \sin\chi^2 - n_z^2 \cos\chi^2 \\ 2f \sin\chi n_x n_z \\ 2f \sin\chi n_y n_z \\ f^2 \sin\chi^2 n_z^2 \end{pmatrix} \tag{3.14}$$

Ortogonal-Fisheye

$$\begin{pmatrix} \hat{x}^2 & \hat{x}\hat{y} & \hat{y}^2 & \hat{x} & \hat{y} & 1 \end{pmatrix} \Omega_{ortho} = 0 \tag{3.15}$$

where

$$\Omega_{ortho} = \begin{pmatrix} -n_x^2 - n_z^2 \\ -2n_x n_y \\ -n_y^2 - n_z^2 \\ 0 \\ 0 \\ f^2 n_z^2 \end{pmatrix} \tag{3.16}$$

Equisolid-Fisheye

$$\begin{pmatrix} \hat{x}^4 & \hat{x}^3\hat{y} & \hat{x}^2\hat{y}^2 & \hat{x}^2 & \hat{x}\hat{y}^3 & \hat{x}\hat{y} & \hat{y}^4 & \hat{y}^2 & 1 \end{pmatrix} \mathbf{Q} = 0 \tag{3.17}$$

where

$$\mathbf{Q} = \begin{pmatrix} -4\left(n_x^2 + n_z^2\right) \\ -8n_x n_y \\ -4n_x^2 - 4n_y^2 - 8n_z^2 \\ 4f^2\left(n_x^2 + n_z^2\right) \\ -8n_x n_y \\ 8f^2 n_x n_y \\ -4\left(n_y^2 + n_z^2\right) \\ 4f^2\left(n_y^2 + n_z^2\right) \\ -f^4 n_z^2 \end{pmatrix} \tag{3.18}$$

3.4 The straight-line constraint on the image

In Section 3.3 we have presented an unified line-image description of the line projection in systems with revolution symmetry. We have also presented the unified main calibration parameter $\hat{r}_{vl}$ encoding the calibration of these systems. In this section we use this description to develop and to particularize the plumb-line constraint for the different projection models presented in Section 3.2. The goal is computing both, the main calibration parameter $\hat{r}_{vl}$ and the projecting plane of the line $\mathbf{n}$. The results are two kind of constraints: location based constraints which allow to compute the line-image and $\hat{r}_{vl}$ from the coordinates of at least three points and gradient based constraints which allow to compute the line-image and $\hat{r}_{vl}$ from the coordinates and the gradients of a minimum of two points.

3.4.1 Location-based line-image constraint

As presented in Section 3.3, the projecting plane Π describing the line projection is completely defined by two points and the view-point. That implies that the calibration of the system is embedded in the geometry of this curve. If the calibration of the

(a) Planar	$(l_1+l_2+l_3)=0$
(b) Paracatadioptric	$\hat{r}_{vl}=\sqrt{\frac{l_1\hat{r}_1^2+l_2\hat{r}_2^2+l_3\hat{r}_3^2}{l_1+l_2+l_3)}}$
(c) Hypercatadioptric	$\hat{r}_{vl}=\sqrt{\left(\frac{l_1\sqrt{\hat{r}_1^2+f^2}+l_2\sqrt{\hat{r}_2^2+f^2}+l_3\sqrt{\hat{r}_3^2+f^2}}{l_1+l_2+l_3}\right)^2-f^2}$
(d) Equiangular-Fisheye	$l_1\hat{r}_1\cot\left(\frac{\pi}{2}\frac{\hat{r}_1}{\hat{r}_{vl}}\right)+l_2\hat{r}_2\cot\left(\frac{\pi}{2}\frac{\hat{r}_2}{\hat{r}_{vl}}\right)+l_3\hat{r}_3\cot\left(\frac{\pi}{2}\frac{\hat{r}_3}{\hat{r}_{vl}}\right)=0$
(e) Stereographic-Fisheye	$\hat{r}_{vl}=\sqrt{\frac{l_1\hat{r}_1^2+l_2\hat{r}_2^2+l_3\hat{r}_3^2}{l_1+l_2+l_3}}$
(f) Orthogonal-Fisheye	$l_1\sqrt{\hat{r}_{vl}^2-\hat{r}_1^2}+l_2\sqrt{\hat{r}_{vl}^2-\hat{r}_2^2}+l_3\sqrt{\hat{r}_{vl}^2-\hat{r}_3^2}=0$
(g) Equisolid-Fisheye	$l_1\frac{\hat{r}_1^2-\hat{r}_{vl}^2}{\sqrt{2\hat{r}_{vl}^2-\hat{r}_1^2}}+l_2\frac{\hat{r}_2^2-\hat{r}_{vl}^2}{\sqrt{2\hat{r}_{vl}^2-\hat{r}_2^2}}+l_3\frac{\hat{r}_3^2-\hat{r}_{vl}^2}{\sqrt{2\hat{r}_{vl}^2-\hat{r}_3^2}}=0$

Table 3.3: Three points line-image constraint for different central projection systems with revolution symmetry.

system is known, the projecting plane normal **n** can be recovered from two image points (3.11). However, when constraint (3.6) is satisfied the line-image can be recovered even if camera calibration is unknown. Notice that when the number of equations defining (3.11) is greater than two we obtain a redundant system. Therefore when having three points lying on a line-image the rank of the homogeneous matrix must be two. Imposing this condition we find a constraint using three points of a line image. The line-image constraint can be written as

$$l_1\hat{\alpha}_1+l_2\hat{\alpha}_2+l_3\hat{\alpha}_3=0, \tag{3.19}$$

where $l_1=\hat{x}_2\hat{y}_3-\hat{x}_3\hat{y}_2$, $l_2=\hat{x}_3\hat{y}_1-\hat{x}_1\hat{y}_3$ and $l_3=\hat{x}_1\hat{y}_2-\hat{x}_2\hat{y}_1$.

Thus, with three points lying on the line-image and equation (3.19) it is possible to obtain the main calibration parameter $\hat{r}_{vl}$. Substituting the expression of $\hat{\alpha}$ in In Table 3.3 we show the corresponding expressions for each system which are computed by substituting the expression of $\hat{\alpha}$ in (3.19) and expressing them in terms of $\hat{r}_{vl}$.

- **Computation of $\hat{r}_{vl}$ in perspective systems.** It is not possible to extract any additional information (Table 3.3 (a)).
- **Computation of $\hat{r}_{vl}$ in paracatadioptric and stereographic systems systems.** In these cases the radius $\hat{r}_{vl}$ is directly computed (Table 3.3 (b)(e)). Notice that, as $\hat{r}_{vl}$ must be positive only the positive solution of the root square is a valid solution.

- **Computation of $\hat{r}_{vl}$ in equiangular-fisheye systems.** For equiangular-fisheye systems the constraint (Table 3.3 (d)) is solved by minimization. To initialize this minimization the value of $\hat{r}_{vl}$ is computed from a linear approximation of the constraint

$$\hat{r}_{vl} \approx \frac{\pi}{2}\sqrt{\frac{1}{3}\frac{\left(l_1\hat{r}_1^2 + l_2\hat{r}_2^2 + l_3\hat{r}_3^2\right)}{(l_1+l_2+l_3)}}\,. \tag{3.20}$$

  In Fig. 3.10 (a) we show some examples of line-images defined with 3 points in equiangular-fisheye systems.

- **Computation of $\hat{r}_{vl}$ in orthogonal-fisheye systems.** The equation in table 3.3 (f) becomes to the bi-quadratic polynomial equation

$$c_1\hat{r}_{vl}^4 + 2c_2\hat{r}_{vl}^2 + c_3 = 0 \tag{3.21}$$

  where

$$c_1 = \prod_{i=0}^{3}\sum_{j=1}^{3}\left((-1)^{\delta_{i,j}+1}\,l_j\right) \tag{3.22}$$

$$c_2 = \sum_{i=1}^{3} l_i^2\hat{r}_i^2 \sum_{j=1}^{3}\left((-1)^{\delta_{i,j}}\,l_j^2\right) \tag{3.23}$$

$$c_3 = \prod_{i=0}^{3}\sum_{j=1}^{3}\left((-1)^{\delta_{i,j}+1}\,l_j\hat{r}_j\right) \tag{3.24}$$

  $\delta_{i,j}$ being the Kronecker delta.

  This equation has direct solution for $\hat{r}_{vl}$. Notice that despite the bi-quadratic equation has a priori four solutions, two of them are always negative ($\hat{r}_{vl} = \pm\sqrt{\hat{r}_{vl}^2}$) and in practice the wrong one can be discarded using the equation of Table 3.3 (f) and rejecting the non-real solutions.

- **Computation of $\hat{r}_{vl}$ in equisolid-fisheye systems.** In equisolid-fisheye systems the equation in table 3.3 (g) becomes to a bi-eight degree equation which has a direct solution for $\hat{r}_{vl}$. The polynomial equation is

$$\sum_{m=0}^{8}\omega_m\hat{r}_{vl}^{2m} = 0 \tag{3.25}$$

  where $\omega_m = \omega_{m,123} + \omega_{m,213} + \omega_{m,312}$ and

Para-Catadioptric	Hyper-Catadioptric	Equiangular-Fisheye
$\frac{1}{2fp}$	$\frac{\cot\chi}{\sqrt{\hat{r}^2+f^2}}$	$\frac{1}{f}\left(1-\frac{f}{\hat{r}}\cot\frac{\hat{r}}{f}+\cot^2\frac{\hat{r}}{f}\right)$
Stereographic-Fisheye	**Orthogonal-Fisheye**	**Equisolid-Fisheye**
$\frac{1}{2f}$	$\frac{1}{\sqrt{f^2-\hat{r}^2}}$	$\frac{\hat{r}\left(3f^2-2\hat{r}^2\right)}{2\left(f^2-\hat{r}^2\right)^{3/2}}$

Table 3.4: $\frac{\partial\hat{\alpha}}{\partial\hat{r}}\frac{1}{\hat{r}}$ for different central projection systems with revolution symmetry.

$$\omega_{0,ijk} = -l_i^4\hat{r}_i^8\hat{r}_j^4\hat{r}_k^4 + 2\hat{r}_i^4 l_j^2 l_k^2\hat{r}_j^6\hat{r}_k^6 \tag{3.26}$$

$$\omega_{1,ijk} = 4\left(\hat{r}_i^2\left(\hat{r}_j^2+\hat{r}_k^2\right)+\hat{r}_j^2\hat{r}_k^2\right)\left(l_i^4\hat{r}_i^6\hat{r}_k^2 - 2l_j^2 l_k^2\hat{r}_i^2\hat{r}_j^4\hat{r}_k^4\right) \tag{3.27}$$

$$\begin{aligned}\omega_{2,ijk} = &-16l_i^4\hat{r}_i^4\left(\hat{r}_i^2\hat{r}_j^2\hat{r}_k^2\left(\hat{r}_j^2+\hat{r}_k^2\right)+\hat{r}_i^4\hat{r}_j^2\hat{r}_k^2\right)+\ldots\\ &+32l_j^2 l_k^2\hat{r}_j^2\hat{r}_k^2\left(\hat{r}_i^2\hat{r}_j^2\hat{r}_k^2\left(\hat{r}_j^2+\hat{r}_k^2\right)+\hat{r}_i^4\hat{r}_j^2\hat{r}_k^2\right)+\ldots\\ &-2l_i^4\hat{r}_i^4\left(2\hat{r}_i^4\left(\hat{r}_j^4+\hat{r}_k^4\right)+3\hat{r}_j^4\hat{r}_k^4\right)+\ldots\\ &+10l_j^2 l_k^2\hat{r}_j^2\hat{r}_k^2\hat{r}_i^2\left(\hat{r}_j^4+\hat{r}_k^4\right)+\ldots\\ &8l_j^2 l_k^2\hat{r}_j^6\hat{r}_k^6\end{aligned} \tag{3.28}$$

$$\begin{aligned}\omega_{3,ijk} = &16l_i^4\hat{r}_i^6\left(\hat{r}_i^2\hat{r}_j^2+\hat{r}_i^2\hat{r}_k^2+\hat{r}_j^4+\hat{r}_k^4+4\hat{r}_j^2\hat{r}_k^2\right)+\ldots\\ &-8l_i^2\hat{r}_i^6\left(4\left(l_j^2\hat{r}_j^4+l_k^2\hat{r}_k^2\right)+5\hat{r}_j^2\hat{r}_k^2\left(l_j^2+l_k^2\right)\right)+\ldots\\ &-8\hat{r}_i^2\hat{r}_j^4\hat{r}_k^4\left(5l_i^2\left(l_j^2+l_k^2\right)+16l_j^2 l_j^2\right)+\ldots\\ &+4l_i^4\left(6\hat{r}_i^4\hat{r}_j^2\hat{r}_k^2\left(\hat{r}_j^2+\hat{r}_k^2\right)+\hat{r}_i^2\hat{r}_j^4\hat{r}_k^4\right)+\ldots\\ &-4l_i^2\hat{r}_i^6\left(l_j^2\hat{r}_k^4+l_k^2\hat{r}_j^4\right)\end{aligned} \tag{3.29}$$

$$\begin{aligned}\omega_{4,ijk} = &+128l_j^2l_k^2\hat{r}_j^4\hat{r}_k^4 + 160\hat{r}_i^4l_i^2\hat{r}_j^2\hat{r}_k^2\left(l_j^2+l_k^2\right)+...\\ &-l_i^4\left(\hat{r}_j^4\hat{r}_k^4+64\hat{r}_i^6\left(\hat{r}_j^2+\hat{r}_k^2\right)\right)+...\\ &-16l_i^4\left(\hat{r}_i^8+\hat{r}_i^2\hat{r}_j^2\hat{r}_k^2\left(\hat{r}_j^2+\hat{r}_k^2\right)\right)+...\\ &+4\hat{r}_i^6l_i^2\left(4\left(l_j^2\hat{r}_k^2+l_k^2\hat{r}_j^2\right)+10\left(l_j^2\hat{r}_j+l_k^2\hat{r}_k^2\right)\right)+...\\ &\hat{r}_i^4l_j^2l_k^2\left(16\left(\hat{r}_j^2+\hat{r}_k^2\right)+50\hat{r}_j^2\hat{r}_k^2\right)+...\\ &-24l_i^4\hat{r}_i^4\left(\left(\hat{r}_j^4+\hat{r}_k^4\right)+4\hat{r}_j^2\hat{r}_k^2\right)\end{aligned} \tag{3.30}$$

$$\begin{aligned}\omega_{5,ijk} = &\,64l_i^2\hat{r}_i^2\left(l_i^2\hat{r}_i^4+l_i^2\hat{r}_j^2\hat{r}_k^2-\hat{r}_i^2\left(l_j^2\hat{r}_k^2+l_k^2\hat{r}_j^2\right)\right)+...\\ &+32l_i^2\hat{r}_i^2\left(3l_i^2\hat{r}_i^2\left(\hat{r}_j^2+\hat{r}_k^2\right)-5\hat{r}_i^2\left(l_j^2\hat{r}_j^2+l_k^2\hat{r}_k^2\right)\right)+...\\ &-200l_j^2l_k^2\hat{r}_i^2\hat{r}_j^2\hat{r}_k^2+...\\ &16l_i^2\hat{r}_i^2\left(l_i^2\left(\hat{r}_j^4+\hat{r}_k^4\right)-\hat{r}_i^4\left(l_j^2+l_k^2\right)\right)+...\\ &+\left(\hat{r}_j^2+\hat{r}_k^2\right)\left(4l_i^4\hat{r}_j^2\hat{r}_k^2-20\hat{r}_i^4l_j^2l_k^2\right)\end{aligned} \tag{3.31}$$

$$\begin{aligned}\omega_{6,ijk} = &\,64l_i^2\hat{r}_i^2\left(-l_i^2\left(\hat{r}_j^2+\hat{r}_k^2\right)+\hat{r}_i^2\left(l_j^2+l_k^2\right)\right)+...\\ &16l_i^2\hat{r}_i^2\left(-6l_i^2\hat{r}_i^2+5\left(l_j^2\hat{r}_k^2+l_k^2\hat{r}_j^2\right)\right)+...\\ &+4\left(50l_j^2\hat{r}_j^2l_k^2\hat{r}_k^2-4l_i^4\hat{r}_j^2\hat{r}_k^2-l_i^4\left(\hat{r}_j^4+\hat{r}_k^4\right)+2\hat{r}_i^4l_j^2l_k^2\right)\end{aligned} \tag{3.32}$$

$$\omega_{7,ijk} = 16\left(4l_i^4\hat{r}_i^2+l_i^4\hat{r}_j^2+l_i^4\hat{r}_k^2-2\hat{r}_i^2l_j^2l_k^2-5l_i^2\hat{r}_i^2\left(l_j^2+l_k^2\right)\right) \tag{3.33}$$

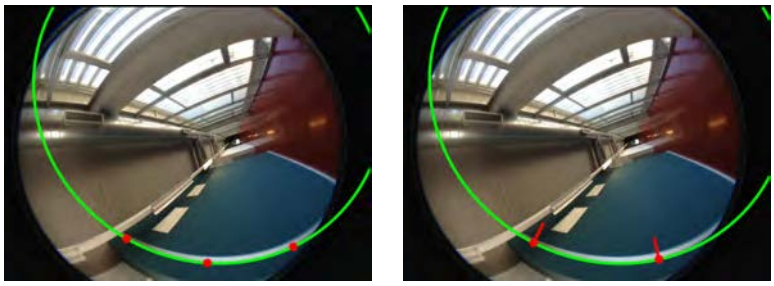
$$\omega_{8,ijk} = 16\left(-l_i^4+2l_j^2l_k^2\right) \tag{3.34}$$

- **Computation of $\hat{r}_{vl}$ in hypercatadioptric systems .**

  In hypercatadiopric systems (Table 3.3 (c)) the focal distance of the camera is needed to compute the main calibration parameter $\hat{r}_{vl}$. This is because two calibration parameters (f and χ) are involved and they cannot be coupled to reduce a degree of freedom.

  When f is known the equation in Table 3.3 (d) is solved for χ and it yields

$$\cos\chi = \frac{f\left(l_1+l_2+l_3\right)}{l_1\sqrt{\hat{r}_1^2+f^2}+l_2\sqrt{\hat{r}_2^2+f^2}+l_3\sqrt{\hat{r}_3^2+f^2}}\,. \tag{3.35}$$



(a) (b)

Figure 3.10: (a) Extraction of a line-image from three points.(b) Extraction of a line-image from two points using gradient constraint (in red the direction of ∇I). In this example, the linear approximation is used for the two points method.

The focal distance f can be directly computed using gradient constraints (see Section 3.4.4) although the method is quite unstable. In practice it is possible to complete the extraction of the line-image with a rough approximation of f.

3.4.2 Gradient-based line-image constraint

Here we present the line-image gradients for central systems with revolution symmetry, which gives additional constraints for line-image fitting. They allow to reduce the minimum number of points from three to two.

On the one hand, the normal direction to a line image is described by the gradient of the homogeneous expression (3.10) which is expressed by

$$\frac{\partial d}{\partial \hat{x}} = n_x - n_z \frac{\partial \hat{\alpha}}{\partial \hat{r}} \frac{\hat{x}}{\hat{r}} \tag{3.36}$$

$$\frac{\partial d}{\partial \hat{y}} = \pm n_y - n_z \frac{\partial \hat{\alpha}}{\partial \hat{r}} \frac{\hat{y}}{\hat{r}} \, . \tag{3.37}$$

where $\frac{\partial \hat{\alpha}}{\partial \hat{r}}$ is computed from the definition of $\hat{\alpha}$ (see Table 3.4).

On the other hand, given an intensity image $I(x,y)$, the gradient $\nabla I(x,y) = (\nabla I_x, \nabla I_y)$ of each edge lying on a line-image is aligned with the gradient of the line-image distance (3.10). The dot product between the analytical gradient (3.37) and the vector $\nabla I^{\perp} = (-\nabla I_y, \nabla I_x)^T$ describes the parallelism between both gradients. This expression can be written as

$$-\nabla I_y n_x \pm \nabla I_x n_y + n_z \frac{\partial \hat{\alpha}}{\partial \hat{r}} \frac{1}{\hat{r}} (\hat{x} \nabla I_y - \hat{y} \nabla I_x) = 0, \tag{3.38}$$

	$K_0 = \hat{\alpha}(\hat{r}_{vl})$	$K_1 = \frac{\partial \hat{\alpha}(\hat{r}_{vl})}{\partial \hat{r}}$	$K_2 = \frac{\partial^2 \hat{\alpha}(\hat{r}_{vl})}{\partial \hat{r}^2}\hat{r}_{vl}$
Para-Catadioptric	0	1	1
Hyper-Catadioptric	0	$\cos\chi$	$\cos^3\chi$
Equiangular-Fisheye	0	$\frac{\pi}{2}$	π
Stereographic-Fisheye	0	1	1
Orthogonal-Fisheye	0	∞	∞
Equisolid-Fisheye	0	2	6

Table 3.5: Derivatives of function $\hat{\alpha}(\hat{r})$ at $\hat{r}_{vl}$.

and gives an additional constraint when solving a line-image. In particular, knowing the system calibration and given the location and the gradient of a single point lying on a line-image , the linear system

$$\begin{pmatrix} \hat{x} & \pm\hat{y} & -\hat{\alpha} \\ -\nabla I_y & \pm\nabla I_x & \frac{\partial \hat{\alpha}}{\partial \hat{r}}\frac{1}{\hat{r}}(\hat{x}\nabla I_y - \hat{y}\nabla I_x) \end{pmatrix} \mathbf{n} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{3.39}$$

can be solved for $\mathbf{n}$ resulting

$$\mathbf{n} = \begin{pmatrix} \nabla I_x \hat{\alpha} + \frac{\partial \hat{\alpha}}{\partial \hat{r}}\frac{\hat{y}}{\hat{r}}(\hat{x}\nabla I_y - \hat{y}\nabla I_x) \\ \nabla I_y \hat{\alpha} - \frac{\partial \hat{\alpha}}{\partial \hat{r}}\frac{\hat{x}}{\hat{r}}(\hat{x}\nabla I_y - \hat{y}\nabla I_x) \\ \hat{y}\nabla I_y + \hat{x}\nabla I_x \end{pmatrix}. \tag{3.40}$$

3.4.3 Unified computation of main calibration parameter $\hat{r}_{vl}$

In previous sections, we have solved the plumb-line problem using the exact expression of $\hat{\alpha}$. In this section we explore the alternative of approximating $\hat{\alpha}(\hat{r})$ using a generic description depending on $\hat{r}$ and the main calibration parameter $\hat{r}_{vl}$. The goal of this approach is performing a direct method to exploit the gradient information in line-image extraction. In practice that means recovering the line-image and the main calibration parameter $\hat{r}_{vl}$ from two edge points reducing the number of iterations needed in a RANSAC approach. Important aspects to consider are the degree of the approximation and the approximating point (the point around which the neighbourhood of the function is approximated).

Linear approximation When approximating $\hat{\alpha}(\hat{r})$ by a linear function we consider the adequate approximating point is $\hat{r}_{vl}$. In this case ($\hat{r} = \hat{r}_{vl}$) implies $\hat{\alpha} = 0$ (see Section 3.3 and Fig. 3.4), and simplifying the expression it becomes

$$\hat{\alpha}(\hat{r}) = \frac{\partial \hat{\alpha}(\hat{r}_{vl})}{\partial \hat{r}} (\hat{r} - \hat{r}_{vl}). \tag{3.41}$$

This kind of linear approximation is similar to series description of distortion function used in [Tardif et al., 2006]. However, notice that in our approach we are approximating $\hat{\alpha}$ around $\hat{r} = \hat{r}_{vl}$ instead of approximating $\hat{\alpha}$ around $\hat{r} = 0$. When linearizing the function $\hat{\alpha}$ around $\hat{r} = 0$ the derivative of the function is equal to 0 (see Fig. 3.4) for all the systems meaning that the approximation is a constant $\hat{\alpha}(\hat{r}) = \hat{\alpha}(0)$ not depending on $\hat{r}$ which is a worse approximation [1].

In addition, linearising around $\hat{r}_{vl}$ allows computing $\hat{r}_{vl}$ even if the class of the central system is unknown. Using (3.41) as approximation of $\hat{\alpha}$ the system (3.39) becomes

$$\begin{pmatrix} \hat{x} & \pm\hat{y} & -\hat{r} & 1 \\ -\nabla I_y & \pm\nabla I_x & \frac{\hat{x}\nabla I_y - \hat{y}\nabla I_x}{\hat{r}} & 0 \end{pmatrix} \mathbf{m} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{3.42}$$

which can be solved from the coordinates and the gradient of two image points obtaining the vector $\mathbf{m} = \lambda \left(n_x, n_y, n_z \frac{\partial \hat{\alpha}(\hat{r}_{vl})}{\partial \hat{r}}, n_z \frac{\partial \hat{\alpha}(\hat{r}_{vl})}{\partial \hat{r}} \hat{r}_{vl} \right)^{\mathsf{T}}$. The main calibration parameter $\hat{r}_{vl}$ can be computed directly from $\mathbf{m}$ as $\hat{r}_{vl} = \frac{m_3}{m_4}$.

In this vector, n_z and $\frac{\partial \hat{\alpha}(\hat{r}_{vl})}{\partial \hat{r}}$ are coupled, therefore $\mathbf{n}$ can not be completely recovered. However if we know the system kind we can include the value of $\frac{\partial \hat{\alpha}(\hat{r}_{vl})}{\partial \hat{r}}$ in the equation system (see Table 3.5) and the equations system becomes

$$\begin{pmatrix} \hat{x} & \pm\hat{y} & -K_1\hat{r} & K_1 \\ -\nabla I_y & \pm\nabla I_x & K_1 \frac{\hat{x}\nabla I_y - \hat{y}\nabla I_x}{\hat{r}} & 0 \end{pmatrix} \mathbf{m} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{3.43}$$

where $\mathbf{m} = \lambda (n_x, n_y, n_z, n_z \hat{r}_{vl})^{\mathsf{T}}$. In this case both, $\hat{r}_{vl}$ and $\mathbf{n}$ are directly computed from $\mathbf{m}$. Notice that in this case we have 3 DOFs and 4 equations, therefore we still have margin for using a higher approximation.

Second order approximation If we consider the second order approximation in the neighbourhood of $\hat{r} = \hat{r}_{vl}$

$$\hat{\alpha}(\hat{r}) = \frac{\partial \hat{\alpha}(\hat{r}_{vl})}{\partial \hat{r}} (\hat{r} - \hat{r}_{vl}) + \frac{1}{2} \frac{\partial^2 \hat{\alpha}(\hat{r}_{vl})}{\partial \hat{r}^2} (\hat{r} - \hat{r}_{vl})^2 \tag{3.44}$$

which, using Table 3.5, can be expressed in terms of constants depending on the system kind

$$\hat{\alpha}(\hat{r}) = K_1 (\hat{r} - \hat{r}_{vl}) + \frac{1}{2} \frac{K_2}{\hat{r}_{vl}} (\hat{r} - \hat{r}_{vl})^2 \tag{3.45}$$

[1] In the case of an orthogonal system the derivative of the function at $\hat{r} = \hat{r}_{vl}$ is ∞ meaning that this is not the proper point for linearisation

therefore the system (3.39) becomes

$$\mathbf{m}^{\mathsf{T}}\begin{pmatrix} \hat{x} & -\nabla I_y \\ \pm\hat{y} & \pm\nabla I_x \\ -\hat{r}\,(K_1-K_2) & \frac{(\hat{x}\nabla I_y-\hat{y}\nabla I_x)(K_1-K_2)}{\hat{r}} \\ \left(K_1-\frac{K_2}{2}\right) & 0 \\ -\frac{K_2}{2}r^2 & K_2\,(\hat{x}\nabla I_y-\hat{y}\nabla I_x) \end{pmatrix} = \begin{pmatrix} 0 & 0 \end{pmatrix} \tag{3.46}$$

where $\mathbf{m} = \lambda\left(n_x\hat{r}_{vl}, n_y\hat{r}_{vl}, n_z\hat{r}_{vl}, n_z\hat{r}_{vl}^2, n_z\right)^{\mathsf{T}}$. Having 4 DOFs this description allows to construct a system of four equations from two points lying on the line-image.

This result is particularly useful for the case of equiangular fisheye system. As presented in Section 3.4.1, most of the equations of Table 3.3 have an analytical solution. However in the case of the equiangular fisheye system the constraint has not a closed solution and it is solved by minimization.

In addition, in this last proposal for the case of equiangular-fisheye the value of $K_1 - \frac{K_2}{2} = 0$ meaning that the equations system (3.46) simplifies becoming

$$\mathbf{m}^{\mathsf{T}}\begin{pmatrix} \hat{x} & -\nabla I_y \\ \pm\hat{y} & \pm\nabla I_x \\ \frac{\pi}{2}\hat{r} & -\frac{\pi}{2\hat{r}}\,(\hat{x}\nabla I_y-\hat{y}\nabla I_x) \\ -\frac{\pi}{2}r^2 & \pi\,(\hat{x}\nabla I_y-\hat{y}\nabla I_x) \end{pmatrix} = \begin{pmatrix} 0 & 0 \end{pmatrix} \tag{3.47}$$

where $\mathbf{m} = \lambda\left(n_x\hat{r}_{vl}, n_y\hat{r}_{vl}, n_z\hat{r}_{vl}, n_z\right)^{\mathsf{T}}$ and the computing of $\mathbf{n}$ and $\hat{r}_{vl}$ is direct.

Finally, we consider the case of the orthogonal fisheye. As the derivative in $\hat{r}_{vl}$ is ∞ it makes sense to use a second order approximation in the neighbourhood of $\hat{r} = 0$ obtaining the system

$$\begin{pmatrix} \hat{x} & \pm\hat{y} & 1 & -\frac{1}{2}r^2 \\ -\nabla I_y & \pm\nabla I_x & 0 & (\hat{x}\nabla I_y-\hat{y}\nabla I_x) \end{pmatrix}\mathbf{m} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{3.48}$$

where $\mathbf{m} = \lambda\left(n_x\hat{r}_{vl}, n_y\hat{r}_{vl}, n_z\hat{r}_{vl}^2, n_z\right)^{\mathsf{T}}$.

In Fig. 3.10 (b) we show an example of a line-image defined with two points and the corresponding gradient of the intensity field I at these points. In this example, the linear approximation is used.

In all these methods we are using as input the coordinates of two points and the gradient orientation at these points. As gradient information is usually noisy, we propose low-pass filtering the gradient orientation to increase the precision.

3.4.4 Computing the focal distance in hypercatadioptric systems

In this Section we expand a way for computing the focal length in hypercatadioptic systems. This allows us to compute both calibration parameters. Instead of using the

plumb-line constraint or a combination between plum-line and gradient constraints the normal $\mathbf{n}$ can be computed from a pair of points using the gradient constraint (3.38).

$$\begin{pmatrix} -\nabla I_{y1} & \pm\nabla I_{x1} & \frac{\partial\hat{\alpha}_1}{\partial\hat{r}}\frac{1}{\hat{r}_1}\left(\hat{x}_1\nabla I_{y1}-\hat{y}_1\nabla I_{x1}\right) \\ -\nabla I_{y2} & \pm\nabla I_{x2} & \frac{\partial\hat{\alpha}_2}{\partial\hat{r}}\frac{1}{\hat{r}_2}\left(\hat{x}_2\nabla I_{y2}-\hat{y}_2\nabla I_{x2}\right) \end{pmatrix}\mathbf{n}=\begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{3.49}$$

In practice, the solution is noisy and does not imply a real advantage with respect to the two point's location approach (3.11). The previous constraint (3.49) using 2 points and their gradients is enough to define a line-image. Therefore, when adding a third point with (3.38) in (3.49) one of the equations can be expressed in combination of the other two. In practice this means that,

$$\varpi_1\frac{\partial\hat{\alpha}_1}{\partial\hat{r}}\frac{1}{\hat{r}_1}+\varpi_2\frac{\partial\hat{\alpha}_2}{\partial\hat{r}}\frac{1}{\hat{r}_2}+\varpi_3\frac{\partial\hat{\alpha}_3}{\partial\hat{r}}\frac{1}{\hat{r}_3}=0 \tag{3.50}$$

where

$$\varpi_1=\left(\nabla I_{x2}\nabla I_{y3}-\nabla I_{x3}\nabla I_{y2}\right)\left(\hat{x}_1\nabla I_{y1}-\hat{y}_1\nabla I_{x1}\right) \tag{3.51}$$
$$\varpi_2=\left(\nabla I_{x3}\nabla I_{y1}-\nabla I_{x1}\nabla I_{y3}\right)\left(\hat{x}_2\nabla I_{y2}-\hat{y}_2\nabla I_{x2}\right) \tag{3.52}$$
$$\varpi_3=\left(\nabla I_{x1}\nabla I_{y2}-\nabla I_{x2}\nabla I_{y1}\right)\left(\hat{x}_3\nabla I_{y3}-\hat{y}_3\nabla I_{x3}\right) \tag{3.53}$$

This constraint is similar to (3.19) but using gradients (notice that location information is also used).

As noted before, gradient information is noisier than location information, therefore there is no advantage in using this constraint instead of (3.19). However, there is a case in which this constraint is useful. The constraint (3.19) is solved for each system in Table 3.3. Most of the devices taken into account have a single calibration parameter defining distortion. This is the case of equiangular, stereographic, orthogonal and equisolid. The parabolic case is defined by two parameters f and p but a coupled parameter $r_{vl}=fp$ can be used instead. In these cases the constraint involving three points can be used to estimate the calibration of the system. However, in the hyperbolic case the two parameters χ and f can not be coupled and only one of them can be estimated from this constraint.

By contrast, when simplifying the equation (3.50) for hypercatadioptric systems we found that mirror parameter χ is not involved:

$$\varpi_1\frac{1}{\sqrt{\hat{r}_1^2+f^2}}+\varpi_2\frac{1}{\sqrt{\hat{r}_2^2+f^2}}+\varpi_3\frac{1}{\sqrt{\hat{r}_3^2+f^2}}=0 \tag{3.54}$$

As a consequence, the focal distance f can be computed from the gradient orientation and the location of three points lying on a line-image.

Equation (3.54) can be expressed as a polynomial of degree 8 (but bi-quartic),

$$\sum_{m=0}^{4} \beta_m f^{2m} = 0 \tag{3.55}$$

where $\beta_m = \beta_{m,123} + \beta_{m,213} + \beta_{m,312}$ and

$$\beta_{0,ijk} = -\varpi_i^4 \hat{r}_j^4 \hat{r}_k^4 + 2\hat{r}_i^4 \hat{r}_j^2 \hat{r}_k^2 \varpi_j^2 \varpi_k^2 \tag{3.56}$$

$$\beta_{1,ijk} = 2\left(2\hat{r}_i^2 \hat{r}_j^2 \hat{r}_k^2 + \hat{r}_i^4 \left(\varpi_j^2 + \varpi_k^2\right)\right) \varpi_j^2 \varpi_k^2 - 2\varpi_i^4 \hat{r}_j^2 \hat{r}_k^2 \left(\hat{r}_j^2 + \hat{r}_k^2\right) \tag{3.57}$$

$$\beta_{2,ijk} = \hat{r}_j^2 \hat{r}_k^2 \left(4\varpi^2 \left(\varpi_j^2 + \varpi_k^2 - \varpi_i^2\right) + 2\varpi_j^2 \varpi_k^2\right) - \hat{r}_i^4 \left(\varpi_j^2 - \varpi_k^2\right)^2 \tag{3.58}$$

$$\beta_{3,ijk} = \hat{r}_i^2 \left(\varpi_i^2 \left(\varpi_j^2 + \varpi_k^2\right) + 2\varpi_j^2 \varpi_k^2 - \left(\varpi_j^4 + \varpi_k^4\right)\right) \tag{3.59}$$

$$\beta_{4,ijk} = -\varpi_i^4 + 2\varpi_j^2 \varpi_k^2 \tag{3.60}$$

This equation has 4 solutions (because the negative values for f have not sense).

3.5 Uncalibrated line-image extraction

In this section we present the algorithm for line-image extraction from uncalibrated omnidirectional cameras. The algorithm relies on the constraints presented in Section 3.4.1 based on a minimal set of 3 points. The algorithm is also developed for using gradient constraints (Section 3.4.3), based on a minimal set of 2 points. The proposed algorithm is described next (see Fig. 3.11).

The inputs of the method are the edges and their gradients obtained from a Canny detector (see Fig. 3.12 (a)). These edges are stored in connected components called boundaries and the gradient orientation is low-pass filtered to reduce noise. Then, a first splitting of the boundaries is done based on the variation of the gradient orientation. Each split is called a sub-boundary. The threshold of this splitting process has been chosen to minimize the number of splits. Therefore a sub-boundary can contain more than one segment. Not all the line-images contain relevant information about calibration (e.g. vertical lines). For this reason, a heuristic criterion is used to select a subset of these sub-boundaries containing relevant line-images. This heuristic technique is explained as follows. We define the descriptor $\kappa = s_{size} \cdot \Delta\theta$ which increases as the sub-boundary is better for computing $\hat{r}_{vl}$ where s_{size} is the size of the sub-boundary and $\Delta\theta$ is the angle covered by the sub-boundary rounding the principal point. Sub-boundaries are sorted by κ and then the ones which cover 50% of an accumulated histogram of κ are chosen (see Fig. 3.12 (b)).

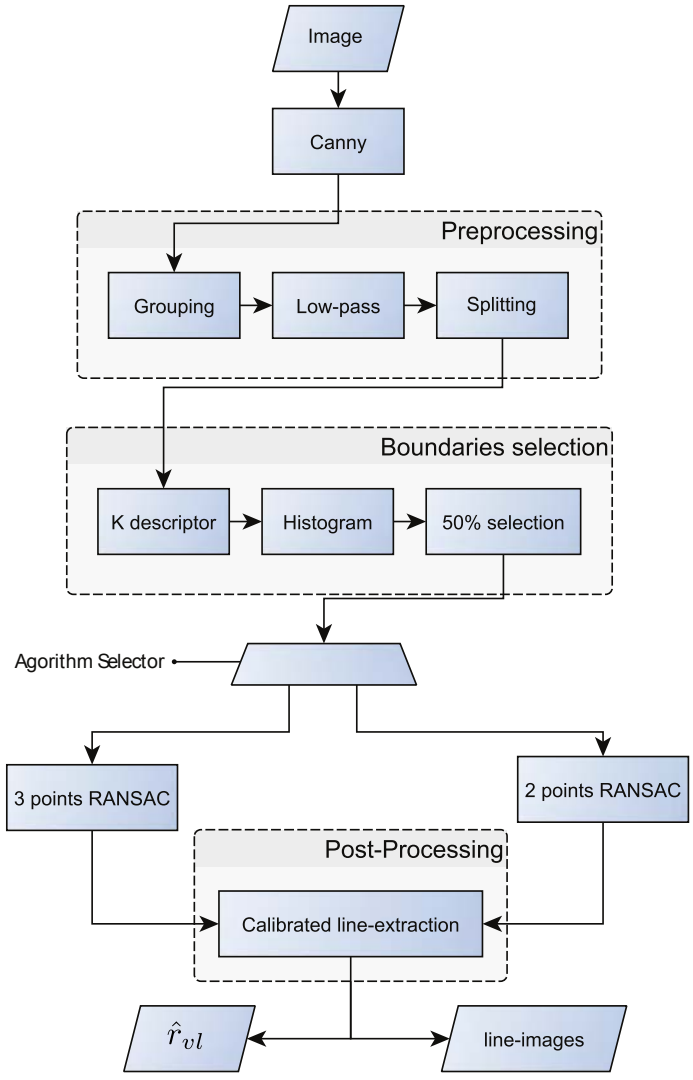
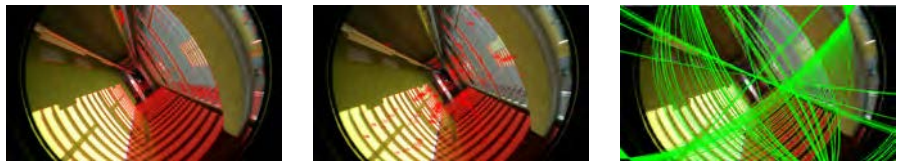


Figure 3.11: Uncalibrated line-image extraction procedure.



(a) (b) (c)

Figure 3.12: Steps in the line-image extraction: (a) Canny. (b) Preprocessing. (c) Line-Extraction

Algorithm 2 RANSAC approach for computing a set of line-images from a boundary of edge points.

for each boundary **do**
  while boundary contains line-images **do**
    for $iAttempt = 1$ to $nAttempts$ **do**
      $\hat{\mathbf{x}}_k = randomPoints\left(\hat{\mathbf{x}}_{boundary}\right)$;
      $[\hat{r}_{vl}, \mathbf{n}] = computeLineImage\left(\hat{\mathbf{x}}_k\right)$
      $dist = computeDistance\left(\mathbf{n}, \hat{\mathbf{x}}_{boundary}\right)$
      $\hat{\mathbf{x}}_{th} = computeThresholdPoint\left(\mathbf{n}, \hat{\mathbf{x}}_k, \delta_{px}\right)$
      $threshold = computeDistance\left(\mathbf{n}, \hat{\mathbf{x}}_{th}\right)$
      $votes\left(iAttempt\right) = dist < threshold$
    end for
    $lineImage.add\left(getMaxVotedLI\left(votes\right)\right)$
    $\hat{\mathbf{x}}_{boundary} = removeVotingPoints\left(\hat{\mathbf{x}}_{boundary}, votes\right)$
  end while
end for

Once we have selected boundaries containing line-images well conditioned for computing $\hat{r}_{vl}$ we use a RANSAC-based approach [Fischler and Bolles, 1981] (see Algorithm 2). Points of a boundary are selected randomly to generate candidate line-images which are voted by the other points of the boundary. Two approaches are provided depending on the constraint considered:

3-points $\hat{r}_{vl}$ estimation: $\hat{r}_{vl}$ is estimated from the location coordinates of a minimum of three points of the boundary and the line-image constraint presented in Section 3.4.1.

2-points $\hat{r}_{vl}$ estimation: From the location coordinates of a minimum of two points and theirs gradients we compute $\hat{r}_{vl}$ using the approximations presented in Section 3.4.3.

From any of these approaches, the normal vector $\mathbf{n}$ is computed from the defining points and $\hat{r}_{vl}$ using the expression (3.11). The distance to determine if a point is lying on the line-image is the algebraic distance (3.10) which is fast enough for the intensive evaluation needed in a RANSAC or in a Hough transform approach. In Fig. 3.8 we can see the different thickness for a given threshold depending on the different kind of systems. The region is not perfectly uniform along the curve but enough to decide which points belong to the line-image. Given a fixed threshold the thickness of this region also changes depending on the elevation of the projection plane $\mathbf{n}$. That means that a threshold being good enough for fitting some lines could not be good enough for others. To deal with this problem we compute a different threshold for each attempt of the RANSAC process (see Algorithm 2). Given a fixed threshold (δ_{px} in Algorithm 2) describing the thickness of the region in pixels, we compute the algebraic threshold corresponding to the line-image computed in the current attempt. As the algebraic threshold changes with each line-image we call it dynamic threshold (notice that δ_{px} remains constant). The process to estimate this dynamic threshold from δ_{px} and the vector $\mathbf{n}$ describing the line image is the following: Because of the nature of the RANSAC process we also have the location of the points defining the line-image of a given attempt ($\hat{\mathbf{x}}_k$ in Algorithm 2). As we have computed the line-image passing though these points we can compute the analytical gradient of the line-image on these points using equations (3.36-3.37) defining two straight lines passing though the points $\hat{\mathbf{x}}_k$ and orthogonal to the line-images (see Fig. 3.13). Along these straight lines we can define a point from a given distance δ_{px} of the line-image. On these points we compute the algebraic distance (3.10) and we consider its mean the threshold for the testing points.

Each point of the boundary whose distance is minor than the dynamic threshold gives a vote for the candidate. The most voted candidate is selected as best fit. From the previous procedure we obtain for each line-image: the points lying on it, the corresponding normal vector, and an estimation of the calibration via the $\hat{r}_{vl}$ value. From the collection of estimations for $\hat{r}_{vl}$ we compute a single value using the median. Thanks to the pre-selection based on κ most of the line-images return similar values of $\hat{r}_{vl}$ getting a small number of spurious. In Fig. 3.14 we show a box-plot representing the disper-

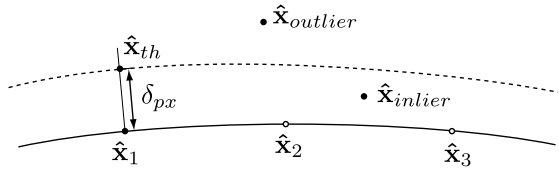


Figure 3.13: Computing the dynamic threshold from the defining points of the line-image.

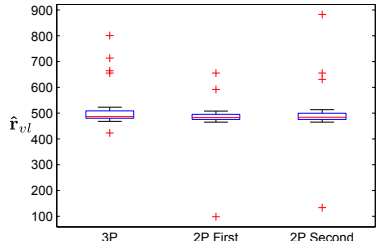


Figure 3.14: Dispersion in values of $\hat{r}_{vl}$ for the different extracted line-images. 3P for 3-points approach, 2P First for 2-points linear approximation approach and 2P Second for 2-points second order approximation approach.

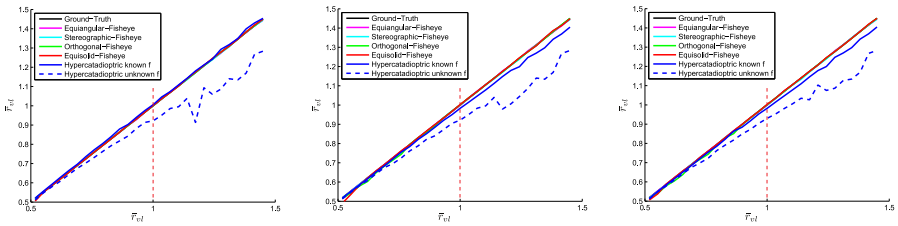
sion in the values of $\hat{r}_{vl}$ from the line-images extracted in Fig. 3.12. The estimation for the value of $\hat{r}_{vl}$ is similar in the three approaches. Finally, main calibration parameter and normals are refined in a non-linear optimization process and then, the line-images are re-extracted using this value.

3.6 Experiments

We present different experiments to validate the proposal and to compare the two approaches presented in section 3.5. The accuracy in line extraction and calibration is measured using synthetic images with known ground-truth. We also evaluate the influence of the error in the principal point and compare our results with a calibration method. Finally, we present experiments to show the behaviour of the method with real images.

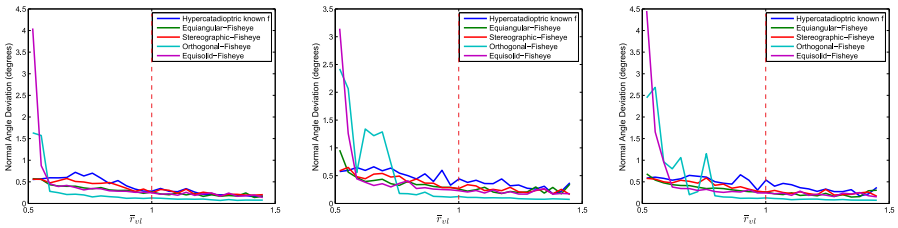
3.6.1 Experiments with synthetic data

To evaluate the accuracy of the algorithm we have used synthetic images generated for the different catadioptric and dioptric models considered in Section 3.2. A useful measure for comparing calibration estimations among the different systems is the ra-



(a) (b) (c)

Figure 3.15: Estimation of $\hat{r}_{vl}$ from uncalibrated line extraction. (a) 3-points approach, (b) 2-points linear approximation approach, (c) 2-points second order approximation approach. In all, the horizontal axis corresponds to the actual $\hat{r}_{vl}$ and the vertical axis to the estimated one.



(a) (b) (c)

Figure 3.16: Angle deviation between obtained normals and ground truth. (a) 3-points approach, (b) 2-points linear approximation approach, (c) 2-points second order approximation approach.

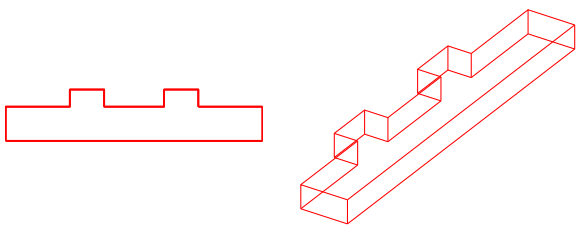
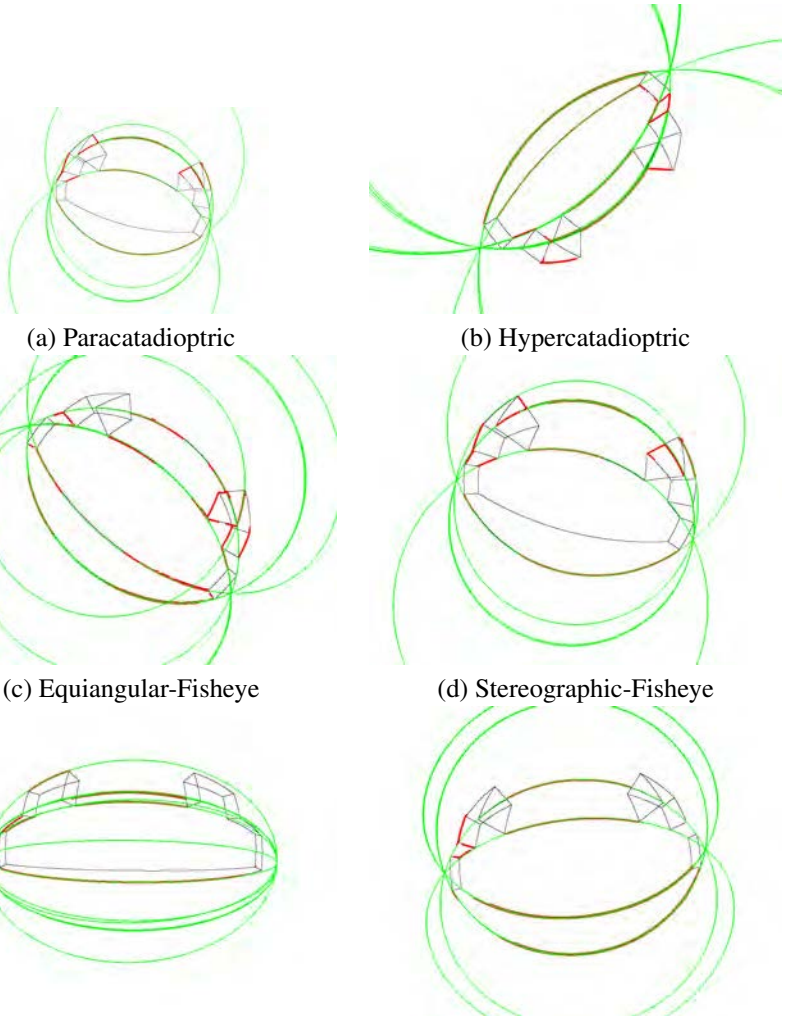


Figure 3.17: (a) Top-view of the simulated scenario. (b) Isometric view of the simulated scenario.



(a) Paracatadioptric (b) Hypercatadioptric

(c) Equiangular-Fisheye (d) Stereographic-Fisheye

(e) Orthogonal-Fisheye (f) Equisolid-Fisheye

Figure 3.18: Extraction example on synthetic images simulating different classes of central systems.

dius of the vanishing line $\hat{r}_{vl}$. This radius depends only on the calibration and it is a magnitude common to any central system.

First experiment evaluates the deviation of the estimated radius of the vanishing line $\hat{r}_{vl}$. The setup of the experiment is the following: Given an scene composed by segments in a corridor (see Fig. 3.17) we want to characterize the error of estimation in $\hat{r}_{vl}$. For each value of $\hat{r}_{vl}$ and for each system we generate 10 different images with a resolution of 1024×1024 pixels. A randomized perturbation in the orientation and location of the camera is introduced in each capture. From each image we obtain a value of $\hat{r}_{vl}$ and a set of line projections described by the normal vector $\mathbf{n}$. The experiment is repeated for each proposed method.

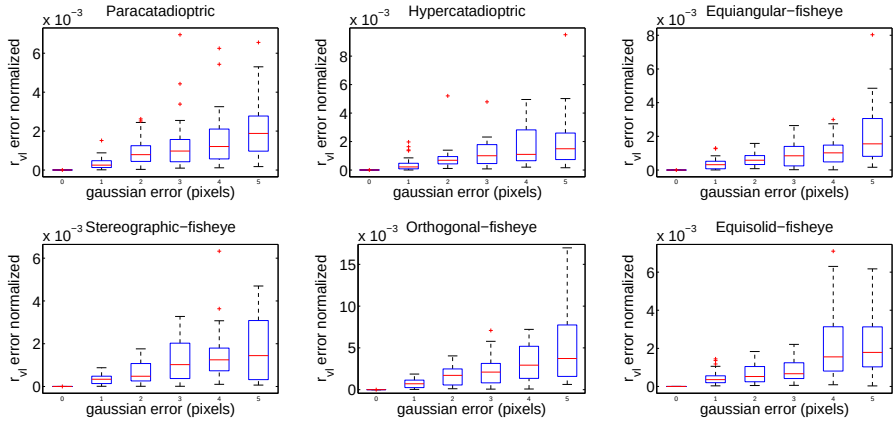
In Fig. 3.15 we show the deviation of the estimated radius of the vanishing line $\hat{r}_{vl}$, which has been normalized to make it independent from the image size. For each value of $\hat{r}_{vl}$ we have computed 10 estimations of $\hat{r}_{vl}$ using 10 different images; we represent the mean of these estimations. The normalized value $\bar{r}_{vl} = 1$ corresponds to the case of vanishing line projection passing through the corner of the image (e.g. with a resolution of 1024×1024, values in pixels of $\bar{r}_{vl} = \frac{\hat{r}_{vl}}{724}$). When $\bar{r}_{vl} < 1$ the FOV of the system is greater than 180 degrees. Otherwise the FOV of the system is smaller than 180 degrees.

Fig. 3.15 (a) depicts the results using the 3-points-$\hat{r}_{vl}$ estimation. Fig. 3.15 (b-c) depicts the results using the 2-points-$\hat{r}_{vl}$ estimation when using first and second approximation. For the case of the hypercatadioptric system we show two curves. The continuous curve corresponds to the estimation of $\hat{r}_{vl}$ assuming that parameter f is known. The dashed line corresponds to the the estimation of $\hat{r}_{vl}$ when none of the parameters are known.

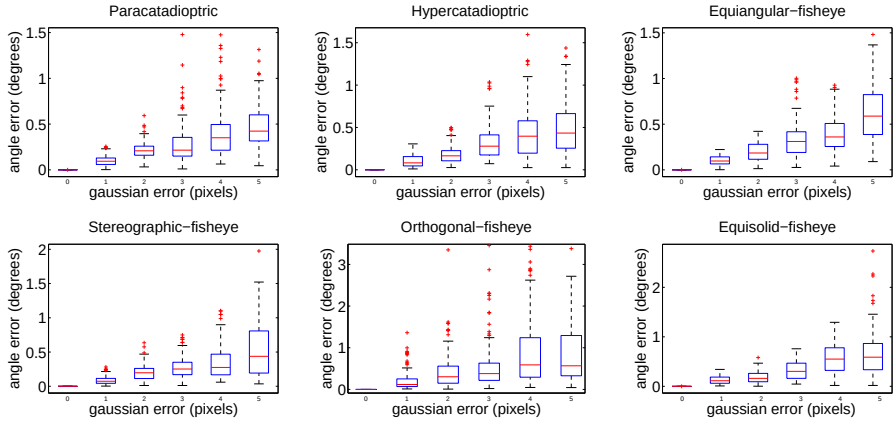
Another measure of the accuracy in estimation is the deviation in the orientation of the normals $\mathbf{n}$ describing the line projections. In Fig. 3.16 we show the error in degrees between the ground truth normals and the estimated ones depending on $\bar{r}_{vl}$. From each image the mean of the deviations of all the line-projections is used. As we have 10 samples for each value of $\bar{r}_{vl}$ we also compute the median of these values.

We observe that, in general, accuracy in $\hat{r}_{vl}$ estimation decreases when the radius is greater than the image size (values of $\bar{r}_{vl}$ greater than one). That means that the smaller the FOV of the system the lesser the accuracy of the algorithm. The extreme case is the perspective camera where $\hat{r}_{vl} = \infty$ and curvature of lines is independent from the focal distance. We also note that the hypercatadioptric case is more difficult to solve than others because of the multiple parameters involved. The accuracy decreases considerably in systems with FOV less than 180 degrees. However if one of the calibration parameters is known the behaviour is similar to the other systems. In Fig. 3.18 we show some examples of the simulated images for different catadioptric and dioptric devices and the corresponding extracted line-images.

Influence of principal point accuracy In the previous experiment we have assumed that the principal point is known. To evaluate the robustness of the algorithm we eval-

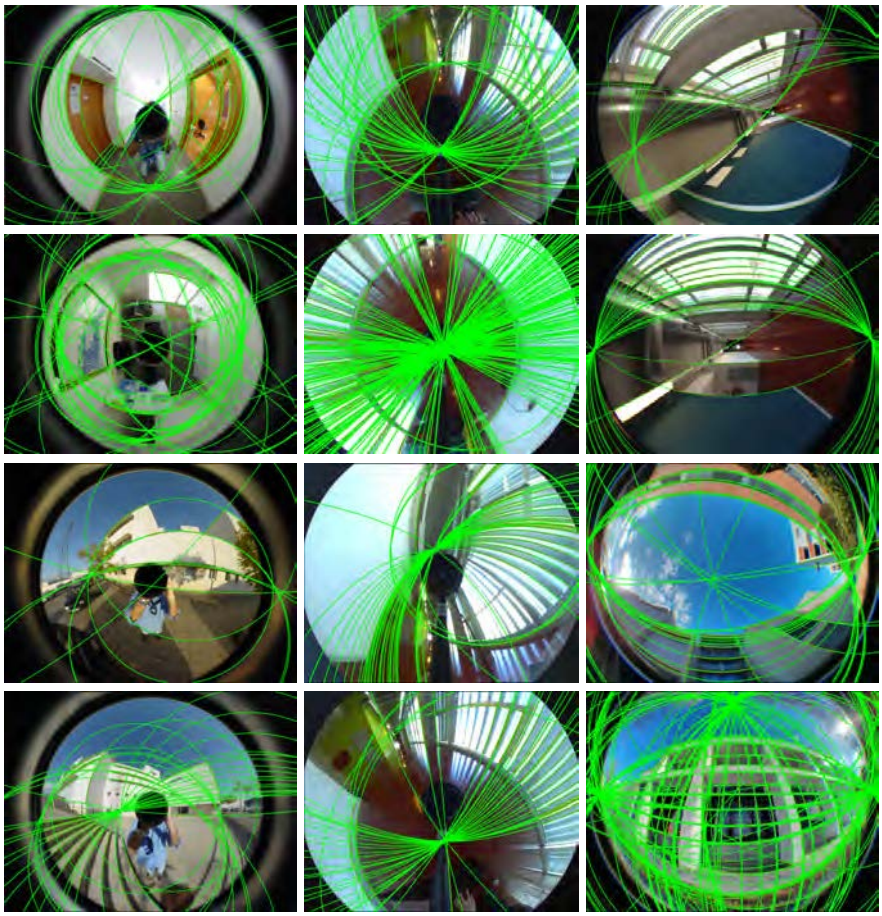


(a) $\hat{r}_{vl}$ estimation error.



(b) angle estimation error.

Figure 3.19: Influence of principal point: (a) $\hat{r}_{vl}$ estimation error. (b) angle estimation error.



(a) (b) (c)

Figure 3.20: Line extraction on uncalibrated omni-images (3-points algorithm). (a) Paracatadioptric. (b) Hypercatadioptric. (c) Equiangular-Fisheye

uate the influence of Gaussian error in the principal point. The ground-truth of calibration $\hat{r}_{vl}$ is fixed to 750 pixels and the image size is 1024×768 pixel (FOV > 180 deg). In Fig. 3.19 we represent the error when estimating the main calibration parameter $\hat{r}_{vl}$ and the error in degrees between the ground-truth normals describing each line projection and the estimated ones. For the case of the hypercatadioptric system we estimate $\hat{r}_{vl}$ and the normals $\mathbf{n}$ assuming that parameter f is known. We can observe that the median of the error in the deviation of the normal does not exceed 0.5 degrees.

Calibration method comparison The objective of our proposal is the extraction of line projections in central system when the calibration is unknown. In addition we compute the radius of the vanishing line $\hat{r}_{vl}$ which is considered the main calibration parameter of the system. To illustrate the behaviour of our proposal we have carried out a "proof of concept" comparison between our approach and the method proposed in [Barreto and Araujo, 2005]. We use the same imagery as input as in both methods.

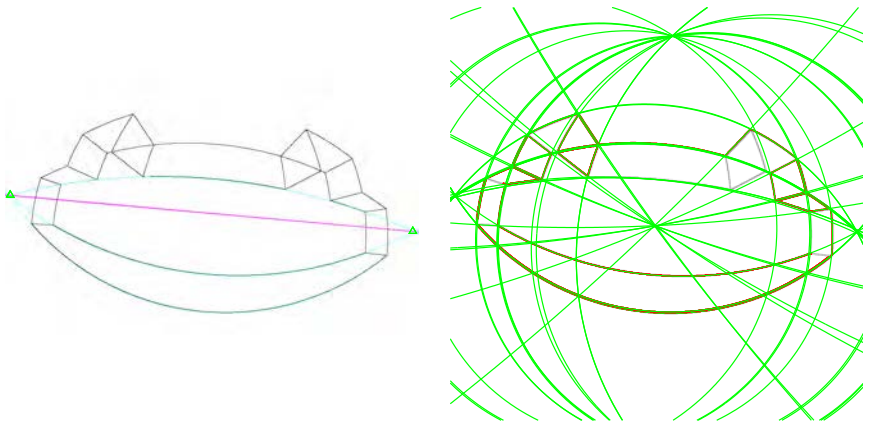
Although the results from both methods can be directly compared due to the similarity of the projection models, there are some difference we need to take into account:

- The proposal of [Barreto and Araujo, 2005] is only valid for catadioptric systems, hence we cannot compare the approaches proposed for fisheye systems.
- In [Barreto and Araujo, 2005] the full calibration is performed in a closed form. Our proposal does not compute the principal point.
- Although a self-calibration is possible with [Barreto and Araujo, 2005] approach, in practice it is a hard problem with 5 degrees of freedom hence in the toolbox the user has to manually select the contours containing the line-images. By contrast our toolbox is automatic and the supervision of the user consists of the parameters tuning.

The input image is a synthetic image of a paracatadioptric system used in the previous experiments with a ground truth of $\hat{r}_{vl} = 500$ *pixels*. The method presented in [Barreto and Araujo, 2005] gives a result of $\hat{r}_{vl} = 485.94$ *pixels*. Our method, returns a value of $\hat{r}_{vl} = 500.36$ *pixels*. Notice that, while Barreto's method is computing the full calibration, we assume the principal point is known having not skewness. In Fig. 3.21 (a) we show the output image of the method presented in [Barreto and Araujo, 2005] containing the line-images used for calibrating and in Fig. 3.21 (b) the output of our proposal showing the extracted line-images. We can see that the parametric representation of the line-images in both images is similar.

3.6.2 Experiments with real images

In order to show how the method works with real images several tests have been performed with different catadioptric and fisheye cameras. The principal point has been coarsely estimated. The 3-points algorithm have been used for all these examples. In



(a) (b)

Figure 3.21: Output of the toolboxes showing the line-images: (a) Method of Barreto & Araujo. (b) Our proposal.

Fig. 3.20 (a) we show the behaviour of the proposal using paracatadioptric images from the CatPack toolbox [Barreto and Araujo, 2006b]. In Fig. 3.20 (b) we show our results in hypercatadioptric images with resolution of 1280×1024 pixels and has been acquired with a hypercatadioptric system composed of an IDS uEye camera and an hyperbolic mirror made by Neovision. In Fig. 3.20 (c) a fisheye based camera is used. The images has been taken with an iPhone 4S camera (3264×2498 pixels) and a Nexus 4 camera (3264×2448 pixels) with an equiangular fisheye for cell phones made by Pixeet. Notice that line-images of parallel 3D lines intersect in the vanishing points of the image because line-images have been well extracted.

In second experiment we have applied the algorithm to an image sequence taken with a camera in hand. The objective of this experiment is evaluating the robustness of the proposal when supervision is not available. The sequence has been acquired with a Nexus 4 camera using an equiangular fisheye with a resolution of 1920×1080 pixels [2]. Each frame is independently computed to emphasize the robustness of the proposal without supervision. The processing has been off-line computed using the 3-points approach. In Fig. 3.22(b) we show the estimation of the main calibration parameter $\hat{r}_{vl}$ along the sequence. The mean estimation for $\hat{r}_{vl}$ is 568.41 pixels and the standard deviation is 9 pixels, meaning that the proposal is robust enough to be applied in a sequence of images (an implementation in real-time must consider aspects not addressed in this chapter). As this scenario is composed of parallel lines, a qualitative measure of the quality of the extracted line-images is the location of the vanishing points on these images. Line-images from video (see Fig. 3.22(a)) intersect at the

[2] webdiis.unizar.es/%7Ebermudez/suppMat.html

vanishing points of the image (notice that we do not have imposed this constraint).

The presented approach can be tested with a toolbox for Matlab [3] available on line. This toolbox allows to obtain the calibration of a fisheye or a catadioptric system just from a single picture. The setup is very simple because neither specific pattern nor special movements of the camera are needed.

3.7 Summary

We have presented a method to extract line-images from uncalibrated central images with revolution symmetry. We consider a general framework which encodes projection models for dioptric and catadioptric system using a common main calibration parameter (the radius of the vanishing line $\hat{r}_{vl}$). We characterize the line projections for different types of catadioptric and dioptric systems evaluating common properties. Due to the relation between projection models and curvature of line-images an estimation of the main calibration parameter of the system is obtained simultaneously. We observed that line-images are correctly extracted and the obtained normals are accurate enough to be used in 3D computations. Nevertheless, in hypercatadioptric systems the accuracy is lower because two calibration parameters are involved. However, the accuracy is similar to the other systems if one of the parameters is known or when the FOV of the system is greater than 180 degrees. Besides characterizing the proposal it has been tested with real images from catadioptric and dioptric systems. Tests with real and synthetic images can be replicated with an implementation of the method provided as open source for research purposes.

[3] webdiis.unizar.es/%7Ebermudez/toolbox.html

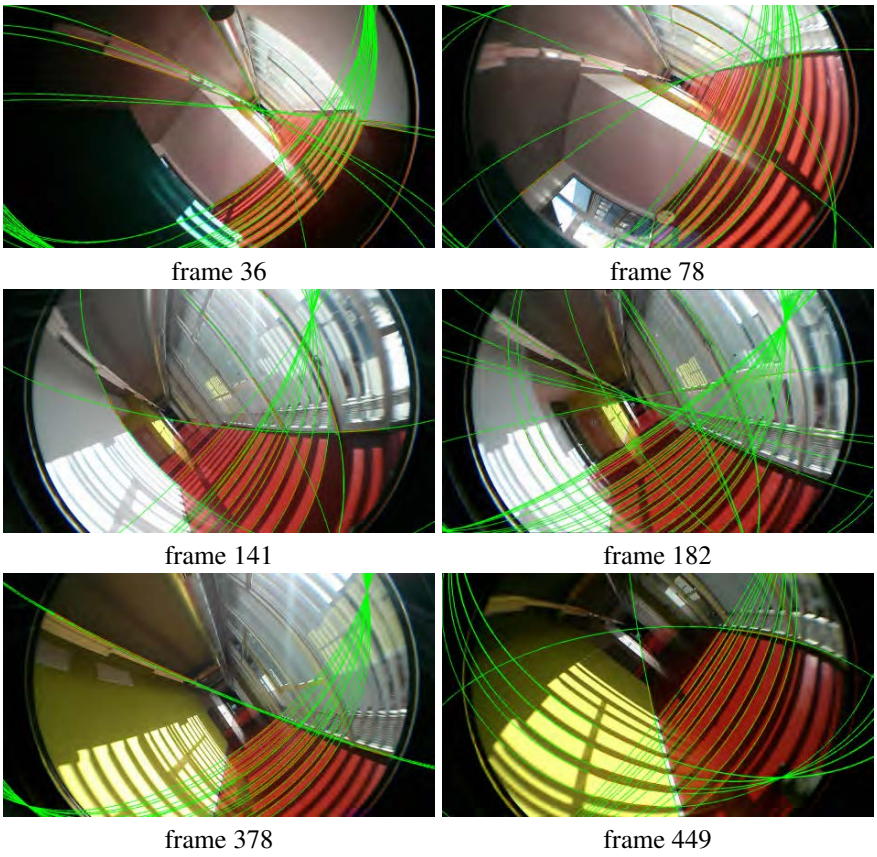


Figure 3.22: Line extraction on uncalibrated sequence: (a) Extracted line-images in sequence (b) $\hat{r}_{vl}$ estimation in blue $\hat{r}_{vl}$ mean value in red

Part II

Line-images in non-central omnidirectional cameras

Chapter 4

Line-image fitting in non-central catadioptric systems

Second part of this thesis is dedicated to line projections in non-central systems. By contrast with central systems, where the projection surface of a line is a plane, in most non-central systems the projection surface of a line is a ruled surface having at least four degrees of freedom. The interest of this property is that the resulting line-image is a curve which contains the 4 degrees of freedom of the 3D line allowing to recover the original 3D line from a single projection. However, the extraction of this curve is also qualitatively harder. In this chapter we investigate the analytic representation of line-images in different non-central catadioptric systems. This representation is used in a method for automatic line-image extraction from non-central catadioptric images. For performing this method we have analytically solved the metric distance from a point to line-image for different kinds of non-central catadioptric systems. We also propose a measure we call effective baseline measuring the quality of the reconstruction of a 3D line from 4 rays. This measure allows pre-evaluating the random attempts of a robust scheme which is a key issue for line-image extraction. This allows reducing the number of trials in the whole process. The proposal is evaluated in intensive simulations and tested with synthetic and real images. Up to our knowledge, this is the first work successfully addressing line-image extraction in non-central systems.

4.1 Introduction

In central systems the projection of a 3D line is a plane passing through the viewpoint of the camera. In this class of projection some of the information of the 3D line is lost because any line lying on this plane is projected on the same line-image. In other

words, a 3D line occludes any other line located behind because the projection surface is a plane.

By contrast, in non-central systems the projecting rays do not intersect a common viewpoint. The locus of the viewpoint is, in general, tangent to a caustic [Agrawal et al., 2010] which is an envelope surface of the projecting rays. When the system has revolution symmetry, the projecting rays coming from a 3D line are skew 3D lines forming a ruled surface which lies on the axis of symmetry. In [Teller and Hohmeyer, 1999, Kneebone et al., 1998, Griffiths and Harris, 2011] it is proven that four generic lines induce two incident lines [1]. Therefore, when we have a surface defined by at least four skew and non cohyperbolic rays only two lines intersecting all the rays lie on this surface. The previous result implies that the projection surface contains the axis of revolution, the given line and any line intersecting both the axis and the line. No additional line can belong to this surface, hence in axial non-central projections, it is not possible to occlude a line with other line.

As a consequence, the complete geometry of a 3D line is mapped on a single non-central image and it can be completely recovered from at least 4 line-image points or projecting rays. On the contrary, points do not provide 3D information since points always occlude other points.

This particularity implies a geometric advantage of lines with respect to points in non-central cameras and justifies the use of line features. The richer information about the localization of the line in the space facilitates the tracking of the features even if the photometric information is not very discriminative. Notice that, from the photometric point of view, points are more discriminative than line projections, however line features usually represent boundaries of the scene that remain even in texture-less scenarios.

Some previous approaches have tried to fit 3D lines from a single image in non-central catadioptric systems. The approach presented by [Teller and Hohmeyer, 1999] exploits the intersection operator between lines for defining a linear system from 4 rays computing the two incident lines. Since in non-central systems with symmetry of revolution one of these lines is always the axis of revolution, this approach is used in [Caglioti and Gasparini, 2005, Caglioti et al., 2007a, Gasparini and Caglioti, 2011] for estimating 3D lines from line projections of non-central catadioptric images and studying the degeneracies and singular configurations. In [Lanman et al., 2006] the same approach is used with spherical catadioptric mirrors, and two additional non-central systems for reconstruction. Work in [Swaminathan et al., 2008] extends the approach from lines to planar curves. Some simplifications have been used to improve the reconstructions by reducing the DOFs of the problem: considering horizontal lines [Pinciroli et al., 2005, Chen et al., 2011], exploiting cross-ratio properties [Perdigoto and Araujo, 2012] or imposing additional constraints such as parallelism or perpendicularity [Bermudez-Cameo et al., 2014a].

[1] Four lines are generic if no two of them are coplanar, no three of them are coconical or cocylindrical, and the four are not cohyperbolic, i.e. do not lie on the same ruled quadric surface.

Line projections have been also used to estimate the calibration of non-central systems in a generalization of the plumb-line approach. For example in [Caglioti et al., 2007b] they exploit the fact that there exists less ambiguity when the system is off-axis with impressive results. However, that approach does not allow them to obtain an analytical expression of the line projection. In [Agrawal and Ramalingam, 2013] they exploit particular geometric properties of spherical mirrors for computing extrinsic calibration parameters. As application, the pose of non-central catadioptric systems is estimated in an image sequence [Miraldo and Araujo, 2014, Miraldo et al., 2015] using known 3D lines.

Automatically extracting the projection of a line on non-central images is a challenging task. In omnidirectional central systems the problem has been recently solved for calibrated [Bazin et al., 2010, Ying and Zha, 2005] and uncalibrated images [Bermudez-Cameo et al., 2015c, Tardif et al., 2006]. In non-central images difficulty increases due to the high distortions induced on line-images, the elevated number of degrees of freedom involved in the extraction and the lack of effective baseline of current non-central systems.

In this Chapter we present a method for automatic extraction of line projections or line-images in non-central catadioptric systems with symmetry of revolution. Up to our knowledge, this is the first work addressing this problem. The proposal has been considered for systems with symmetry of revolution and it has been particularized for conical catadioptric and spherical catadioptric systems. This procedure automatically segments the collection of edges corresponding to line-images. The complete 3D localization of a line is also recovered from the extraction in a single image even if the accuracy of the result is high sensitive to noise. The contributions of this work are the following:

- A unified framework for describing line-images in non-central systems with revolution symmetry.
- Polynomial expressions of the line-images for conical and spherical catadioptric systems.
- A closed form solution for computing the geometry of the mirror from 5 points lying on a line-image in conical catadioptric systems.
- Solutions based on polynomial roots for computing the Euclidean distance from point to line-image for conical and spherical catadioptric systems.
- A feature for measuring effective baseline in a set of rays in non-central systems.
- A proposal for automatic line-image extraction in non-central images.
- A comparison between catadioptric systems of similar sizes in terms of accuracy.

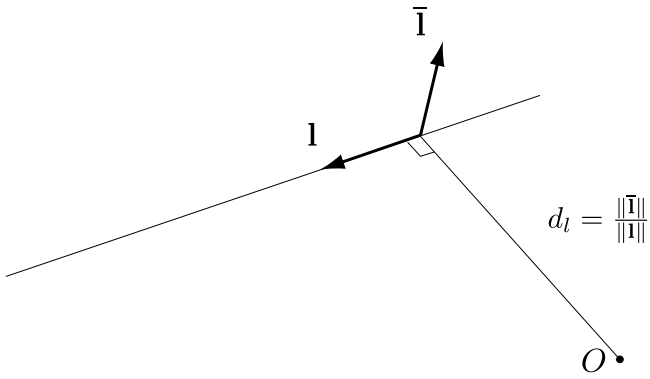


Figure 4.1: Euclidean interpretation of Plücker coordinates.

In Section 4.2 we introduce the necessary background to understand the proposal. In Section 4.3 we present a unified framework for describing rays and line projections in non-central systems with symmetry of revolution. In Section 4.4 we analytically solve the metric distance from a point to the line-image. In Section 4.5 we present the proposed description of the line-image for conical catadioptric systems. We particularize the corresponding metric distance previously presented in Section 4.4 and we introduce the computing of the mirror geometry parameter from 5 points of the line-image. In Section 4.6 we show the polynomial description of the line-image for spherical catadioptric systems and we particularize the corresponding metric distance for this case. In Section 4.7 we present the algorithm for line-image extraction in non-central images including the proposed feature for measuring the effective baseline of a set of rays. In Section 4.8 we evaluate the method with synthetic and real images. Finally in Section 4.9 we present the discussions.

4.2 Background

In this section we introduce the relevant geometric concepts and notation used in this Chapter. In particular, we summarize the description used for lines which is based on Grassmann-Cayley algebra (see Appendix A for more details), the transformations between systems of references and the side operator between two lines.

4.2.1 Plücker coordinates

The Plücker coordinates of a 3D line is an homogeneous representation of a line $\mathbf{L} \in \mathbb{P}^5$ defined by the null space of two $\mathbb{P}^3$ points of the line. When correctly arranged, this representation can be decomposed in two $\mathbb{R}^3$ vectors $\mathbf{L} = \left(\mathbf{l}^\mathsf{T}, \bar{\mathbf{l}}^\mathsf{T}\right)^\mathsf{T}$ with geometrical

meaning in Euclidean geometry. $\mathbf{l} \in \mathbb{R}^3$ is called the direction vector and represents the direction of the line. $\bar{\mathbf{l}} \in \mathbb{R}^3$ is called the moment vector and represents the normal to a plane passing through the 3D line and the origin of the reference system O. Not all elements of $\mathbb{P}^5$ correspond to 3D lines. Any point of $\mathbb{P}^5$ corresponding to a line in $\mathbb{P}^5$ must satisfy $\mathbf{l}^\mathsf{T}\,\bar{\mathbf{l}} = 0$ which is known as Plücker identity. The Euclidean interpretation of this identity is the orthogonality between the direction vector $\mathbf{l}$ and the projection plane vector $\bar{\mathbf{l}}$ (see Fig. 4.1). The minimum distance from the origin O to the 3D line is computed as $d_l = \frac{\|\bar{\mathbf{l}}\|}{\|\mathbf{l}\|}$.

4.2.2 Change of reference of Plücker coordinates

Consider $\mathbf{X} \in \mathbb{P}^3$ the representation of a 3D point in homogeneous coordinates in an Euclidean reference system and $\mathbf{X}' \in \mathbb{P}^3$ the representation in another Euclidean reference system; the transformation describing the Euclidean transformation between both reference systems is represented by the matrix

$$\mathsf{T} = \begin{pmatrix} \mathsf{R} & \mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix} \text{ such that } \mathbf{X} = \mathsf{T}\mathbf{X}' \text{ and } \mathbf{X}' = \mathsf{T}^{-1}\mathbf{X}, \tag{4.1}$$

where $\mathsf{R} \in SO^3$ is a rotation matrix and $\mathbf{t} \in \mathbb{E}^3$ a translation vector. The corresponding transformation for changing the reference of a line $\mathbf{L} \in \mathbb{P}^5$ expressed in Plücker coordinates is

$$\mathsf{G} = \begin{pmatrix} \mathsf{R} & \mathbf{0} \\ [\mathbf{t}]_\times \mathsf{R} & \mathsf{R} \end{pmatrix} \text{ such that } \mathbf{L} = \mathsf{G}\mathbf{L}' \text{ and } \mathbf{L}' = \mathsf{G}^{-1}\mathbf{L}, \tag{4.2}$$

where $[\mathbf{t}]_\times = \begin{pmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{pmatrix}$ is the antisymmetric operator.

4.2.3 The side operator

Given two 3D lines expressed in Plücker coordinates $\mathbf{L}_i = \left(\mathbf{l}_i^\mathsf{T}, \bar{\mathbf{l}}_i^\mathsf{T}\right)^\mathsf{T}$ and $\mathbf{L}_j = \left(\mathbf{l}_j^\mathsf{T}, \bar{\mathbf{l}}_j^\mathsf{T}\right)^\mathsf{T}$, the side operator between them is defined as

$$\mathsf{side}\,(\mathbf{L}_i, \mathbf{L}_j) = \mathbf{l}_i^\mathsf{T}\bar{\mathbf{l}}_j + \mathbf{l}_j^\mathsf{T}\bar{\mathbf{l}}_i. \tag{4.3}$$

The operator $\mathsf{side}\,(\mathbf{L}_i, \mathbf{L}_j)$ is a signed distance whose sign defines the side of the line $\mathbf{L}_j$ with respect the line $\mathbf{L}_i$, such that

$$|\mathsf{side}\,(\mathbf{L}_i, \mathbf{L}_j)| = d_E \|\mathbf{l}_i\| \|\mathbf{l}_j\| |\sin\alpha| \tag{4.4}$$

where d_E is the metric distance between the closest point of both lines and α the angle between $\mathbf{l}_i$ and $\mathbf{l}_j$ (see Fig. 4.2).

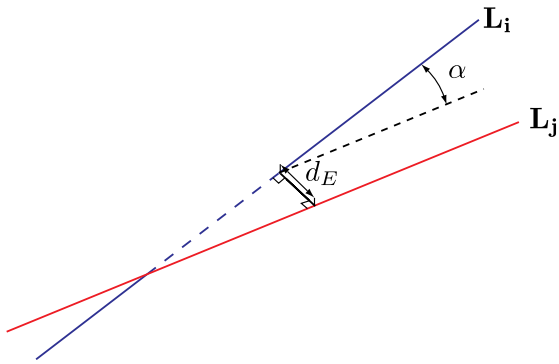


Figure 4.2: Distance d_E between two skew rays.

4.3 Line projection in non-central systems

In this section we present a unified framework for describing rays and line projections in non-central systems with symmetry of revolution.

4.3.1 Projection model in non-central systems with revolution symmetry

To exploit the revolution symmetry property we assume that image points $\mathbf{x} = (x, y, z)^{\mathsf{T}} \in \mathbb{P}^2$ are homogeneous coordinates expressed in a reference centred in the projection of the axis of revolution. This coordinate system is normalized with the camera parameters. The relation between these coordinates and final images coordinates (u, v) is an affine transformation K_c.

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \underbrace{\begin{pmatrix} f_x & s & u_0 \\ 0 & \pm f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix}}_{\mathsf{K}_c} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \tag{4.5}$$

The back projection model. In a general non-central system each ray is defined by two points [Sturm et al., 2011], e.g. the 3D point and a point in which the ray is tangent to a surface called caustic. When the system has symmetry of revolution any ray can be expressed in terms of three parameters: elevation angle ϕ, azimuth angle θ and distance to the intersection between the ray and the vertical axis Z_r (see Figure 4.3). The representation of this ray in Plücker coordinates is

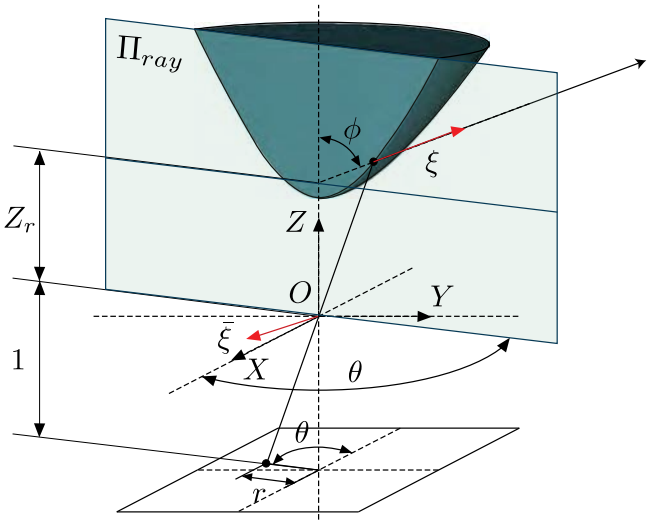


Figure 4.3: Back projection of a point $\mathbf{X}$ in non-central catadioptric axial system.

$$\Xi = \begin{pmatrix} \xi \\ \bar{\xi} \end{pmatrix} = \begin{pmatrix} \sin\phi\cos\theta \\ \sin\phi\sin\theta \\ \cos\phi \\ -Z_r\sin\phi\sin\theta \\ Z_r\sin\phi\cos\theta \\ 0 \end{pmatrix} \tag{4.6}$$

with $\xi = (\xi_1, \xi_2, \xi_3)^\mathsf{T}$ and $\bar{\xi} = \left(\bar{\xi}_1, \bar{\xi}_2, \bar{\xi}_3\right)^\mathsf{T}$.
Notice that, when taking polar coordinates ($r = \sqrt{x^2+y^2}$ and $\theta = \mathsf{atan2}\,(y,x)$) in this reference, θ corresponds to the azimuth angle shown in the ray description. The elevation angle ϕ and Z_r depend on image radius r and the system calibration.

4.3.2 Generic line projection model for non-central systems with revolution symmetry

Consider an image point $\mathbf{x}$, its corresponding projecting ray $\Xi = \left(\xi^\mathsf{T}, \bar{\xi}^\mathsf{T}\right)^\mathsf{T}$ intersects a 3D line $\mathbf{L} = \left(\mathbf{l}^\mathsf{T}, \bar{\mathbf{l}}^\mathsf{T}\right)^\mathsf{T}$ when

$$\mathsf{side}\,(\mathbf{L}, \Xi) = \xi^\mathsf{T}\bar{\mathbf{l}} + \bar{\xi}^\mathsf{T}\mathbf{l} = 0\,. \tag{4.7}$$

Having at least four points and their corresponding rays, the intersection of the projecting rays $\Xi_{1..4}$ with the line $\mathbf{L}$ is described by a linear system. In [Teller and Hohmeyer, 1999] and [Gasparini and Caglioti, 2011] the solution of this system of equations is used to compute the Plücker representation of the 3D line. Since Plücker coordinates is an over-parametrized representation the null space of the solution has one dimension. However, not all six-elements vector corresponds to a Plücker line. By imposing the Plücker line constraint ($\mathbf{l}^{\mathsf{T}}\bar{\mathbf{l}} = 0$) two solutions are obtained. One is the axis of symmetry and the other is the sought line.

Notice that there exist some degenerated cases in which projecting surfaces are planes (called Planar Viewing Surfaces (PVS) in [Gasparini and Caglioti, 2011]) and the geometry of the 3D line cannot be recovered. These degenerated cases are: the Axial-PVS case when the line is coplanar with the axis of symmetry and the Horizontal-PVS case when all the projecting rays lie in an horizontal plane ($\phi = \frac{\pi}{2}$).

A line-image is a curve on a two dimensional projected space which defines the collection of rays intersecting a 3D line. This curve is obtained by embedding the back-projection model in equation (4.7). Generalizing the framework proposed in [Bermudez-Cameo et al., 2012a] to non-central systems, the general expression for a line-image is

$$g(\mathbf{x}) = Z_r(l_2 x - l_1 y) + \left(\bar{l}_1 x + \bar{l}_2 y\right) + \bar{l}_3 r \cot\phi = 0\ . \tag{4.8}$$

where $r = \sqrt{x^2+y^2}$ (see Fig. 4.3), l_1,l_2,l_3 are the components of $\mathbf{l}$ and $\bar{l}_1$,$\bar{l}_2$,$\bar{l}_3$ are the components of $\bar{\mathbf{l}}$. With at least 4 points of the line-image we can compute the 3D line in a direct way by solving the linear system

$$(-Z_{ri}y_i, Z_{ri}x_i, x_i, y_i, r_i\cot\phi_i)\,\tilde{\mathbf{L}} = 0\ ,\ \text{for } i = 1,...,4\ , \tag{4.9}$$

where $\tilde{\mathbf{L}} = \left(l_1, l_2, \bar{l}_1, \bar{l}_2, \bar{l}_3\right)^{\mathsf{T}}$. Notice that l_3 has disappeared in the equation and we are obtaining an element of $\mathbb{P}^4$ because the system has revolution symmetry and $\bar{\xi}_3 = 0$. As a consequence, the null space is a single solution instead of the one dimension space obtained when solving (4.7) in [Teller and Hohmeyer, 1999]. Actually, the Plücker identity used to reduce this space in [Teller and Hohmeyer, 1999] is used here to compute l_3 due to the redundancy in Plücker coordinates representation.

4.4 Algebraic and metric distances

When evaluating if a point belongs to a line-image for fitting the curve is necessary a function measuring the distance from a point to the line-image. The quality of the extracted line-image changes depending on this distance. In this Section we present a qualitative comparison among distances and we propose different approaches for tackling with this problem.

Having the line-image equation (4.8) a measure of distance is the algebraic distance

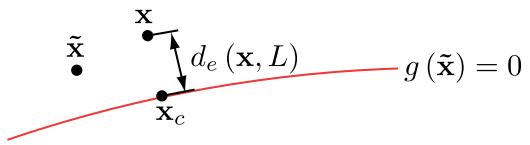


Figure 4.4: Euclidean distance from point $\mathbf{x}$ to line-image L.

$$d_{alg}(\mathbf{x}, L) = |Z_r(l_2 x - l_1 y) + (\bar{l}_1 x + \bar{l}_2 y) + \bar{l}_3 r \cot\phi| \tag{4.10}$$

which is measured in pixels when the Plücker coordinates of the line $\mathbf{L}$ are normalized with $\|\bar{\mathbf{l}}\| = 1$.

The second distance considered is the Euclidean metric distance in $\mathbb{E}^3$ which is the minimum Euclidean distance between two 3D lines (the 3D line and the projecting ray generated from an image point). The Euclidean metric distance is defined as in [Pottmann and Wallner, 2001]

$$d_E(\mathbf{L}_i, \mathbf{L}_j) = \frac{|\mathbf{l}_i{}^{\mathsf{T}} \bar{\mathbf{l}}_j + \mathbf{l}_j{}^{\mathsf{T}} \bar{\mathbf{l}}_i|}{\|\mathbf{l}_i \times \mathbf{l}_j\|} \tag{4.11}$$

which is defined in 3D space units (meters).

Both distances (d_{alg} and d_E) tend to give higher reward to lines which are close to the origin, conditioning the robust extraction process. Besides, the equidistant region defined by these distances is not homogeneous on the image plane (see Fig. 4.5), particularly in the case of conical catadioptric camera.

To avoid this effect the metric distance $\mathbb{E}^2$ from point to line-image can be used. Consider the Euclidean distance between two image points $\mathbf{x}_i$, $\mathbf{x}_j$ satisfying the three metric conditions: no negativity, symmetry and triangle inequality

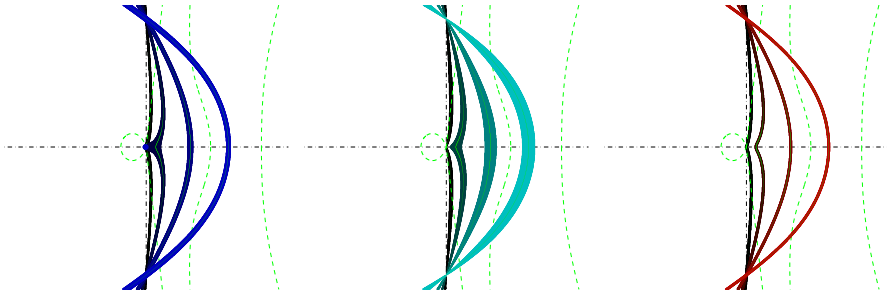
$$d_e(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^{\mathsf{T}} (\mathbf{x}_i - \mathbf{x}_j)}\,, \tag{4.12}$$

given a metric space (X, d_e) and $L \subset X$ a subset of points lying on a line-image, the metric distance from a point $\mathbf{x}$ to the line-image is

$$d_e(\mathbf{x}, L) = \inf\{d_e(\mathbf{x}, \mathbf{x}_l) : \mathbf{x}_l \in L\}\,, \tag{4.13}$$

which is the minimum distance from $\mathbf{x}$ to the line-image. The distance between $\mathbf{x}$ and the closest point of the line-image $\mathbf{x}_c$ is the minimum distance from point to line-image $d_e(\mathbf{x}, L) = d_e(\mathbf{x}, \mathbf{x}_c)$.

Let $\tilde{\mathbf{x}} = (\tilde{x}, \tilde{y})^{\mathsf{T}} \in \mathbb{R}^2$ be a point of the image plane, computing the point of the line-image $\mathbf{x}_c$ which is closer to the given point $\mathbf{x}$ (see Fig. 4.4) is equivalent to minimizing the distance function $f_d(\tilde{\mathbf{x}}) = d_e(\mathbf{x}, \tilde{\mathbf{x}})$ subject to the constraint $g(\tilde{\mathbf{x}}) = 0$ where g is the line-image equation as expressed in (4.8) or in any other representation depending



(a) (b) (c)

Figure 4.5: Behaviour of the different metrics defined to decide if a point lies on a line-image. The thin lines are the actual line-images. The coloured region around the lines denotes the points of the region which have a distance minor than a threshold. Example for conical catadioptric system. (a) d_{alg}. (b) d_E. (c) d_e.

on coordinates of the image. Since $\arg\min_{\mathbf{x}} d_e(\mathbf{x},L) = \arg\min_{\mathbf{x}} d_e^2(\mathbf{x},L)$ the problem is stated as

$$d_e(\mathbf{x},L) = d_e(\mathbf{x},\mathbf{x}_c) \text{ such that } \mathbf{x}_c = \arg\min_{\tilde{\mathbf{x}}} f_d^2(\tilde{\mathbf{x}}) \text{ subject to } g(\tilde{\mathbf{x}}) = 0\,. \tag{4.14}$$

The method of Lagrange multipliers justifies that this statement is equivalent to find the critical points of the Lagrangian function

$$\mathscr{L}(\tilde{\mathbf{x}},\lambda) = f_d^2(\tilde{\mathbf{x}}) + \lambda g(\tilde{\mathbf{x}}) = \|\tilde{\mathbf{x}}-\mathbf{x}\|^2 + \lambda g(\tilde{\mathbf{x}}) \tag{4.15}$$

which are estimated by solving $\nabla_{\tilde{x},\tilde{y},\lambda}\mathscr{L}(\tilde{x},\tilde{y},\lambda) = 0$.

Eliminating the Lagrange parameter λ we finally obtain the equation system

$$g(\tilde{\mathbf{x}}) = 0 \;\;,\quad h(\tilde{\mathbf{x}}) = (\tilde{x}-x)\frac{\partial g}{\partial \tilde{y}} - (\tilde{y}-y)\frac{\partial g}{\partial \tilde{x}} = 0, \tag{4.16}$$

in such a way that the point $\tilde{\mathbf{x}}$ is on the line-image and satisfies that the perpendicular line passing by this point must intersect $\mathbf{x}$.

In Fig. 4.5 we show the different regions defined by the three distances given a threshold. We show the case of conical catadioptric system. Fig. 4.5 (a) corresponds to the regions for the algebraic distance d_{alg}, observing that there exist variations in the thickness of the region when the projection is close to the center of the image. The region defined by the 3D Euclidean distance between line and rays d_E (see Fig. 4.5 (b)) variates its thickness depending on the distance of the point of the 3D line. Finally, the Euclidean distance on the image d_e defines a region with an homogeneous thickness (see Figure 4.5 (c)).

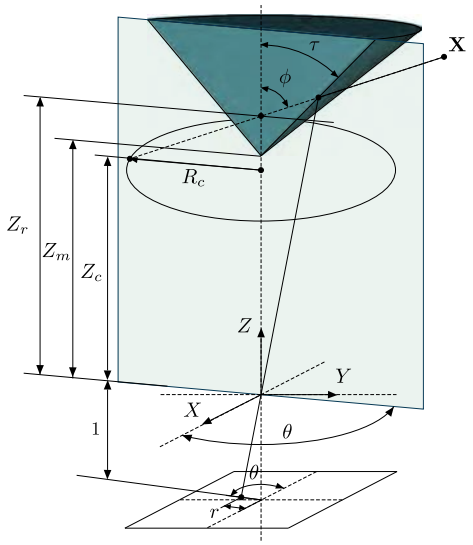


Figure 4.6: Conical catadioptric projection of a point **X**.

In Section 4.5.2 and Section 4.6.2 we solve (4.16) for the two selected cases of conical catadioptric systems and the spherical catadioptric systems finding analytical solutions which are roots of a polynomial. This is possible because for these cases the line-image equation g can be written as a polynomial but in general the solution has to be solved using iterative methods for minimizing (4.16).

4.5 Conical mirror systems

In conical catadioptric systems with the camera located in the axis of revolution of the mirror, the locus of viewpoint is a circle of radius R_c centred in the vertical axis at height Z_c [Baker and Nayar, 2001, López-Nicolás and Sagüés, 2010]. The locus of this circle, which depends on the distance Z_m between the camera and the vertex of the mirror and the aperture angle τ of the mirror, is

$$R_c = Z_m \sin 2\tau \quad , \quad Z_c = Z_m \left(1 - \cos 2\tau\right). \tag{4.17}$$

The forward projection model. Since the viewpoint locus is a circle and there exist revolution symmetry the forward projection is unambiguous and direct. Given a 3D point $\mathbf{X} = (X_1, X_2, X_3, X_4)^\mathsf{T} \in \mathbb{P}^3$ in the camera reference, the non-central projection ray lies in a plane containing the axis of revolution of the mirror (See Fig. 4.6). The intersection of this plane with the circle is a single point **C**, therefore the projection ray

is completely defined by $\mathbf{X}$ and $\mathbf{C}$. The intersection of this line with the mirror gives the point $\mathbf{x}_I$ which is projected in point $\mathbf{x} \in \mathbb{P}^2$ on the normalized plane.

$$\begin{aligned} x &= \left(\sin 2\tau \frac{(X_3 - Z_m X_4)}{\sqrt{X_1^2 + X_2^2}} - \cos 2\tau \right) X_1 \\ y &= \left(\sin 2\tau \frac{(X_3 - Z_m X_4)}{\sqrt{X_1^2 + X_2^2}} - \cos 2\tau \right) X_2 \\ z &= Z_m X_4 + (X_3 - Z_m X_4)\cos 2\tau + \sqrt{X_1^2 + X_2^2}\sin 2\tau \end{aligned} \tag{4.18}$$

This projection is related to the image plane with a perspective camera model involving a linear transformation and a distortion model. We use the matrix K_c (see (4.5) to transform the coordinates of the normalized plane $(x, y, z) \in \mathbb{P}^2$ to the image coordinates $(u, v) \in \mathbb{R}^2$. The principal point (u_0, v_0) corresponds to the vertex cone projection.

The back projection model. In conical catadioptric systems Z_r depends on $\cot\phi$

$$Z_r = Z_c + R_c \cot\phi \tag{4.19}$$

which is related with r by

$$\cot\phi = \frac{z + r\tan 2\tau}{z\tan 2\tau - r}\,. \tag{4.20}$$

Substituting these expressions in (4.6) the equation of the back projection model becomes

$$\Xi = \begin{pmatrix} x(z\tan 2\tau - r) \\ y(z\tan 2\tau - r) \\ r(z + r\tan 2\tau) \\ -y\left(Z_c(z\tan 2\tau - r) + R_c(z + r\tan 2\tau)\right) \\ x\left(Z_c(z\tan 2\tau - r) + R_c(z + r\tan 2\tau)\right) \\ 0 \end{pmatrix}. \tag{4.21}$$

4.5.1 Lines-images in conical catadioptric systems

When particularizing expression (4.8) to the conical catadioptric system, the line image equation results on a polynomial expression of degree 4 (a quartic described by 15 monomials in general). However, when the equation is expressed in polar coordinates ($x = r\cos\theta$, $y = r\sin\theta$) this expression can be written in a compact form with 6 parameters encapsulating the Plücker coordinates of the line and the mirror parameters of the system. The line-image is then written as

$$(r\cos\theta, r\sin\theta, r, z\cos\theta, z\sin\theta, z)\,\omega = 0 \tag{4.22}$$

where

$$\omega = \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \\ \omega_5 \\ \omega_6 \end{pmatrix} = \begin{pmatrix} \frac{1-\cos 2\tau}{\cos 2\tau} Z_m l_2 - \bar{l}_1 \\ -\frac{1-\cos 2\tau}{\cos 2\tau} Z_m l_1 - \bar{l}_2 \\ \bar{l}_3 \tan 2\tau \\ \tan 2\tau \left(\bar{l}_1 + Z_m l_2\right) \\ \tan 2\tau \left(\bar{l}_2 - Z_m l_1\right) \\ \bar{l}_3 \end{pmatrix} . \tag{4.23}$$

This expression allows us to linearly compute the line-image from five or more points without knowing neither the aperture angle of the mirror τ nor the distance to the mirror Z_m, by solving

$$\left(r_i x_i, r_i y_i, r_i^2, x_i z_i, y_i z_i, r_i z_i\right) \omega = 0 \text{ for } i = 1, ..., 5 . \tag{4.24}$$

Once the line-image $\omega \in \mathbb{P}^5$ is estimated τ is easily computed from $\tan 2\tau = \omega_3/\omega_6$. Notice that the distance to the mirror Z_m is coupled with direction vector $\mathbf{l}$ so it is not possible to separate them. Because of this, we conclude that in conical catadioptric mirrors, if the distance of the camera to the mirror Z_m is unknown, it is not possible to reconstruct the scale of a scene from line-images in a single image.

Parametric Description and Singularity

Expression (4.22) allows expressing r in terms of θ

$$r = \frac{-\left(\omega_4 \cos\theta + \omega_5 \sin\theta + \omega_6\right)}{\omega_1 \cos\theta + \omega_2 \sin\theta + \omega_3} \tag{4.25}$$

therefore, the parametric expression of the line-image curve becomes

$$x(\theta) = -\left(\omega_4 \cos\theta + \omega_5 \sin\theta + \omega_6\right) \cos\theta \tag{4.26}$$

$$y(\theta) = -\left(\omega_4 \cos\theta + \omega_5 \sin\theta + \omega_6\right) \sin\theta \tag{4.27}$$

$$z(\theta) = \omega_1 \cos\theta + \omega_2 \sin\theta + \omega_3 . \tag{4.28}$$

In conical catadioptric systems the vertex cone projection ($x = 0$, $y = 0$) is a singularity for line-images passing through it. If the line-image lies on this point, equation (4.25) returns negative values of r for some values of θ. At the singularity, the curve is continuous but not derivable. Considering the points with negative radius the curve is derivable on the singularity. Actually, these points are not obtained on the real catadioptric image (dotted points in Figure 4.7).

From equation (4.25) we can determine the range of values of θ in which the radius is negative. The values of θ limiting this range are computed from

$$\omega_4 \cos\theta + \omega_5 \sin\theta + \omega_6 = 0 , \tag{4.29}$$

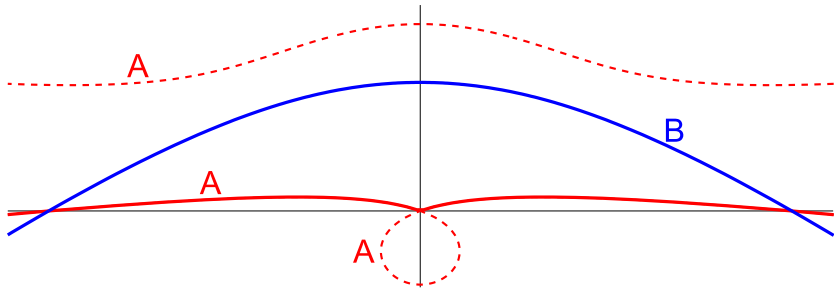


Figure 4.7: Detail of two line projections A and B in a non-central catadioptric image with conical mirror. Line A (in red) passes through the singularity at the principal point (dotted points correspond to negative radius).

and solving for $\tan\theta$ yields:

$$\tan\theta = \frac{-\omega_4\omega_5 \pm \omega_6\sqrt{\omega_4^2+\omega_5^2-\omega_6^2}}{\omega_5^2-\omega_6^2}. \tag{4.30}$$

Notice that it does not exist real solution if the value inside the square root is negative. In other words, all θ values (4.25) gives $r > 0$ and the line-image does not belong to the singularity. So, we can state that a line-image passes through the singularity if and only if $\omega_4^2+\omega_5^2 > \omega_6^2$.

4.5.2 Algebraic and metric distances in conical catadioptric systems

The Euclidean distance $d_e(\mathbf{x}, L)$ between the image point $\mathbf{x}$ and the line-image L is obtained estimating the closest point of the line-image $\mathbf{x}_c$. In Section 4.4 we have shown that the point $\tilde{\mathbf{x}} \in \mathbb{E}^2$ of the Euclidean plane, is the closest point of the line-image $\mathbf{x}_c$ when satisfying $g(\tilde{\mathbf{x}}) = 0$ and $h(\tilde{\mathbf{x}}) = 0$. Particularizing equation (4.16) to conical catadioptric systems we can reach an analytical solution. First we expand the compact line-image description (4.22) to get a polynomial equation for $g(\tilde{\mathbf{x}}) = 0$, in this case the quartic

$$g(\tilde{\mathbf{x}}) = r^2(\omega_6+\omega_1\tilde{x}+\omega_2\tilde{y})^2 - \left(\omega_4\tilde{x}+\omega_5\tilde{y}+\omega_3\tilde{r}^2\right)^2 = 0 \tag{4.31}$$

where $\tilde{r}^2 = \tilde{x}^2+\tilde{y}^2$. The resulting polynomial equation $h(\tilde{\mathbf{x}}) = 0$ obtained from this expression is the quartic

$$h(\tilde{\mathbf{x}}) = \left(\tilde{x}^3, \tilde{x}^2\tilde{y}, \tilde{x}^2, \tilde{x}\tilde{y}^2, \tilde{x}\tilde{y}, \tilde{x}, \tilde{y}^3, \tilde{y}^2, \tilde{y}\right) \mathsf{W} \begin{pmatrix} \tilde{x}-x \\ \tilde{y}-y \end{pmatrix} = 0 \tag{4.32}$$

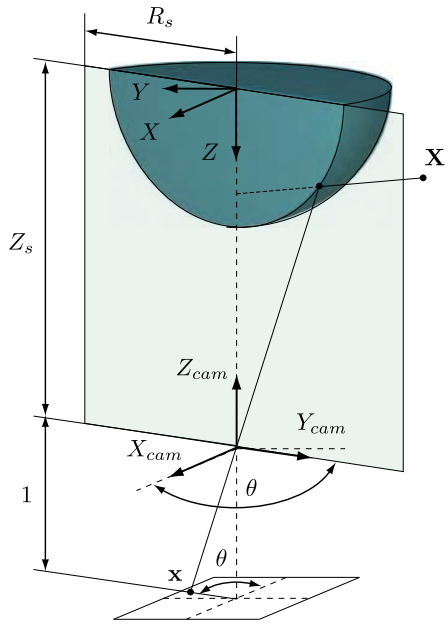


Figure 4.8: Spherical catadioptric projection of a point $\mathbf{X}$.

where

$$\mathsf{W} = \begin{pmatrix} \omega_1\omega_2 & 2\omega_3^2 - 2\omega_1^2 \\ \omega_1^2 + \omega_2^2 - 2\omega_3^2 & -3\omega_1\omega_2 \\ \omega_2\omega_6 - \omega_3\omega_5 & 3\omega_3\omega_4 - 3\omega_1\omega_6 \\ 3\omega_1\omega_2 & -\omega_1^2 - \omega_2^2 + 2\omega_3^2 \\ 2\omega_1\omega_6 - 2\omega_3\omega_4 & 2\omega_3\omega_5 - 2\omega_2\omega_6 \\ -\omega_4\omega_5 & \omega_4^2 - \omega_6^2 \\ 2\omega_2^2 - 2\omega_3^2 & -\omega_1\omega_2 \\ 3\omega_2\omega_6 - 3\omega_3\omega_5 & \omega_3\omega_4 - \omega_1\omega_6 \\ \omega_6^2 - \omega_5^2 & \omega_4\omega_5 \end{pmatrix} \tag{4.33}$$

Computing the resultant between both equations from the variable $\tilde{y}$ we obtain a single polynomial equation depending on $\tilde{x}$ with degree 12.

$$\left(\tilde{x}^{12}, \tilde{x}^{11}, \tilde{x}^{10}, \tilde{x}^{9}, \tilde{x}^{8}, \tilde{x}^{7}, \tilde{x}^{6}, \tilde{x}^{5}, \tilde{x}^{4}, \tilde{x}^{3}, \tilde{x}^{2}, \tilde{x}, 1\right) \Omega\hat{x} = 0 \tag{4.34}$$

where $\Omega(\boldsymbol{\omega}) \in \mathbb{R}^{13x15}$ and $\hat{x} \in \mathbb{R}^{15}$ such that $\hat{x} = \left(x^4, x^3y, x^3, x^2y^2, x^2y, x^2, xy^3, xy^2, xy, x, y^4, y^3, y^2, y, 1\right)^{\mathsf{T}}$.

For a given point on the normalized plane $(x,y)^\mathsf{T}$, solving (4.34) results in 12 solutions for $\tilde{x}$. The correct solution can be found by checking g and h for each solution. From the remaining results we choose the one with minimal distance. Notice that Ω only has to be computed for each line-image ω whereas (4.34) is solved for each point.

4.6 Spherical mirror systems

An spherical catadioptric system is a non-central system composed by an spherical mirror and a perspective camera. Due to the symmetry of the sphere there exist symmetry of revolution wherever the perspective camera is located. The system is characterized by the radius of the sphere R_s, the distance from the camera to the center of the sphere Z_s, the intrinsic parameters of the perspective camera and the relative rotation of the camera with respect to the axis of revolution. This axis is defined by the center of the sphere and the location of the perspective camera.

Forward projection model. There is no single closed solution for the forward projection model in an spherical catadioptric system. The problem is tackled in [Agrawal and Ramalingam, 2013,Agrawal et al., 2011,Gonçalves, 2010,Gonçalves and Nogueira, 2009].

Back projection model. Equation (4.6) can be particularized to the case of the spherical catadioptric system obtaining the back projection model. Consider a projective point (x,y,z) in the reference of the perspective camera with the axis Z_{cam} aligned with the axis of revolution. Given $r^2 = x^2+y^2$ and $\rho^2 = x^2+y^2+z^2$ the Plücker coordinates of a projection ray expressed on the reference of the spherical mirror X,Y,Z (See Fig. 4.8, this reference is chosen for simplicity of the expressions) is

$$\Xi = (-x\delta, y\delta, -\zeta, \varepsilon y Z_s, \varepsilon x Z_s, 0)^\mathsf{T} \tag{4.35}$$

where

$\delta = 2r^2 Z_{Rel}^4 - 2z\sqrt{\gamma} Z_{Rel}^2 - 3\rho^2 Z_{Rel}^2 + \rho^2$,

$\varepsilon = \left(-r^2+z^2\right) Z_{Rel}^2 + 2\sqrt{\gamma} z + \rho^2$,

$\zeta = 2r^2 z Z_{Rel}^4 - z\rho^2 Z_{Rel}^2 - 2\sqrt{\gamma}\left(-r^2 Z_{Rel}^2 + \rho^2\right) - z\rho^2$,

$\gamma = \left(-r^2 Z_{Rel}^2 + \rho^2\right) Z_{Rel}^2$ and $Z_{Rel} = \frac{Z_s}{R_s}$.

4.6.1 Line images in spherical mirror systems

Following (4.7) the side operator between a line $\mathbf{L} = \left(\mathbf{l}^\mathsf{T}, \overline{\mathbf{l}}^\mathsf{T}\right)^\mathsf{T}$ and the projection ray Ξ defines the line image equation as

$$\delta\left(x\bar{l}_1 - y\bar{l}_2\right) + \zeta\bar{l}_3 - \varepsilon Z_s\left(yl_1 + xl_2\right) = 0 \tag{4.36}$$

Considering $\hat{\mathbf{l}} = Z_s\mathbf{l}$ the expression becomes

$$\delta\left(x\bar{l}_1 - y\bar{l}_2\right) + \zeta\bar{l}_3 - \varepsilon\left(y\hat{l}_1 + x\hat{l}_2\right) = 0 \tag{4.37}$$

which can be rewritten as

$$g\left(\tilde{\mathbf{x}}\right) = \tilde{\mathbf{L}}^{\mathsf{T}}\mathsf{N}\tilde{\mathbf{X}} = 0 \tag{4.38}$$

where

$$\tilde{\mathbf{L}} = \left(\hat{l}_1^2, \hat{l}_1\hat{l}_2, \hat{l}_1\bar{l}_1, \hat{l}_1\bar{l}_2, \hat{l}_1\bar{l}_3, \hat{l}_2^2, \hat{l}_2\bar{l}_1, \hat{l}_2\bar{l}_2, \hat{l}_2\bar{l}_3, \bar{l}_1^2, \bar{l}_1\bar{l}_2, \bar{l}_1\bar{l}_3, \bar{l}_2^2, \bar{l}_2\bar{l}_3, \bar{l}_3^2\right)^{\mathsf{T}}, \tag{4.39}$$

$$\tilde{\mathbf{X}} = \left(\tilde{x}^4, \tilde{x}^3\tilde{y}, \tilde{x}^3\tilde{z}, \tilde{x}^2\tilde{y}^2, \tilde{x}^2\tilde{y}\tilde{z}, \tilde{x}^2\tilde{z}^2, \tilde{x}\tilde{y}^3, \tilde{x}\tilde{y}^2\tilde{z}, \tilde{x}\tilde{y}\tilde{z}^2, \tilde{x}\tilde{z}^3, \tilde{y}^4, \tilde{y}^3\tilde{z}, \tilde{y}^2\tilde{z}^2, \tilde{y}\tilde{z}^3, \tilde{z}^4\right)^{\mathsf{T}}, \tag{4.40}$$

and

$$\mathsf{N} = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 1 & 2\nu & 0 & 0 & \nu^2 & 0 & 0 & 0 & 0 & \nu^2-1 \\
0 & 2 & 2\nu & 0 & 0 & 0 & 0 & -2\nu & 0 & 0 & -2\nu^2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2\nu & 0 & 0 & -2 & 0 & 0 & 0 \\
1 & 0 & 0 & -2\nu & 0 & 1 & 2\nu & 0 & 0 & \nu^2 & 0 & 0 & \nu^2 & 0 & 2\nu^2-2 \\
0 & 0 & 0 & 0 & -2\nu & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & -2 & 0 & 0 & -2\mu & 0 & 0 & 0 & 0 & -2\mu \\
0 & 2 & 2\nu & 0 & 0 & 0 & 0 & -2\nu & 0 & 0 & -2\nu^2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2\nu & 0 & 0 & -2 & 0 & 0 & 0 \\
0 & 2 & -2 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 2\mu & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & -2 & 0 & 0 & 0 \\
1 & 0 & 0 & -2\nu & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \nu^2 & 0 & \nu^2-1 \\
0 & 0 & 0 & 0 & -2\nu & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\
1 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2\mu & 0 & -2\mu \\
0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}^{\mathsf{T}} \tag{4.41}$$

with $\nu = 2Z_{Rel}^2 - 1$ and $\mu = 2\nu + 1$.

Notice that $\mathbf{L}$ has been defined in the reference system of the spherical mirror (see X,Y and Z in Fig. 4.8). This reference system has been chosen because of the simplicity of the expressions and is different of the camera reference chosen for the conical catadioptric system. The transformation between both reference systems is

$$\mathsf{G} = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 \\
0 & Z_s & 0 & 1 & 0 & 0 \\
Z_s & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & -1
\end{pmatrix} \text{ such that } \mathsf{G} = \mathsf{G}^{-1}\ . \tag{4.42}$$

Parametric Description

The parametric description of the line-image can be obtained composing the lifted image vector $\tilde{\mathbf{X}}$ with the polar representation $x = r\cos\theta$, $y = r\sin\theta$ and $z = 1$. The result is a polynomial of degree 4 in r which can be solved when θ is known. Four different solutions are obtained but only one of them corresponds to the sought parametric line-image. Points corresponding to the right solution must also satisfy equation (4.38).

4.6.2 Euclidean distance d_e in spherical catadioptric systems

The Euclidean distance $d_e(\mathbf{x}, L)$ between the image point $\mathbf{x}$ and the line-image L is obtained estimating the closest point of the line-image $\mathbf{x}_c$. In Section 4.4 we have shown that the point $\tilde{\mathbf{x}} \in \mathbb{E}^2$ of the Euclidean plane, is the closest point of the line-image $\mathbf{x}_c$ when satisfying the systems of equations (4.16). In particular, when considering an spherical catadioptric system, the function $g(\tilde{\mathbf{x}})$ is defined by the polynomial representation of the line-image (4.38). The constraint $h(\tilde{\mathbf{x}})$ enforces the parallelism between the gradient of the line-image and the line passing through the closest point of the line-image and the given point. Substituing the gradients of expression (4.38) in (4.16) we obtain

$$h(\tilde{\mathbf{x}}) = \tilde{\mathbf{L}}^{\mathsf{T}} \mathsf{N} \mathsf{J} \tilde{\mathbf{X}} = 0 \tag{4.43}$$

where $\mathsf{J}(\mathbf{x})$ is

$$\mathsf{J} = \begin{pmatrix}
0 & -4 & 4y & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & -x & -3 & 3y & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -3 & 3y & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 2 & 0 & 0 & -2x & 0 & -2 & 2y & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & -x & 0 & -2 & 2y & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 2y & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 3 & 0 & 0 & 0 & -3x & 0 & 0 & -1 & y & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & -2x & 0 & 0 & -1 & y & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -x & 0 & 0 & -1 & y & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & y \\
0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & -4x & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & -3x & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & -2x & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -x \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}. \tag{4.44}$$

Both expressions $g(\tilde{\mathbf{x}}) = 0$ and $h(\tilde{\mathbf{x}}) = 0$ constitute a polynomial system of equations of degree four. Considering $z = 1$ and the operator

$$\mathrm{tri}(\mathbf{v}) = \begin{pmatrix}
0 & 0 & 0 & 0 & v_1 \\
0 & 0 & 0 & v_2 & v_3 \\
0 & 0 & v_4 & v_5 & v_6 \\
0 & v_7 & v_8 & v_9 & v_{10} \\
v_{11} & v_{12} & v_{13} & v_{14} & v_{15}
\end{pmatrix}$$

these equations can be rewritten on terms of a bi-quartic expression.

$$g = \tilde{\mathbf{x}}_Q^{\mathsf{T}} \mathrm{tri}\left(\tilde{\mathbf{L}}^{\mathsf{T}} \mathsf{N}\right) \tilde{\mathbf{y}}_Q = 0 \tag{4.45}$$

$$h = \tilde{\mathbf{x}}_Q^{\mathsf{T}} \mathrm{tri}\left(\tilde{\mathbf{L}}^{\mathsf{T}} \mathsf{NJ}\right) \tilde{\mathbf{y}}_Q = 0 \tag{4.46}$$

where $\tilde{\mathbf{x}}_Q = \left(\tilde{x}^4, \tilde{x}^3, \tilde{x}^2, \tilde{x}, 1\right)^{\mathsf{T}}$ and $\tilde{\mathbf{y}}_Q = \left(\tilde{y}^4, \tilde{y}^3, \tilde{y}^2, \tilde{y}, 1\right)^{\mathsf{T}}$.

One of the two variables of the systems of equations can be eliminated computing the resultant between them. For this, we compute the Sylvester matrix S respect to $\tilde{y}$ obtaining a 8×8 matrix. The determinant of this matrix is the resultant, that is a polynomial equation such that

$$|S| = 0 \Leftrightarrow g = 0 \text{ and } h = 0. \tag{4.47}$$

To reduce the difficulty of computing this determinant we exploit the property of determinants

$$|S| = \begin{vmatrix} S_A & S_B \\ S_C & S_D \end{vmatrix} = |S_A| \left|S_D - S_C S_A^{-1} S_B\right| .$$

Since $|S_A| \neq 0$ the degree of the resultant decreases, obtaining the determinant of a 4×4 matrix

$$\left|S_D - S_C S_A^{-1} S_B\right| = 0 \tag{4.48}$$

which is a polynomial of degree 16 in $\tilde{x}$. The roots of this polynomial are the solutions of $\tilde{x}$ for the equation system. Using (4.45) we obtain 4 solutions of $\tilde{y}$ for each of the 16 solutions obtained for $\tilde{x}$. Each pair $\tilde{\mathbf{x}}_i = (\tilde{x}_i, \tilde{y}_i)^{\mathsf{T}}$ corresponds to a candidate to be the closer point from the line-image to the original point $\mathbf{x}$. The right solutions must hold (4.45). From the remaining solutions we choose the one with minimal Euclidean distance d_e.

4.7 Robust line extraction in non-central systems

In this Section we present a method for line-image extraction in non-central systems. Line extraction in non-central systems is an unsolved challenging task due to the difficulties of mapping the 3D of the line on a 2D space, the sensitivity to noise in fitting, and the elevate number of degrees of freedom involved. Up to our knowledge, this is the first work addressing this problem.

Assuming the system is calibrated, four points of the line-image are needed to define a line-image and its corresponding 3D line. When using a robust approach like RANSAC the number of iterations is considerably greater than in the central case because the DOFs increase from 2 to 4. To handle this problem we can check the candidates for minimal subsets before computing the minimal model like in PROSAC [Chum and Matas, 2005] and USAC [Raguram et al., 2013]. That allows to reduce the number of hypothesis being computed and evaluated. In particular, we remove collections of four samples having low effective baseline. The baseline of a set of four

points is estimated using our proposal for measuring the effective baseline presented in the following Section. To evaluate if an image point belongs to a particular line-image hypothesis we use the metric distance presented in Section 4.4.

4.7.1 A feature for measuring effective baseline in a set of rays

To compute a 3D line from a set of points rays (see Section 4.3.2) the rays must not be coplanar. Consider for example a set of four rays intersecting a line, two of them being projections in a camera and the other two projections in other camera of a stereo pair. If we get a ray from a camera and a ray from the other camera they are skew and there exist a distance between them related with the baseline of the stereo system. The accuracy of the line reconstruction is strongly related with this distance. Similarly, when having a non-central system, the accuracy of the reconstruction of a 3D line from a single projection strongly depends on the distances between the defining rays which depend on the geometry and the size of the imaging system. In this section we propose a feature measuring the quality of a set of rays for reconstructing a 3D line. Establishing a similarity with the stereo pair we call this measure *effective baseline*.

The proposed feature is based on the 3D Euclidean distances among the defining rays d_E (4.11). Consider the combination of distances among a set of defining rays; a simple feature is just the mean among all these distances. However, the mean distance is not a solid choice. For example we can have a high mean distance due to a sole distant ray but if the other rays are coplanar the configuration is degenerated.

To measure the effective baseline of a set of n rays our proposal is the function z_{bs} depending on the metric distances $d_E(\Xi_i, \Xi_j)$ among the defining rays

$$z_{bs} = \frac{n!}{2\,(n-2)!} \left(\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{1}{d_E(\Xi_i, \Xi_j)} \right)^{-1} \tag{4.49}$$

which takes into account the balance among the distances. This function increases when the distances between rays increase but also assures that not any individual distance $d_E(\Xi_i, \Xi_j)$ turns to zero. The feature is normalized with the number of combinations $\frac{n!}{2(n-2)!}$ in order to have a scaled measure.

To illustrate our reasoning let us first consider a simple case. Let us consider two points A and B in $\mathbb{R}^2$ and we want to evaluate a third point using the distances from this third point to the other two (d_A and d_B). The higher the distances the better the result, but we need to penalize small individual distances.

If we use the mean of the distances we have infinite cases and some of these cases with values of d_A or d_B close to zero (see Fig 4.9 (a)) which are degenerated cases. However, inspired by electrical potentials, we could use the sum of the inverse of the distance defining a potential surface which is the minimum at the equidistant point (see Fig 4.9 (b)) which is the best conditioned case. However, this measure tends to infinity when one of the distances is zero, hence considering the inverse of this (see Fig 4.9 (c)) we have a feature which is maximum when the distances are balanced.

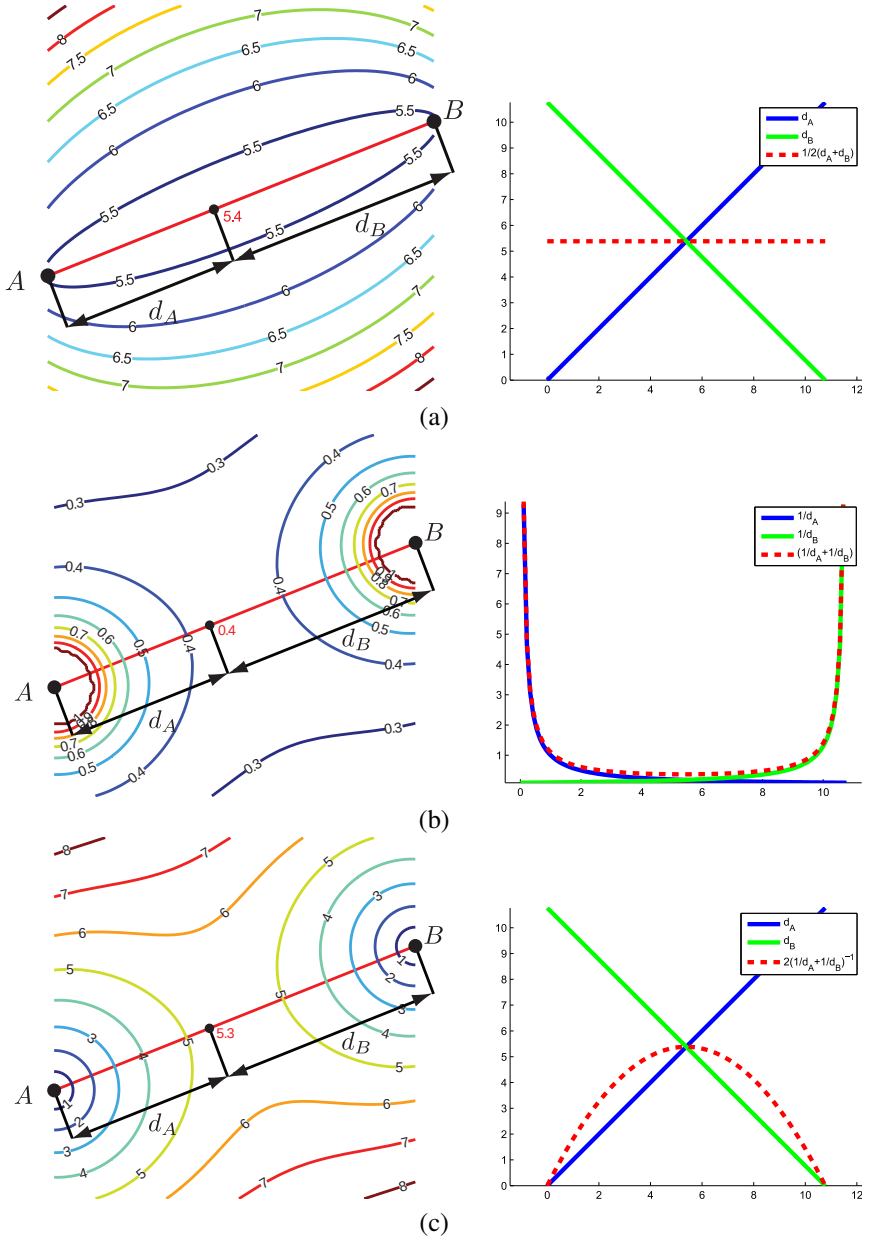


Figure 4.9: Different proposals for computing a measure of baseline of a set of rays based on the distances between rays. (a) Mean of the distances. (b) Sum of inverse of the distances. (c) Our proposal.

4.7.2 Extraction algorithm

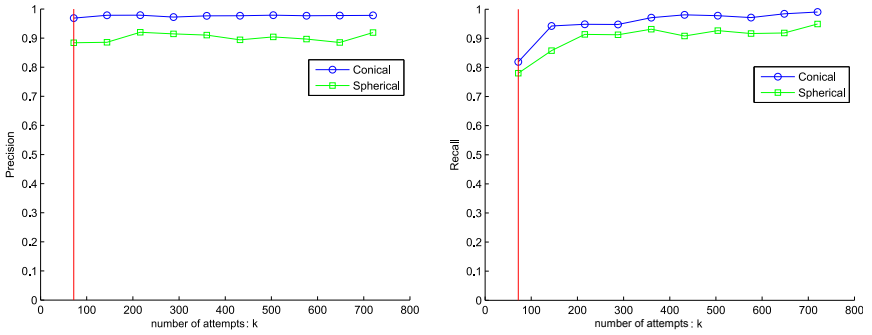
The proposed extraction procedure is the following. First, the image is preprocessed using a Rolling Guidance Filter [Zhang et al., 2014] to reduce textured patterns but conserving the edges. Then, the edges of the image are detected using Canny detector and stored in connected components. Each of these connected components can contain one or more line-images. A particular line-image can also be distributed along different connected components. We have considered two different robust strategies for solving the multi-fitting process required to extract line-images from the edges.

Greedy PROSAC approach. In this approach we check the candidates for minimal subsets using the effective-baseline measure. From the subsets that achieve this previous test we compute the line-image models. Then, all the points of the connected component are tested using the corresponding metric distance. In robust approaches like RANSAC it is assumed that the input of the algorithm is a collection of points fitting a single model. The points not supporting the model are discriminated as outliers. When having more than one model or line-image, the greedy approach considers a cascade application of the robust estimation. The model best supporting the collection of points is extracted. For the other models there are two options: First repeating the complete extraction on the remaining points until reaching some stop criterion. Second, removing the inlier points from the original votes matrix of the first extraction and use these corrected votes for selecting new lines. Second way is faster because hypothesis and distances are computed only once.

Greedy approach selecting a subset of well conditioned points. In this approach we compute a subset of well conditioned points from each connected component. The subset of points is big enough to homogeneously cover the whole connected component and it is computed using the effective baseline measure to assure that there exist enough baseline among any collection of four points from this subset. Hypothesis are generated from this subset of points and then tested using the metric distance. The distance can be computed from each hypothesis to the whole collection of points of the connected component or from the subset of points only. In this case, the supporting points are computed only on the best voted solution. Another option is testing the hypothesis with points of the whole image. The greedy cascade approach is used for the multi-fitting process using any of the previously presented methods.

4.8 Experimental evaluation

In this section, we present the experimental evaluation of the proposal. First, we show simulations to study the behaviour of the proposed method. Then we present selected examples of line-extraction in synthetic and real images. Experiments with synthetic



(a) (b)

Figure 4.10: Precision and recall when discriminating between two line-images.

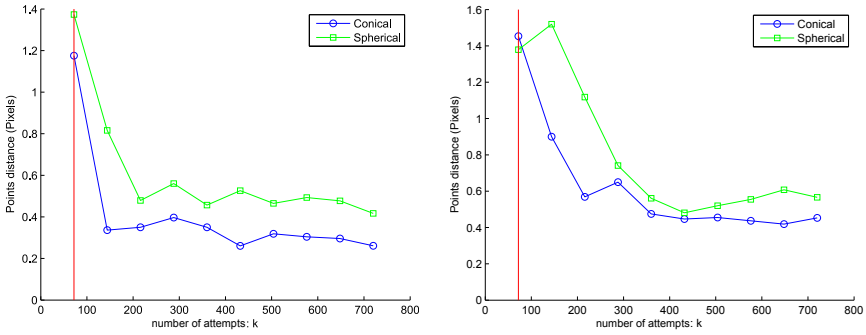
images allows testing the complete algorithm knowing the ground truth and with absence of calibration errors. The extraction method is also tested on images taken with real catadioptric systems. Finally we present several examples for estimating the calibration in conical catadioptric systems using real images.

4.8.1 Validation of the proposals using simulated projections

For validating the proposals we have performed intensive simulations of line projections where the ground truth is known and randomly generated, the noise and calibration error are controlled and it is possible to obtain results with stochastic meaning. The set-up of the simulation is the following: We have generated 64 random pairs of 10 m length 3D segments passing through a cube 4 meters wide around the visual system. For each segment, points are projected on the image using the forward projection model. The size of all the simulated systems is similar to avoid the bias in effective baseline. The conical mirror has an angle $\tau = 45\ deg$ and the distance from the perspective camera to the mirror is $Z_m = 1\ m$. The spherical mirror has a radius of $R_s = 1\ m$ and the distance from the perspective camera to the center of the sphere is $Z_s = 2\ m$ (the distance to the mirror plus the radius). The resolution of the perspective camera used in both systems is 4096×4096 pixel with a focal distance of 0.625 m and no radial nor tangential distortion.

Robustness evaluation of line-image extraction

In this section we evaluate the robustness of the extraction process. This is performed by discriminating between two different line-images. 64 pairs of random segments with the previous set-up are projected. Points supporting the first line-image are considered inliers and points supporting the other are considered spurious. We extract the line-image variating the used attempts in the RANSAC scheme taking as reference the



(a) (b)

Figure 4.11: Comparison of conical and spherical catadioptric systems in terms of distance of estimated line-images to ground truth points. The vertical red line corresponds to the theoretical value of $k_{attempts}$. (a) Without Gaussian noise. (b) With a Gaussian noise of $\sigma = 0.5$ *pixels*.

theoretical value $k_{attempts} = \frac{\log(1-P)}{\log(1-\eta^p)}$ where p is the number of elements defining the minimal set, P the probability that at least in one random subset p all selected points are inliers and $\eta = \frac{number\ of\ inliers}{total\ number\ of\ points}$.

In Fig. 4.10 we show the results of precision and recall depending on the used attempts in robust extraction, where

$$precision = \frac{true\ positives}{true\ positives + false\ positives}$$

and

$$recall = \frac{true\ positives}{true\ positives + true\ negatives}\ .$$

In line-extraction high precision is needed due to a single false positive can distort the fitting. To evaluate the quality of the extraction we propose using the metric distance from point to line-image. In particular we measure the median of the distances from the fitted line-image to the ground truth points projected from the original line.

In Fig. 4.11 we show the mean of this measures for the collection of 64 random lines when variating the number of attempts $k_{attempts}$ of the RANSAC algorithm. We can observe how quality depends on the number attempts in both cases.

Accuracy of 3D line from single view

In this section we present an evaluation of the 3D accuracy of the extracted lines from a single projection. We also compare, in terms of 3D line accuracy, the conical catadioptric system and the spherical catadioptric system. Once image points of the line-image are projected from segments, we add Gaussian noise with a standard deviation

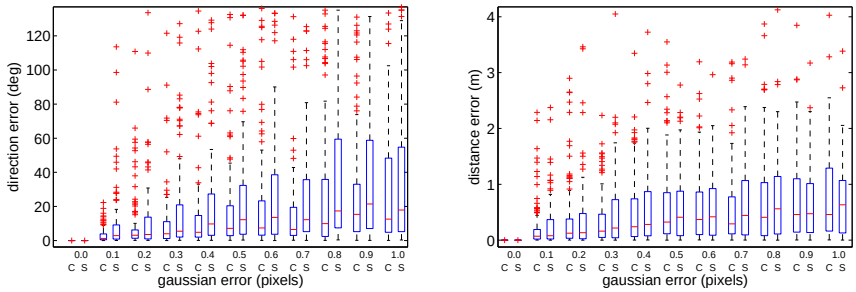


Figure 4.12: Comparison between conical and spherical catadioptric systems in terms of accuracy in lines reconstruction from single projection. (a) Direction error in degrees. (b) Distance error in meters. C for conical catadiotpric system and S for spherical catadioptric system.

σ which varies from 0 to 1 pixel. Then the lines are fitted using the points which maximizes the effective baseline.

In boxplot of Fig. 4.12 we present the estimated accuracy showing the distribution of two errors:

- the direction error between the estimated 3D line and the ground truth line computed as
  $\varepsilon_\varphi = \arccos \mathbf{l}^\top \mathbf{l}_{GT}$.
- the depth error between the estimated 3D line and the ground truth line computed as $\varepsilon_d = \left| \|\bar{\mathbf{l}}\| - \|\bar{\mathbf{l}}_{GT}\| \right|$.

We can observe how the accuracy in 3D fitting dramatically decreases with noise. In terms of accuracy in direction estimation the conical catadioptric system is a little better than the spherical catadioptric system in spite of having similar sizes.

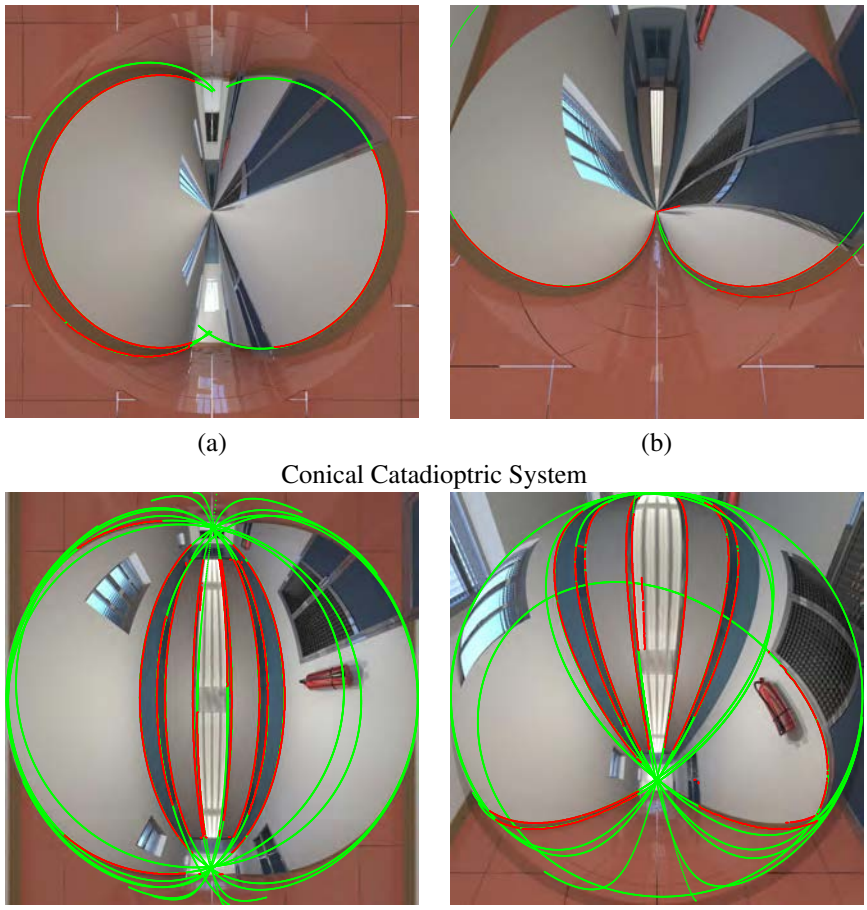
4.8.2 Line-image extraction

In this section we present the experiments for evaluating the complete extraction algorithm on synthetic and real images.

Synthetic images

Synthetic images have been generated using the raytracing software Pov-Ray [of Vision Pty. Ltd. (2004),] modelling the mirrors as geometric forms with perfect reflection. The scenario is a modification of a publicly available synthetic scenario [1]. The original

[1] http://hof.povray.org/office-13.html The Office - Jaime Vives Piqueres, 2004



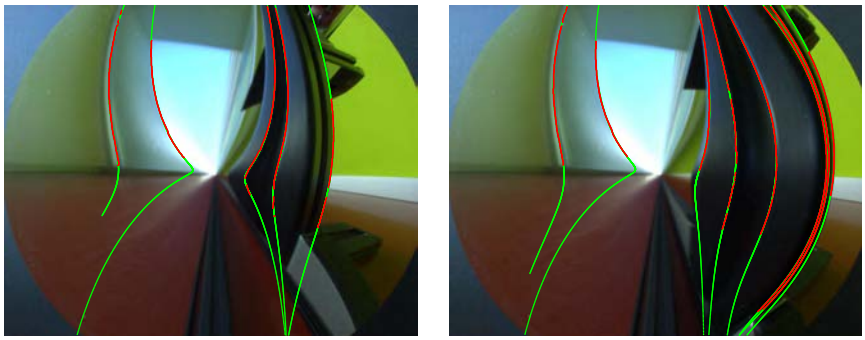
(a) (b)

Conical Catadioptric System

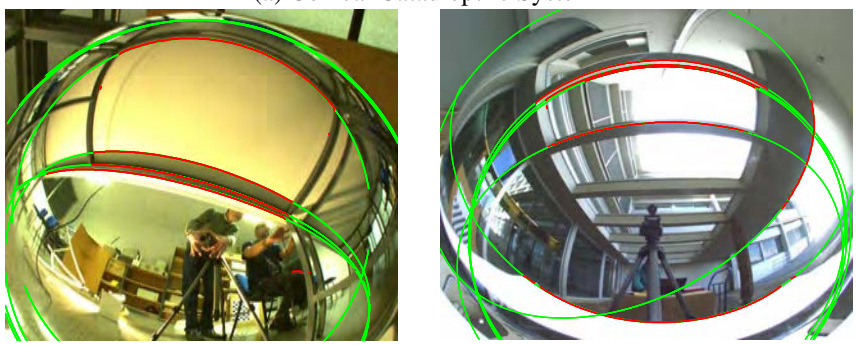
(c) (d)

Spherical Catadioptric System

Figure 4.13: Line-extraction examples from synthetic images. A selection of line-images correctly extracted are shown. The extracted line-images are shown in green and the supporting points on red. (a-b) Conical catadioptric system. (c-d) Spherical catadioptric system.



(a) Conical Catadioptric System



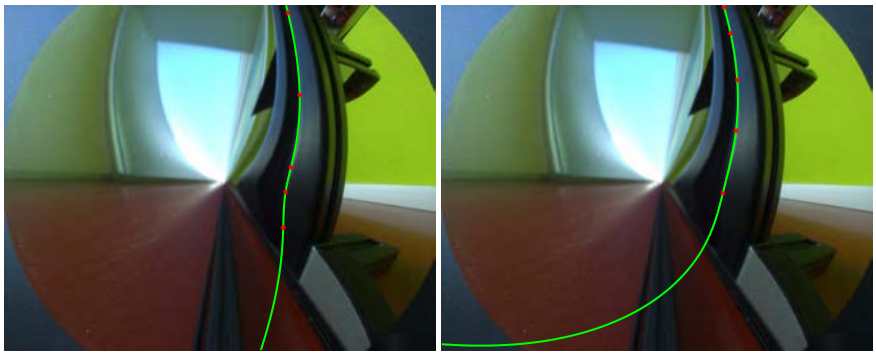
(b) Spherical Catadioptric System

Figure 4.14: Line-extraction examples from real images. A selection of line-images correctly extracted are shown. The extracted line-images are shown in green and the supporting points on red. (a) Conical catadioptric system. (b) Spherical catadioptric system.

office has been modified to look like a corridor. The synthetic images have a resolution of 1024×1024 pixels, The conical mirror has a radius $R_{max} = 30\ mm$ and height $h_{max} = 21.01\ mm$ i.e. of $\tau = 55\ deg$. The distance from the mirror to the camera is $Z_m = 1\ m$. The spherical mirror has a radius of $R_s = 1.25\ m$ and the distance between the center of the sphere and the camera is $Z_s = 2m$. In Fig. 4.13 we show some examples of well extracted lines from synthetic catadioptric images. The extracted line-images are shown in green and the supporting points on red.

Real Images

The real images have been acquired using two catadioptric systems composed by a conventional camera (uEye UI-148xSE-C) and two mirrors, one conic and other spheric. The conventional camera has a resolution of 1280×1024 pixels and has been inde-



(a) (b)

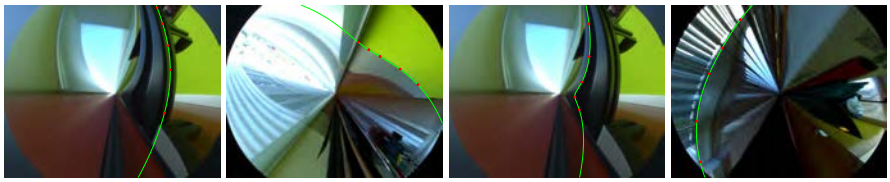
Figure 4.15: Influence of points selection. (a) Good fitting (b) Bad fitting. We can see the high influence of detected points in line extraction.

pendently calibrated using a standard method. Parameters taken into account are focal distance, principal point, skew and radial distortion. The conical mirror has an aperture angle of $\tau = 55\ deg$ and the system has been fixed manually to assure the alignment between the camera and the mirror having symmetry of revolution. The spherical mirror has a radius of $R_s = 0.37\ m$. As a spherical catadioptric system is always axial we do not need to enforce the alignment between camera and mirror but we need to calibrate the rotation of the camera with respect to the axis of revolution. The calibration of the system parameters Z_{Rel} and the rotation matrix is estimated minimizing the forward projection error of a chess pattern of known dimensions. In Fig. 4.14 we show some examples of the line-image extraction for different real images. The extracted line-images are shown in green and the supporting points on red.

4.8.3 Estimation of geometry of the conical mirror

In this section we show some examples of fitting using five points manually selected and we present quantitative results in the estimation of the mirror geometry from lines in real images. Five points from conical catadioptric images (in red) are selected manually to compute the line-image ω who is painted on the image using the parametric description (4.28). From each ω line-image we extract the Plücker coordinates of the line and the aperture angle of the mirror τ. As explained in Section 4.5.1 the distance to the mirror Z_m is coupled with the Plücker coordinates therefore the metric in recovered 3D lines are scaled to this distance.

In Figure 4.15 we show the high influence of error and point selection. Depending on the selected points the line-image fits or not the projected points of the line. Despite both line-images are fitting the defining points and the rest of the projected point of the segment, the error in the estimation of the 4DOFs complicates the right extraction of



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

Figure 4.16: Examples of correct fitting. (a) $\tau = 55.0\ deg$. (b) $\tau = 55.5\ deg$. (c) $\tau = 56.1\ deg$. (d) $\tau = 54.9\ deg$.

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

Figure 4.17: Examples of incorrect fitting. (a) $\tau = 50.9\ deg$. (b) $\tau = 51.1\ deg$. (c) $\tau = 49.7\ deg$. (d) $\tau = 48.4\ deg$.

the line. In Figure 4.16 we show some examples of line projections correctly fitted in different images and the obtained value for τ in each one. We can see how this value is close to the ground truth which is $\tau_{ref} = 55\ deg$. Finally, in Figure 4.17 we show more examples of incorrect fittings, due to the sensitivity of line projection and the large number of degrees of freedom of the curve in non-central systems.

4.9 Summary

In this Chapter we have analyzed the geometry of line projections in non-central catadioptric systems focusing on conical and spherical catadioptric cameras. In both cases line-images are quartic polynomials although in the conical case it can be simplified to a compact description of 6 homogeneous parameters. The underlying structure of line-images has been used for designing a line-image extraction method and to obtain the geometry of the mirror in the case of the conical catadioptric camera. In non-central cameras, it is possible to recover the complete geometry of the original projected line from a single projection. In this context, our proposal has promising applications in robotics (e.g., robot pose estimation and SLAM) and scene modeling (e.g., 3D reconstruction). From the results it follows that despite the line-image extraction can be correctly achieved, the lack of effective baseline of non-central catadioptric sys-

tems encourage to investigate new types of non-central cameras allowing accurate 3D reconstructions from single projections. Another feasible approach is exploiting the redundancy along a sequence of images for improving the 3D reconstruction accuracy.

Chapter 5

Line reconstruction in non-central circular panoramas

As presented in previous chapters, in certain non-central imaging systems, it is possible to recover the geometry of a 3D line from a single line projection. However, with classical non-central catadioptric systems there is not enough effective baseline for a practical implementation of this approach. In this chapter we propose a multi-camera system configuration resembling the circular panoramic model which results in a particular non-central projection allowing the stitching of a non-central panorama. From a single panorama we obtain well-conditioned 3D reconstruction of lines, which are especially interesting in texture-less scenarios. No previous information about the direction or arrangement of the lines in the scene is assumed. The proposed method is evaluated on both synthetic and real images.

5.1 Introduction

In central imaging systems any projection ray intersects a single point usually known as optical center. In the last decade, the need of a wider field of view increased the development of new imaging systems known as omnidirectional systems allowing complete panoramic images. For a better understanding of the images most of these systems were sought to be central (e.g. the paracatadioptric system, the hypercatadioptric system [Baker and Nayar, 2001], multicamera systems with common optical center [Kim et al., 2008]). The advantage of central systems is that they also allow using previous standard algorithms. However, some of the proposed omnidirectional systems were non-central by construction (e.g. conical catadioptric systems, spherical catadioptric systems). In these systems, rays do not intersect in a single optical center, hence they

can not be defined by a direction vector and require richer descriptions like the generalized camera model based on Plücker lines [Lee et al., 2015]. Although the higher complexity is a disadvantage, the properties of line projections in non-central systems can turn this disadvantage into an advantage.

The projection surface of a 3D line in central systems is a plane. All lines lying in this plane share the same line-image because only two of the four degrees of freedom (DOF) of the 3D line are preserved in the projection. Hence, the only way to recover the direction of the lines and depth up to scale from a single view is imposing additional constraints like dominant directions or perpendicularity [Ramalingam and Brand, 2013]. By contrast, in non-central cameras the projection surface of a line encapsulates the four degrees of freedom of the 3D line. Hence, the geometry of a 3D line can be recovered from a single projection. In particular, four points on a line-projection (except some degenerate cases) define four projecting rays providing four independent constraints for computing the complete geometry of the line [Teller and Hohmeyer, 1999]. Unfortunately, this approach is difficult to implement in practice. The quality of the extracted 3D line depends on a magnitude we call the effective baseline of a set of rays which is related with the distances between the four defining skew rays. This effective baseline is too small when using non-central catadioptric systems, so for exploiting this property in practice the system has to be too big.

In this chapter, we rediscover the circular panoramic imaging system as a non-central system [Menem and Pajdla, 2004] which has enough effective baseline for 3D line reconstruction. However, a moving camera introduces constructive and synchronization problems. Therefore, we consider a system for taking a panorama following the circular panoramic projection model by using a rig of cameras. The motivations of using line-features in non-central systems are numerous. As in central systems, line-features are useful for reconstructing texture-less scenarios. Besides, in non-central systems metric reconstruction of 3D lines can be recovered from a single panoramic view and no assumption about the direction or the arrange of the 3D lines is needed. In addition, considering sequences of images, long lines can be seen along large fragments of the sequence reducing the drift in 3D estimation. As drawback the extraction of the line projection in non-central systems is still an open topic.

The geometry of non-central systems has previously been studied for computer vision applications. The back projection of different non-central catadioptric systems is introduced in [Swaminathan et al., 2006]. Contrary to the back projection case, the forward projection of non-central systems does not necessarily have a closed form solution. In [Agrawal et al., 2010] Agrawal et al. present a polynomial solution for the forward projection in axial non-central catadioptric systems based on a quadric mirror which is extended for the non axial case in [Agrawal et al., 2011]. These approaches are used in a calibration method for multi-axial non-central systems in [Agrawal and Ramalingam, 2013]. In [López-Nicolás and Sagüés, 2014] the epipolar geometry of conical catadioptric systems and its corresponding calibration are presented.

Another way to obtain non-central images are images generated from moving cam-

eras without common viewpoint. The epipolar geometry is studied for the case of linear pushbroom camera in [Gupta and Hartley, 1997] and for the case of non-central circular panorama in [Menem and Pajdla, 2004]. A multi-camera system can be also tackled as particular case of a non-central system. In [Lee et al., 2015] Lee et al. propose a new method to solve the pose estimation in a multi-camera system represented by a set of ray bundles in a non-central description using Plücker coordinates.

As previously said, in a non-central image the geometry of 3D line can be recovered from a single line projection. The fundamentals of this approach are exposed in [Teller and Hohmeyer, 1999, Kneebone et al., 1998, Griffiths and Harris, 2011] where it is shown that two lines are the intersection of four generic lines. In [Caglioti and Gasparini, 2005, Caglioti et al., 2007a] this approach is exploited to compute 3D lines from 4 rays in non-central catadioptric systems. Different computation methods are considered and the degeneracies and singular configurations are studied.In [Gasparini and Caglioti, 2011] these works are compiled and the case of circular panorama is also addressed. In [Lanman et al., 2006] the approach is used with spherical catadioptric mirrors, and in addition two non-central systems are used for reconstruction. Work in [Swaminathan et al., 2008] extends the approach to planar curves. In [Bermudez-Cameo et al., 2014b] the approach is derived to the case of conical catadioptric systems obtaining both the 3D line and the mirror geometry. To improve the accuracy in reconstruction using catadioptric systems some approaches have been proposed: considering only horizontal lines [Pinciroli et al., 2005, Chen et al., 2011] , exploiting cross-ratio properties [Perdigoto and Araujo, 2012]. Using off-axis systems [Caglioti et al., 2007b] allows avoiding the degeneracies caused by the revolution symmetry. As application, the pose of non-central catadioptric systems is estimated in an image sequence [Miraldo and Araujo, 2014, Miraldo et al., 2015] using known 3D lines.

In this chapter, we propose the circular panorama imaging system and a multi-camera configuration as a way to have enough effective baseline for a practical application of 3D line reconstruction from a single non-central line projection. We present a new method for automatic line extraction valid for the proposed non-central system. The result is a metric reconstruction of lines-based scenarios from a single panorama without making assumptions about the direction or the arrangement of the lines. The non central circular panoramic system is compared with other catadioptric non-central systems to evaluate the proposed system advantages. We also propose a multi-camera architecture for obtaining an approximation of a circular panorama without the disadvantages of a moving camera. The extraction method is then particularized to the multi-camera case. The methods are tested with realistic simulated scenarios and real images.

5.2 Non-central circular panorama model

In this section, we describe the projection model of non-central circular panoramas. A circular imaging panoramic projection [Menem and Pajdla, 2004, Gasparini and

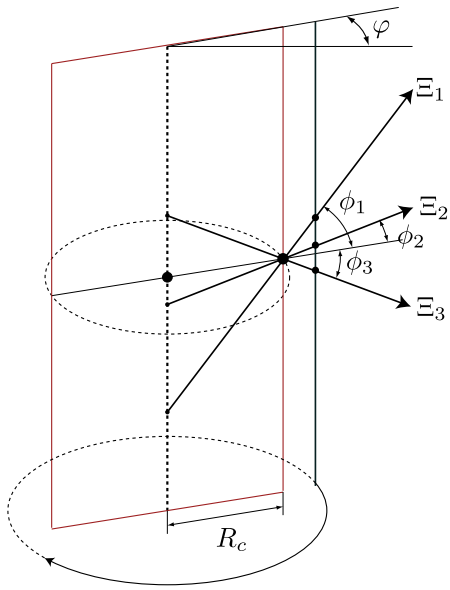


Figure 5.1: Non-central circular imaging projection system. Projecting rays intersect a circle of radius R_c and the vertical axis. Three rays (Ξ_1,Ξ_1,Ξ_1) are shown as example.

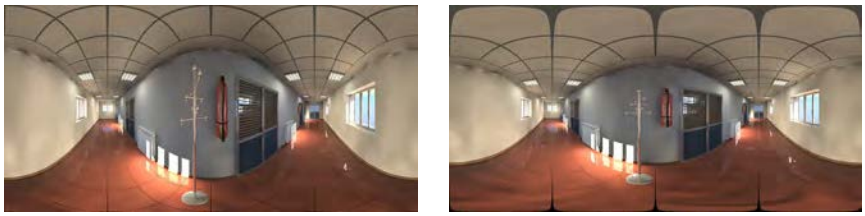
Caglioti, 2011] is a projection model with symmetry of revolution in which any projecting ray intersects both an axis of revolution and a circle of a given radius R_c (see Fig. 5.1). A camera configuration fulfilling this constraint is a linear sensor turning around an axis of revolution. The resulting image of this system is a panoramic image in which the axis of coordinates are a scaled representation of two angles ϕ, φ which are the spherical coordinates of the direction vector of each projecting ray (see Fig. 5.2). The forward and back projection model of a circular panoramic system have analytical solution and its computation is not time consuming.

5.2.1 Back projection model

Consider the angles ϕ and φ, which are related with the image coordinates (j,i) as follows,

$$\varphi = j\frac{\varphi_{end}-\varphi_{ini}}{n_{columns}}+\varphi_{ini}, \quad \phi = i\frac{\phi_{end}-\phi_{ini}}{m_{rows}}+\phi_{ini} \tag{5.1}$$

where $(m_{rows},n_{columns})$ are the dimensions of the panorama and $(\varphi_{ini},\varphi_{end})$ and (ϕ_{ini},ϕ_{end}) are the limits of the field of view of the imaging system.



(a) (b)

Figure 5.2: (a) Circular panoramic image with $R_c = 0.5m$ (Non-central). (b) Circular panoramic image with $R_c = 0m$ (A central spherical panoramic image).

The back projection model can be described by the definition of each projecting ray with Plücker coordinates :

$$\Xi = \begin{pmatrix} \xi \\ \bar{\xi} \end{pmatrix} = \begin{pmatrix} \cos\phi\cos\varphi \\ \cos\phi\sin\varphi \\ \sin\phi \\ R_c\sin\phi\sin\varphi \\ -R_c\sin\phi\cos\varphi \\ 0 \end{pmatrix} \tag{5.2}$$

5.2.2 Forward projection model

Given a 3D point in homogeneous coordinates $\mathbf{X} = (x_1, x_2, x_3, x_0)^\mathsf{T}$ the forward projection is computed as follows

$$\varphi = atan2(x_2, x_1), \quad \phi = \arctan\frac{x_3}{\sqrt{x_1^2 + x_2^2} - x_0 R_c} \tag{5.3}$$

$$j = n_{columns}\frac{\varphi - \varphi_{ini}}{\varphi_{end} - \varphi_{ini}}, \quad i = m_{rows}\frac{\phi - \phi_{ini}}{\phi_{end} - \phi_{ini}} \tag{5.4}$$

5.3 Line estimation in non-central systems

First we present the procedure for 3D line estimation in non-central systems. Secondly we present the expression for the line-image in non-central circular panoramas. Using Plücker coordinates, the intersection between lines is described by the side operator (we use the notation used in [Pottmann and Wallner, 2001]). A given line $\mathbf{L} = \left(\mathbf{l}^\mathsf{T}, \bar{\mathbf{l}}^\mathsf{T}\right)^\mathsf{T} \in \mathbb{P}^5$ (where $\mathbf{l} \in \mathbb{R}^3$ and $\bar{\mathbf{l}} \in \mathbb{R}^3$) intersects a ray $\Xi \in \mathbb{P}^5$ if

$$side\,(\mathbf{L},\Xi) = \mathbf{L}^{\mathsf{T}}\mathsf{W}\Xi = \mathbf{l}^{\mathsf{T}}\bar{\xi} + \bar{\mathbf{l}}^{\mathsf{T}}\xi = 0 \tag{5.5}$$

where $\mathsf{W} = \begin{pmatrix} 0_{3\times3} & I_{3\times3} \\ I_{3\times3} & 0_{3\times3} \end{pmatrix}$.

As a 3D line has four degrees of freedom we need at least 4 equations to solve for $\mathbf{L}$. If four projection rays from a 3D line provide four independent constraints we can compute the 3D line from the system of equations

$$\mathsf{A}\mathbf{L} = \mathbf{0} \tag{5.6}$$

where $\mathsf{A}_i = \begin{pmatrix} \bar{\xi}_i & \xi_i \end{pmatrix}$ is the A_i row of the matrix A. Except from some degenerate cases in non-central systems the projection surface of a line is not a plane and four generic rays induce two incident lines [Teller and Hohmeyer, 1999].

As Plücker coordinates is an over-parametrized representation of a line in $\mathbb{P}^5$ the solution of the equation system (5.6) is a one-dimensional subspace of $\mathbb{P}^5$: $\mathbf{L} = \mathbf{L}_0 + \mathbf{L}_1\lambda$ described by two elements of $\mathbb{P}^5$, $\mathbf{L}_0$ and $\mathbf{L}_1$ which can be computed using a Singular Value Decomposition. Imposing the Plücker identity we obtain the intersection with the Klein Quadric which results in two solutions. One of them is the sought 3D line. If the system is axial the other solutions is the axis of revolution.

In some degenerate cases four rays do not provide four independent equations [Gasparini and Caglioti, 2011]: when the projection surface is a plane (the line is coplanar with the axis of revolution or coplanar with the plane containing the circle) or when the projection surface is a regulus.

5.3.1 Line-Image in non-central circular panorama

Deriving (5.5) for circular panoramic systems the homogeneous expression for the line-image on the panoramic image becomes

$$\cos\phi\left(\bar{l}_1\cos\varphi + \bar{l}_2\sin\varphi\right) + \sin\phi\left(1 + R_c\left(l_1\sin\varphi + l_2\cos\varphi\right)\right) = 0 \tag{5.7}$$

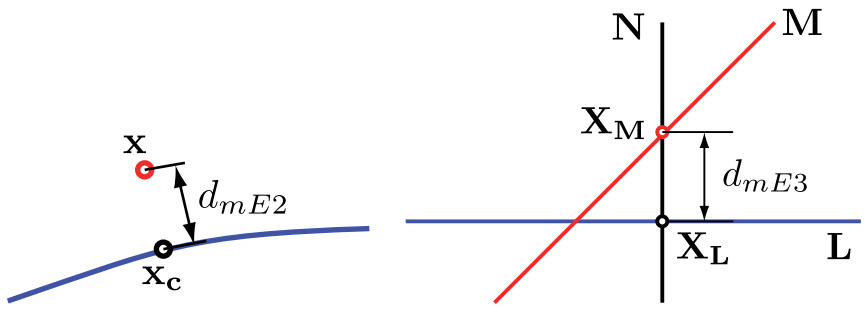
which allows a parametric representation expressing ϕ in terms of φ.

$$\tan\phi = \frac{-\left(\bar{l}_1\cos\varphi + \bar{l}_2\sin\varphi\right)}{1 + R_c\left(l_1\sin\varphi + l_2\cos\varphi\right)} \tag{5.8}$$

Notice that as we know the location of the 3D line we can also compute their vanishing points by projecting the direction on the image.

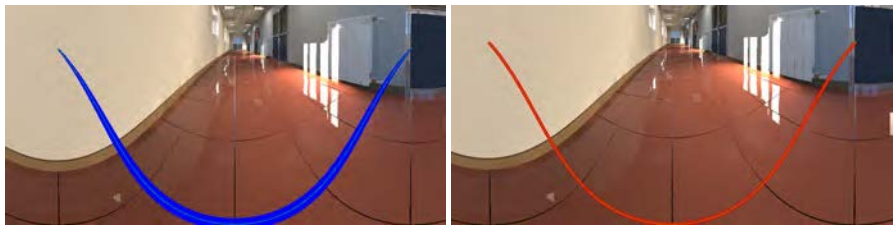
5.4 Distance from points to line-images

In this section we propose a distance measurement to evaluate when an image point belongs to a line-image. It is also used for improving the line estimation in an optimization scheme. Depending on the used distance we will reach different results.



(a) (b)

Figure 5.3: Distance to a line-image. (a) Distance to a line projection on the image. (b) Distance between a ray **M** and a 3D line **L**.



(a) (b)

Figure 5.4: Region around a line projection defined by a given distance. (a) Metric distance $\mathbb{E}^3$ from ray to line. (b) Our proposal.

The Euclidean metric distance in $\mathbb{E}^3$ is the minimum Euclidean distance between two 3D lines (the 3D line and the projecting ray generated from an image point). This distance is defined as in [Pottmann and Wallner, 2001] :

$$d_{mE3} = \frac{|\mathbf{l}_i{}^{\mathsf{T}}\overline{\mathbf{l}}_j + \mathbf{l}_j{}^{\mathsf{T}}\overline{\mathbf{l}}_i|}{\|\mathbf{l}_i \times \mathbf{l}_j\|} \ . \tag{5.9}$$

When determining if a point lies on a line-image in a RANSAC or in a Hough transform scheme this distance tends to give higher reward to lines which are close to the origin. This effect induces errors in the extraction process.

To avoid this effect the metric distance $\mathbb{E}^2$ from point to line-image can be used. This is performed by using the method of Lagrange multipliers to obtain the point of the line-image $\mathbf{x}_c$ which is closer to the given point $\mathbf{x}$ and computing the Euclidean distance between them (see Fig. 5.3(a)). However, except from some particular cases [Bermudez-Cameo et al., 2014b], this approach does not have a closed form solution and needs to be solved iteratively. The alternative we propose is to compute the points

on the 3D lines ($\mathbf{X}_L$ and $\mathbf{X}_M$) which have the minimum metric distance d_{mE3} in $\mathbb{E}^3$ (see Fig. 5.3 b). Instead of using distance in $\mathbb{E}^3$ we propose to project both points and compute the metric $\mathbb{E}^2$ distance on the image. The advantage is that this computation has a closed form and can be used in any non-central system in which the forward projection has a closed solution. Notice that the projection of the closest point in the 3D space $\mathbf{X}_M$ is not exactly the same as the closest point $\mathbf{x}_c$ on the image.

To compute these closest points we have to compute a third line passing through both lines and with direction orthogonal to the directions of both 3D lines (see Fig. 5.3 b). Given a line $\mathbf{L} = \left(\mathbf{l}^\mathsf{T}, \bar{\mathbf{l}}^\mathsf{T}\right)^\mathsf{T}$ and other line $\mathbf{M} = \left(\mathbf{m}^\mathsf{T}, \bar{\mathbf{m}}^\mathsf{T}\right)^\mathsf{T}$ representing the projection ray of a given point on the image we want to compute a third line $\mathbf{N} = \left(\mathbf{n}^\mathsf{T}, \bar{\mathbf{n}}^\mathsf{T}\right)^\mathsf{T}$ which intersects both lines and is orthogonal to $\mathbf{L}$ and $\mathbf{M}$. To enforce the orthogonal constraint we define $\mathbf{n} = \mathbf{l} \times \mathbf{m}$.

The intersections are imposed using the side operator. Finally we obtain a third equation by imposing $\mathbf{N}$ to be a line using the Plücker constraint. The result is a system of three linear equations

$$\bar{\mathbf{l}}^\mathsf{T}(\mathbf{l} \times \mathbf{m}) + \mathbf{l}^\mathsf{T}\bar{\mathbf{n}} = 0 \tag{5.10}$$

$$\bar{\mathbf{m}}^\mathsf{T}(\mathbf{l} \times \mathbf{m}) + \mathbf{m}^\mathsf{T}\bar{\mathbf{n}} = 0 \tag{5.11}$$

$$\bar{\mathbf{n}}^\mathsf{T}(\mathbf{l} \times \mathbf{m}) = 0 \tag{5.12}$$

which is solved for $\bar{\mathbf{n}}$ obtaining

$$\bar{\mathbf{n}} = \begin{pmatrix} \mathbf{l}^\mathsf{T} \\ \mathbf{m}^\mathsf{T} \\ (\mathbf{l} \times \mathbf{m})^\mathsf{T} \end{pmatrix}^{-1} \begin{pmatrix} -\bar{\mathbf{l}}^\mathsf{T}(\mathbf{l} \times \mathbf{m}) \\ -\bar{\mathbf{m}}^\mathsf{T}(\mathbf{l} \times \mathbf{m}) \\ 0 \end{pmatrix} \tag{5.13}$$

Once we have computed the line $\mathbf{N}$, we can compute the intersection between $\mathbf{N}$ and $\mathbf{L}$ by using the meet operator obtaining the points

$$\mathbf{X}_L = \begin{pmatrix} -\bar{\mathbf{n}} \times \bar{\mathbf{l}} \\ \mathbf{n}^\mathsf{T}\bar{\mathbf{l}} \end{pmatrix}, \quad \mathbf{X}_M = \begin{pmatrix} -\bar{\mathbf{n}} \times \bar{\mathbf{m}} \\ \mathbf{n}^\mathsf{T}\bar{\mathbf{m}} \end{pmatrix} \tag{5.14}$$

that are projected on the image using the forward projection model. Then, the metric distance $\mathbb{E}^2$ between the projections is computed. In Fig. 5.4 we compare the region defined by a given distance using both approaches.

5.5 Non-central circular panoramas from multi-camera system

In this section, we consider to model a multi-camera system as a non-central circular panoramic imaging system and discuss their similarities and differences. We also

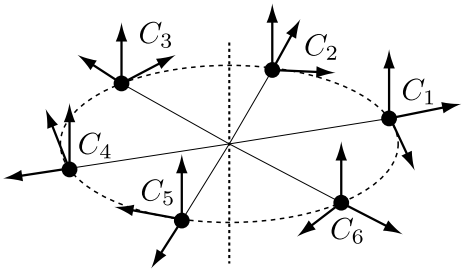


Figure 5.5: Multi-camera system with 6 cameras arranged in a circle.

compare and evaluate the accuracy of different non-central systems in 3D line reconstruction.

A non-central circular panorama can be obtained using an off-centred rotating camera where only the central column of the image is used to stitch the panoramic image. However, this set-up has the synchronization difficulties of a moving camera especially if the whole system is in motion. Instead, it is possible to approximate this configuration with an array of cameras arranged in a circle (see Fig. 5.5). From this configuration we can represent the azimuth and elevation angle of each projecting ray considering the non overlapped vertical columns.

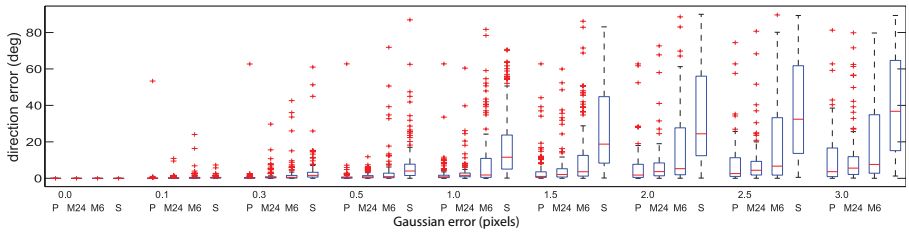
Non-central circular panorama and the proposed multi-camera panorama are equivalent in the columns corresponding with the azimuth of the cameras focal axis. The rest of the panorama is equivalent if the scene is at infinity. From a practical point of view, only the small part of the center columns of the image is extracted for compositing the panorama hence the needed bandwidth reduces considerably. The multi-camera panorama is finally stitched using alpha blending.

5.5.1 Multi-camera vs. circular panorama

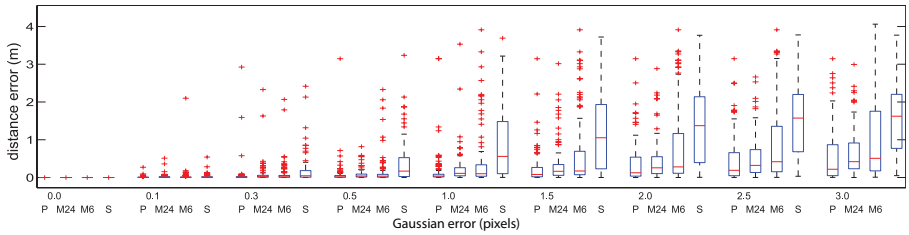
The whole system is a non-central system because the projecting rays do not intersect in a single view-point. However, the proposed multi-camera system is locally central and therefore rays from a single camera can only provide two independent equations in the line fitting. This local centrality must be taken into account when getting random points lying on a line projection on the panoramic image.

Consider a random picking of four rays from the panoramic image. Three cases correspond to valid configurations: each ray is picked from a different camera ([1+1+1+1]), two rays are from the same camera and the two other rays are from two different cameras ([2+1+1]), lastly two rays are selected from two different cameras ([2+2]). The other two cases correspond to non-valid configurations: for rays from a single camera ([4]) and three rays from a camera and one from other ([3+1]).

There is another difference between the multi-camera system and the circular pan-



(a)



(b)

Figure 5.6: Comparison in accuracy reconstructing lines using different non-central systems. (a) Direction error in degrees. (b) Distance error in meters. P for circular panoramic system. M6 for a multi-camera having 6 single fisheye cameras. M24 for a multi-camera having 24 single fisheye cameras. S for spherical catadioptric system.

oramic imaging system. Notice that, the circular panoramic system is an axial system meaning that all the projecting rays intersect the same rotation axis. When computing a 3D line from 4 rays we obtain two solutions, one of them is the sought line and the other is the axis of revolution. However, a multi-camera system is not axial and therefore there is not a single line intersecting the projecting rays which would help to discriminate the correct solution. For the case in which we take 2 rays from two cameras one of the solutions must intersect the line defined by the two optical centres, therefore it must lie in the horizontal plane passing through all the optical centres. In other case, we have to use an additional criterion to distinguish between both solutions. In particular, we include both solutions as hypothesis in the RANSAC scheme. Only the correct solution has supporting points on the image, therefore wrong solutions are directly rejected in the extraction process.

5.5.2 Accuracy of 3D lines in non-central systems

In this section we evaluate the accuracy of the proposed system for single-view 3D line extraction in comparison to non-central catadioptric systems. Hence, we assume

the points of the image belong to a given line projection. The geometric limitations of reconstructing 3D lines from a single line projection in non-central systems are given by the relation between the size of the system and the depth of the line (assuming the line is long enough). That means that in most known non-central systems this limitation prevents the practical using of this method because the estimated 3D lines are too noisy. However, it depends also on the system type. In this analysis we are considering random lines with a depth up to 3 meters. The length of the lines is 20m. Four different systems are compared:

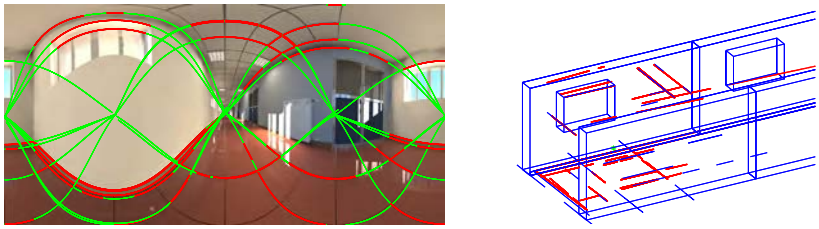
- A spherical catadioptric system composed of a spherical mirror with a radius of 1.2 m and a perspective camera located at 1.8 m from the center of the sphere. The resolution of the camera is 4096×4096 pixels.
- A circular panoramic imaging system defined by a circle of a radius of 0.3 m. The image has a resolution of 4000×8000 pixels.
- Two cases of the proposed multicamera system. The first one is composed of 6 fisheye cameras located in a circle of a radius 0.3m. The second one is composed of 24 fisheye cameras in a circle of radius 0.3 m. In both cases the cameras have a resolution of 768×1024 pixels.

Once the points of the random lines are projected on the corresponding images, we add Gaussian noise of a given σ. We variate the value of σ from 0 to 3 pixels. In Fig. 5.6 we show the results of the comparison. Fig. 5.6 (a) depicts the distribution of the direction error (degrees) of the estimated 3D lines with respect to the ground truth. Fig. 5.6 contains the distribution of distance error. Both the circular panoramic system and multi-cameras clearly outperforms in terms of accuracy the classical catadioptric non-central systems in spite of having smaller sizes (0.3 m vs 1.2 m).

5.6 Line-extraction in non-central panoramas

The extraction of the line projection in non-central systems is still an open topic. In this section we present our proposal for extracting lines in the non-central circular panorama which is also used in the multi-camera panoramas obtained from real images.

To reduce textured patterns but conserving the edges first the panoramic image is preprocessed using a Rolling Guidance Filter [Zhang et al., 2014]. Then, the edges of the image are detected using Canny detector. After that, we propose a split and merge approach. First, edges are stored in connected components. To reduce the complexity of the line extraction we first estimate locally central approximations using a classical RANSAC approach with two degrees of freedom (see 5.7 (a)). For this, a greedy multi-model RANSAC is executed on each connected component obtaining a set of central segments. In the multi-camera approximation the resulting central segments are constrained in each camera region. The extreme points of each segment are



(a) (b)

Figure 5.7: Line extraction in non-central circular panorama. (a) Examples of extracted line-images in green and supporting points in red. (b) 3D reconstruction (in red) obtained from the back projection of the supporting points of the line-image to the extracted 3D line.

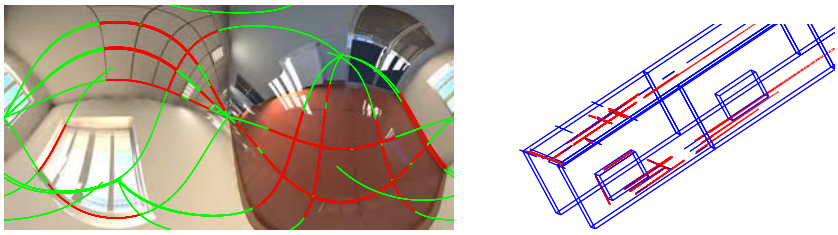
used to obtain the connectivity of the segments. For this, we back-project the points to the torus defined by the projection model. The connectivity map is obtained composing a set of local Delaunay triangulations computed on the tangent map of the manifold. Then, the connectivity of the central segments are exploited to generate a collection of hypothesis in a USAC framework [Raguram et al., 2013]. A set of different combinations of three segments are randomly generated. All the combinations not satisfying the connectivity are removed. Then, the remaining hypothesis are evaluated. Each hypothesis is tested using the distance presented in Section 5.4 and taking into account the effective baseline of each set of rays. To measure the effective baseline of a set of n rays we use the function z_{bs} presented in Chapter 4 which depends on the metric distances d_{mE3} between the defining rays (unless degenerate cases projecting rays are skew in circular panoramic systems)

$$z_{bs} = \frac{n!}{2\,(n-2)!}\left(\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\frac{1}{d_{mE3}\,(\Xi_i,\Xi_j)}\right)^{-1} \tag{5.15}$$

so that this function increases when the distances between rays increase but also assures that not any individual distance d_{mE3} turns to zero. The results of the extraction method are the 3D lines and the corresponding edges supporting the line-images that are directly related with the 3D points on the 3D line (see Fig. 5.7).

5.7 Experimental evaluation

To evaluate the proposed extraction method we have performed experiments with synthetic and real images. Synthetic images have been generated using a spin-off version of Pov-Ray called Mega-Pov which allows to define the non-central circular panoramic camera as a parametric camera. The scenario is a modification of a publicly available



(a) (b)

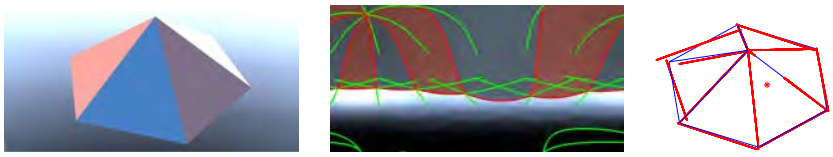
Figure 5.8: Line extraction in non-central circular panorama. (a) Examples of extracted line-images in green and supporting points in red. (b) Obtained 3D reconstruction (in red).

Scenario	angular error (degs)		depth error (m)	
	median	σ	median	σ
Horizontal	0.948	1.180	0.028	0.036
Slanted	0.954	1.105	0.091	0.126

Table 5.1: Error distribution in line-image extraction.

synthetic scenario [1]. The synthetic panoramas have a resolutions of 4096×2048 pixels and the radius of the generation circle is $R_c = 0.5m$. In Fig. 5.7 (a) we show some examples of extracted lines from synthetic panoramas. The extracted line-images are depicted in green and the edge points supporting the line-image in red. The corresponding 3D reconstruction is shown in Fig. 5.7 (b) where the ground truth is coloured in blue and the reconstructed segments are depicted in red. One of the degenerate cases is the case in which the line lies in a plane passing through the axis of revolution of the system (the projected line-image is an straight 2D line). The further a line is from this degeneration the best conditioned is. Because that horizontal lines used to be better conditioned than slanted lines. In Fig. 5.8 we show a similar example having slanted lines after rotating the camera 50 degrees. In Table. 5.1 we show a comparison between the extraction error in both scenarios (horizontal and slanted). Given the error array e_i the standard deviation σ is robustly estimated using the median absolute deviation ($\sigma = 1.4286\ \mathsf{median}\,|e_i - \mathsf{median}\,(e_i)|$). Finally, just to illustrate that we are not imposing or using Manhattan conditions or dominant directions in Fig. 5.9 we present a toy example of reconstruction of a non-Manhattan scenario. In this particular example the defining points of the line-images have been manually selected.

[1] http://hof.povray.org/office-13.html The Office - Jaime Vives Piqueres, 2004



(a) (b) (c)

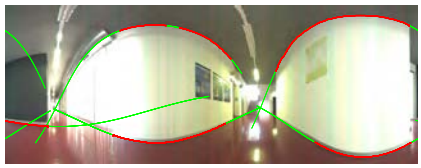
Figure 5.9: Reconstruction of a non-Manhatan scene from a non-central circular panorama. (a) Perspective view of the object. (b) Line-image extraction on panorama. (c) Obtained 3D reconstruction (in red). The marker denotes the camera location.

5.7.1 Experiments with real images

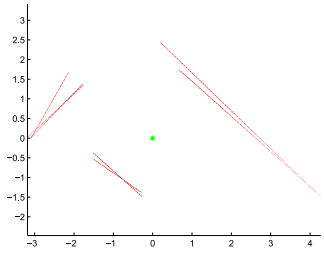
The real images have been composed using a rotating off-axis fisheye camera (uEye UI-148xSE-C with lens Lensagon CF5M1414) with a radius of $R_c = 0.5m$. The panoramic images have a size of 1570×4096 pixels and the original images have a resolution of 1920×2560 pixels. In Fig.5.10 we show an and example of the extraction in a multi-camera panorama generated from 180 cameras. Fig.5.11 shows an extraction from a multi-camera panorama stitched from 20 cameras. For evaluating the reconstruction we have taken a measure of the environment. The width of the corridor measured with a laser electronic distance meter (EDM) is 3.20 m. From the circular panorama shown in Fig. 5.10 we obtain a measure of 3.22m.

5.8 Summary

In this chapter, we propose a non-central system which allows 3D reconstruction of lines from single panoramas without assumptions (such a Manhattan world) about the direction or arrangement of the lines. We compare our configuration with other known non-central cameras concluding that (in contrast with other non-central systems) our configuration has enough effective baseline to perform a 3D reconstruction of a real scenario with normal dimensions. This result opens new possibilities for performing real applications. We also present a new method for automatic extraction of line-images from non-central circular panoramic images. The proposal has been tested with realistic synthetic images and with real images.

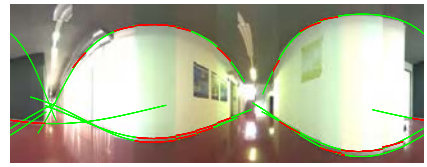


(a)

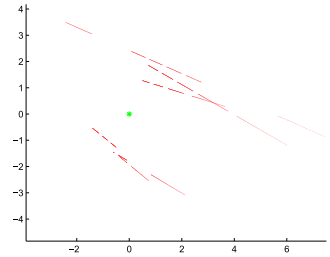


(b)

Figure 5.10: Line extraction in panoramic images from 180 cameras. (a) Examples of extracted line-images in green and supporting points in red. (b) Obtained 3D reconstruction (in red).



(a)



(b)

Figure 5.11: Line extraction in panoramic images from 20 cameras. (a) Examples of extracted line-images in green and supporting points in red. (b) Obtained 3D reconstruction (in red).

Chapter 6

Minimal solution for pairs of lines in non-central cameras

In previous chapters we have dealt with line-images in non-central systems and how to recover the 3D information of the line from a single projection without assumptions. As presented before, the solution of this problem is ill-conditioned due to the lack of effective baseline between rays. This limitation is particularity evident in non-central catadioptric systems where a practical implementation of the approach is limited to lines in regions which are close to the imaging system. In this chapter, we exploit additional geometric constraints to improve the results of line reconstruction from single images in non-central systems. In particular, we obtain the minimal solution for the case of a pair of intersecting orthogonal lines and for the case of a pair of parallel lines considering three rays from each line. The proposal has been evaluated with simulations and tested with real images.

6.1 Introduction

In any central camera the projection surface of a 3D line is a plane. Any line contained in this plane is projected on the same line-image, and therefore two of the four degrees of freedom (DOF) of the 3D line are lost in the projection. By contrast, in certain non-central cameras (non-central implies that the projecting rays do not intersect in a common point) the projection surface of a line is a ruled surface composed by skew rays (except in certain degenerate cases). Through this surface there exist a unique mapping between a 3D line and its projection on the image (line-image) which can be exploited to recover the geometry of the 3D line from a single image. In particular,

four generic rays[1] corresponding to four points on a line projection provide four independent constraints allowing to compute the complete geometry of the 3D line [Teller and Hohmeyer, 1999]. In practice, this is an ill-posed problem and the geometry of the 3D line can be only recovered if the relative depth of the line with respect to the system dimensions is low enough for guaranteeing effective baseline between rays.

In this chapter we exploit the structure of the scene reducing the number of DOFs of the sought solution. In central systems with conventional cameras this idea is used for inferring layouts from line projections which only provide two independent constraints [Lee et al., 2009, Ramalingam and Brand, 2013] for each line-image. In the case of non-central systems we propose to solve the four DOFs of each line using the redundant independent constraints provided by line projections by imposing geometric constraints between pairs of lines. The proposed geometric constraints are: orthogonal intersection between lines and parallelism between lines reducing to six the number of rays needed for a minimal solution of a pair of lines.

The basis for computing the geometry of the 3D line from a single line projection (line-image) is that given four generic lines there exist only two lines intersecting them [Kneebone et al., 1998, Griffiths and Harris, 2011]. In [Teller and Hohmeyer, 1999] this reasoning is introduced for application in computer graphics. In [Caglioti and Gasparini, 2005, Gasparini and Caglioti, 2011] this approach is exploited to compute 3D lines from 4 rays in non-central systems comparing the linear approach with different computation methods and considering the degeneracies and singular configurations. In [Lanman et al., 2006] the approach is used with spherical catadioptric mirrors, and in addition two non-central systems are used for reconstruction. Work in [Swaminathan et al., 2008] extends the approach to planar curves. To improve the accuracy in reconstruction some simplifications have been proposed: considering only horizontal lines [Pinciroli et al., 2005, Chen et al., 2011] or exploiting cross-ratio properties [Perdigoto and Araujo, 2012]. In [Caglioti et al., 2007b] the degeneracies caused by the revolution symmetry are avoided using an off-axis system.

In this chapter, we also present the minimal solution for computing a junction composed by two orthogonal intersecting lines and for computing two parallel lines in non-central systems. This allows to obtain the complete geometry of these pairs of lines from three rays belonging to each line in a calibrated non-central system. The interest of this result can be in robust extraction methods based on minimal sets like RANSAC or in the improvement of the reconstruction accuracy in line fitting. The approach has been implemented for spherical catadioptric systems, using the projection model described in [Agrawal et al., 2010]. The proposal has been evaluated in simulation, and tested with real images.

[1] Four lines are generic if no two of them are coplanar, no three of them are coconical or cocylindrical, and the four are not cohyperbolic, i.e. do not lie on the same ruled quadric surface.

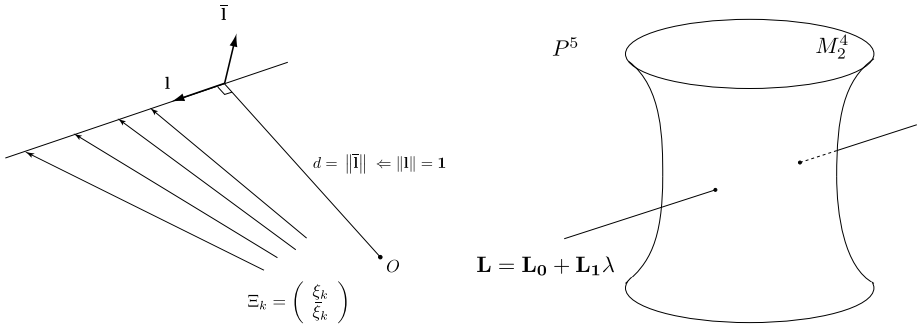


Figure 6.1: (a) Plücker description for lines and line projection. (b) One dimensional subspace in $\mathbb{P}^5$ and Klein Quadric.

6.2 Recovering a 3D line from four skew generic rays

In this section we present the background, geometric concepts and notation used for computing a 3D line from four generic rays. The description used for lines are the Plücker coordinates based on Grassmann algebra.

The Plücker coordinates of a 3D line is a $\mathbb{P}^5$ representation of a line obtained from the null space of any pair of points $X = \left(x_0, \mathbf{x}^\mathsf{T}\right)^\mathsf{T}$, and $Y = \left(y_0, \mathbf{y}^\mathsf{T}\right)^\mathsf{T}$ belonging to the line. The description of a Plücker line from the coordinates of the defining points are $\mathbf{L} = \left(\mathbf{l}^\mathsf{T}, \bar{\mathbf{l}}^\mathsf{T}\right)^\mathsf{T}$ where $\mathbf{l} \in \mathbb{R}^3$, $\mathbf{l} = x_0\mathbf{y} - y_0\mathbf{x}$ and $\bar{\mathbf{l}} \in \mathbb{R}^3$, $\bar{\mathbf{l}} = \mathbf{x} \times \mathbf{y}$ (depending on the author the order and the sign of the elements can differ, here we follow the standard in [Pottmann and Wallner, 2001]).

Notice that not all the elements of the $\mathbb{P}^5$ space correspond to 3D lines. The points of $\mathbb{P}^5$ corresponding to lines in $\mathbb{P}^3$ must hold $\mathbf{l}^\mathsf{T}\bar{\mathbf{l}} = 0$ which is known as Plücker identity. This identity is a quadratic constraint which defines a two dimensional subspace in $\mathbb{P}^5$ called the Klein quadric M_2^4 [Pottmann and Wallner, 2001]. Plücker representation has a geometric interpretation in Euclidean geometry. Vector $\mathbf{l}$ represents the direction of $\mathbf{L}$ (see Fig. 6.1 (a)) and $\bar{\mathbf{l}}$ is the moment vector which can be seen as the normal to a plane passing through the 3D line and the origin of the reference system. The Plücker identity expresses the orthogonality of $\mathbf{l}$ and $\bar{\mathbf{l}}$ given that the direction vector must be contained in the plane defined by $\bar{\mathbf{l}}$. $\mathbf{L}$ is an homogeneous vector but when normalizing respect to $\mathbf{l}$, $\left|\bar{\mathbf{l}}\right|$ is the minimum distance from the origin to the 3D line. Therefore, when normalizing respect to $\bar{\mathbf{l}}$, $|\mathbf{l}|$ is the inverse of that distance.

The constraint resulting of the intersection of a 3D line with a projection ray can be expressed linearly by the side operator which defines the signed distance between two lines $\mathbf{L} = \left(\mathbf{l}^\mathsf{T}, \bar{\mathbf{l}}^\mathsf{T}\right)^\mathsf{T}$ and $\mathbf{M} = \left(\mathbf{m}^\mathsf{T}, \bar{\mathbf{m}}^\mathsf{T}\right)^\mathsf{T}$. It is described as

$$side\,(\mathbf{L},\mathbf{M}) = \mathbf{L}^{\mathsf{T}}\mathsf{W}\mathbf{M} = \mathbf{l}^{\mathsf{T}}\bar{\mathbf{m}} + \bar{\mathbf{l}}^{\mathsf{T}}\mathbf{m} \tag{6.1}$$

where $\mathsf{W} = \begin{pmatrix} 0_{3\times3} & I_{3\times3} \\ I_{3\times3} & 0_{3\times3} \end{pmatrix}$.

If the Plücker representation of both skew lines is normalized with the direction vector then this distance is metric in $\mathbb{R}^3$. The sign of this distance depends on the *side* where a line is located with respect to the other (clockwise or counterclockwise). The intersection between two lines is given by the constraint $\mathbf{L}^{\mathsf{T}}\mathsf{W}\mathbf{M} = 0$. Notice that the self-operation ($side\,(\mathbf{L},\mathbf{L}) = 0$) becomes the Plücker identity.

A 3D line has four degrees of freedom (DOF), as consequence at least four independent constraints are needed to solve the corresponding equations system. In non-central systems, four projection rays provide four independent constraints when they are generic (Fig 6.1 (a)). Notice that in a central system the four equations are not independent given that the four rays are coplanar.

However, given that Plücker coordinates set (without taking into account the Plücker identity) is an over-parametrized description of a line, the solution of a system of four homogeneous equations in $\mathbb{P}^5$ is a one dimensional subspace of $\mathbb{P}^5$. The solution of these equations system can be expressed as a singular value decomposition problem in which A is the collection of constraints such that $\mathsf{A}\mathbf{L} = \mathbf{0}$ with $\mathsf{A}_i = \left(\bar{\xi}_i^{\mathsf{T}}, \xi_i^{\mathsf{T}}\right)$ where A_i is the i^{th} row of A, in order that it can be written as the product of three matrices, U, Σ,V, such that U and V are orthogonal, Σ is diagonal and $\mathsf{A} = \mathsf{U}\Sigma\mathsf{V}^{\mathsf{T}}$.

The null space of this system is spanned by the last two columns of V (denoted $\mathbf{L}_0$ and $\mathbf{L}_1$ respectively). The null-space can be parametrized by $\mathbf{L} = \mathbf{L}_0 + \mathbf{L}_1\lambda$ (see Fig. 6.1 (b)) and imposing the Plücker identity we can compute the intersection of this subspace with the Klein Quadric obtaining two solutions: One is the sought line, the other is the axis of revolution of the visual system if it is axial or an arbitrary line in other case.

6.2.1 Degeneracies

There are some cases where the four rays are not independent and the system is degenerated. When the camera is axial the defining rays can be coplanar forming a Planar Viewing Surface (PVS) [Gasparini and Caglioti, 2011]. These degenerated cases are: the Axial-PVS case when the line is coplanar with the axis of symmetry and the Horizontal-PVS case when all the projecting rays lie in an horizontal plane. If the projection surface of the line is a ruled surface defined by only three rays this surface is called a regulus and the system is also under-determined.

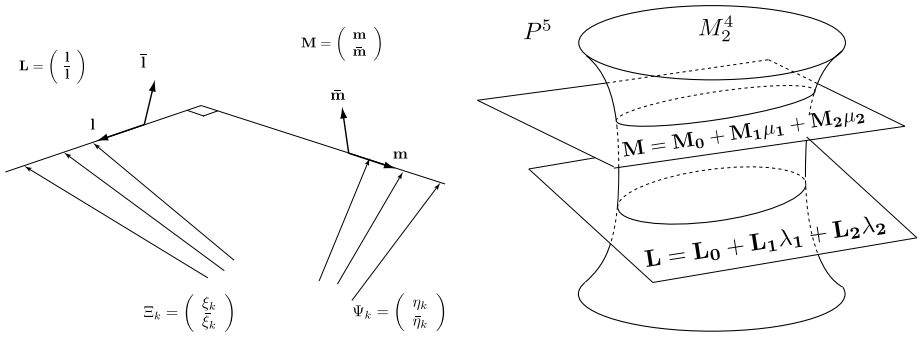


Figure 6.2: (a) Orthogonal junction of two lines. (b) Two dimensional subspaces in $\mathbb{P}^5$ and Klein Quadric.

6.3 Orthogonal junction of two lines

In this section we present the minimal solution for a junction of two orthogonal intersecting lines in non-central systems. Computing a junction composed by two orthogonal intersecting lines is a problem with 6 degrees of freedom (DOF). Four DOFs for one of the 3D lines, other for the depth of the line intersection and the 6th DOF for the angle defining the direction of the orthogonal line.

Given the lines $\mathbf{L} = \left(\mathbf{l}^\mathsf{T}, \bar{\mathbf{l}}^\mathsf{T}\right)^\mathsf{T}$, $\mathbf{M} = \left(\mathbf{m}^\mathsf{T}, \bar{\mathbf{m}}^\mathsf{T}\right)^\mathsf{T}$ and considering three generic rays Ξ_k intersecting the line $\mathbf{L}$ and three generic rays Ψ_k intersecting $\mathbf{M}$ (see Fig. 6.2 (a)), the corresponding null spaces of the under determined linear systems

$$\Xi_k^\mathsf{T} \mathsf{W} \mathbf{L} = 0 \quad , \quad \Psi_k^\mathsf{T} \mathsf{W} \mathbf{M} = 0 \tag{6.2}$$

are two subspaces of dimension 2 in $\mathbb{P}^5$. These subspaces are not contained in the Klein quadric and intersect the Klein quadric in two one-dimensional curves (see Fig. 6.2 (b)). A parametrized description of these 2-dimensional subspaces can be obtained by solving the null space of these systems with a singular value decomposition algorithm.

Taking as example the case of $\mathbf{L}$ and $\Xi_k = \left({\xi_k}^\mathsf{T}, {\bar{\xi}_k}^\mathsf{T}\right)^\mathsf{T}$: the matrix

$\mathsf{A} = \begin{bmatrix} \bar{\xi}_1 & \bar{\xi}_2 & \bar{\xi}_3 \\ \xi_1 & \xi_2 & \xi_3 \end{bmatrix}^\mathsf{T}$ can be written as the product of the three matrices, $\mathsf{A} = \mathsf{U}\Sigma\mathsf{V}^\mathsf{T}$ and the null space is spanned by the three last columns of matrix V denoted as $\mathbf{L}_0$, $\mathbf{L}_1$ and $\mathbf{L}_2$ parametrizing it with $L : \mathbb{P}^2 \mapsto \mathbb{P}^5$ defined as

$$\mathbf{L} = \mathbf{L}_0 + \mathbf{L}_1\lambda_1 + \mathbf{L}_2\lambda_2 \sim \mathsf{L}_m\lambda \tag{6.3}$$

where $\mathsf{L}_m = \begin{bmatrix} \mathbf{L}_0 & \mathbf{L}_1 & \mathbf{L}_2 \end{bmatrix}$ and $\lambda \in \mathbb{P}^2$ with $\lambda = \left(\tilde{\lambda}_0, \tilde{\lambda}_1, \tilde{\lambda}_2\right)^\mathsf{T} \sim (1, \lambda_1, \lambda_2)^\mathsf{T}$.

Analogously, for $\mathbf{M}$ we parametrize the null space with

$$\mathbf{M} = \mathbf{M}_0 + \mathbf{M}_1\lambda_1 + \mathbf{M}_2\lambda_2 \sim \mathsf{M}_m\mu \tag{6.4}$$

where $\mathsf{M}_m = \begin{bmatrix} \mathbf{M}_0 & \mathbf{M}_1 & \mathbf{M}_2 \end{bmatrix}$ and $\mu \in \mathbb{P}^2$ with $\mu = (\tilde{\mu}_0, \tilde{\mu}_1, \tilde{\mu}_2)^\mathsf{T} \sim (1, \mu_1, \mu_2)^\mathsf{T}$.

To obtain the four parameters λ_1, λ_2, μ_1 and μ_2 we need four additional independent constraints. These constraints are the condition of belonging at the Klein quadric for both lines, the perpendicularity between them and the intersection between them. These constraints are explicitly defined by

$$\mathbf{L}^\mathsf{T}\mathsf{W}\mathbf{L} = 0, \quad \mathbf{M}^\mathsf{T}\mathsf{W}\mathbf{M} = 0, \quad \mathbf{l}^\mathsf{T}\mathbf{m} = 0, \quad \mathbf{L}^\mathsf{T}\mathsf{W}\mathbf{M} = 0 \tag{6.5}$$

becoming a system of 4 quadratic equations in terms of λ_1, λ_2, μ_1 and μ_2 with the form

$$\lambda^\mathsf{T}\Omega_1\lambda = 0 \quad \text{where} \quad \Omega_1(i,j) = \mathbf{L}_i{}^\mathsf{T}\mathsf{W}\mathbf{L}_j \tag{6.6}$$

$$\mu^\mathsf{T}\Omega_2\mu = 0 \quad \text{where} \quad \Omega_2(i,j) = \mathbf{M}_i{}^\mathsf{T}\mathsf{W}\mathbf{M}_j \tag{6.7}$$

$$\lambda^\mathsf{T}\Omega_3\mu = 0 \quad \text{where} \quad \Omega_3(i,j) = \mathbf{l}_i{}^\mathsf{T}\mathbf{m}_j \tag{6.8}$$

$$\lambda^\mathsf{T}\Omega_4\mu = 0 \quad \text{where} \quad \Omega_4(i,j) = \mathbf{L}_i{}^\mathsf{T}\mathsf{W}\mathbf{M}_j. \tag{6.9}$$

This system can be manipulated to reduce the number of dimensions but increasing the degree of equations. Given the equations (6.8) and (6.9), imagine the $\mathbb{P}^2$ space of λ and the lines depending on μ:

$$\mathbf{U} = \Omega_3\mu \quad , \quad \mathbf{V} = \Omega_4\mu \tag{6.10}$$

Solving these equations for λ

$$\begin{bmatrix} \mathbf{U}^\mathsf{T} \\ \mathbf{V}^\mathsf{T} \end{bmatrix}\lambda = 0 \tag{6.11}$$

we obtain an explicit linear morphism

$$\lambda = \mathsf{C}\hat{\mu} \tag{6.12}$$

where $\hat{\mu} = \left(\tilde{\mu}_0^2, \tilde{\mu}_0\tilde{\mu}_1, \tilde{\mu}_0\tilde{\mu}_2, \tilde{\mu}_1^2, \tilde{\mu}_1\tilde{\mu}_2, \tilde{\mu}_2^2\right)^\mathsf{T} \sim \left(1, \mu_1, \mu_2, \mu_1^2, \mu_1\mu_2, \mu_2^2\right)^\mathsf{T}$ and $\mathsf{C}_{3\times 6}$ is computed analytically.

Substituting λ in equation (6.6) we have a system of two equations with two unknowns, the first is a quartic expression and the second a quadratic equation.

$$\hat{\mu}^\mathsf{T}\mathsf{C}^\mathsf{T}\Omega_1\mathsf{C}\hat{\mu} = 0 \tag{6.13}$$

$$\mu^\mathsf{T}\Omega_2\mu = 0 \tag{6.14}$$

Substituting (6.14) in (6.13) we obtain a single polynomial equation with one unknown of degree 8 which can be solved for μ_1.

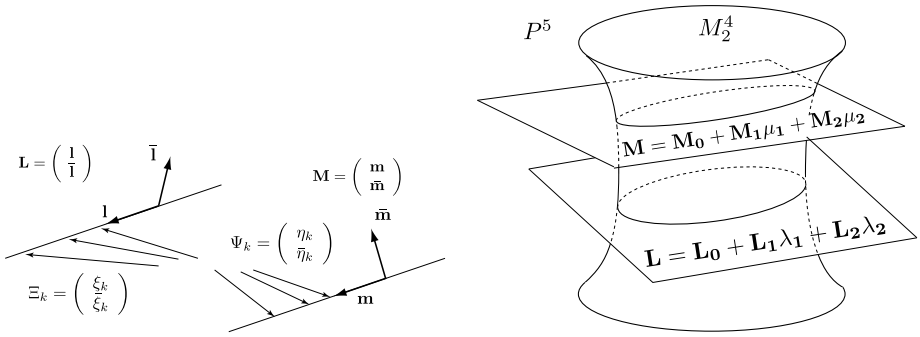


Figure 6.3: (a) Two parallel lines. (b) Two dimensional subspaces in $\mathbb{P}^5$ and Klein Quadric.

$$\sum_{i=0}^{8} c_i \mu_1^i = 0 \tag{6.15}$$

From the fundamental theorem of algebra we know that the number of solutions of this polynomial is 8. Using equation (6.14) we obtain 2 solutions of μ_2 from each solution for μ_1. One solution of (λ_1, λ_2) is obtained from each solution (μ_1, μ_2) (6.12), therefore the maximum number of solutions is 16. However, the majority of them are removed considering only real solutions , compatible with the equations and coherent with the orientation of the defining rays. Usually only 2 solutions remain.

6.4 Two parallel lines

In this section we present the minimal solution for computing the geometry of two parallel lines in non-central systems. Computing two parallel lines is a problem with 6 DOFs: in this particular case two DOFs for the common direction and two additional DOFs for each line. From three generic rays Ξ_k intersecting a line $\mathbf{L}$ and three generic rays Ψ_k intersecting a line $\mathbf{M}$ we compute the two dimensional subspaces (see Fig. 6.3 (b)),

$$\mathbf{L} = \mathbf{L}_0 + \mathbf{L}_1\lambda_1 + \mathbf{L}_2\lambda_2 \sim \mathsf{L}_m\lambda \tag{6.16}$$

$$\mathbf{M} = \mathbf{M}_0 + \mathbf{M}_1\lambda_1 + \mathbf{M}_2\lambda_2 \sim \mathsf{M}_m\mu \tag{6.17}$$

To obtain the four parameters λ_1, λ_2, μ_1 and μ_2 we need four additional constraints. First two are the constraint for each line of being in the Klein quadric or being a line,

$$\mathbf{L}^{\mathsf{T}}\mathbf{W}\mathbf{L} = 0, \tag{6.18}$$

$$\mathbf{M}^{\mathsf{T}}\mathbf{W}\mathbf{M} = 0\,. \tag{6.19}$$

The constraint of being parallels can be expressed as follows,

$$\mathbf{l} = (\mathbf{l}_0 + \mathbf{l}_1\lambda_1 + \mathbf{l}_2\lambda_2) = K\,(\mathbf{m}_0 + \mathbf{m}_1\mu_1 + \mathbf{m}_2\mu_2)\ , \tag{6.20}$$

which means three equations involving an additional unknown K. From (6.20) it is posible to compute μ in terms of λ obtaining the linear mapping between μ and λ

$$\mu = (\mathbf{m}_0, \mathbf{m}_1, \mathbf{m}_2)^{-1}\,(\mathbf{l}_0, \mathbf{l}_1, \mathbf{l}_2)\,\lambda\,. \tag{6.21}$$

Substituting μ in equation (6.19) we obtain two quadratic equations (6.18) and (6.19) depending on λ ,which can be considered as the intersection between two conics in $\mathbb{P}^2$ having four solutions.

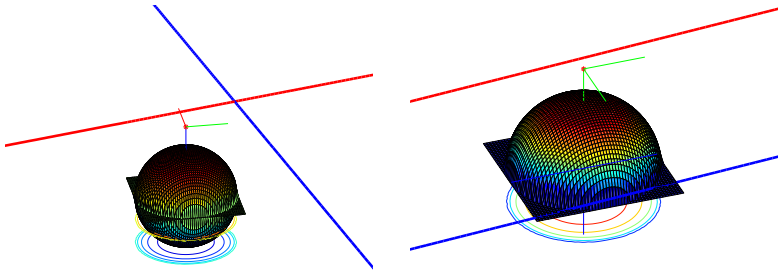
6.5 Simulations

In this section we evaluate the proposed method performing simulations of the line fitting process. As reference method we take the approach of Teller et al. [Teller and Hohmeyer, 1999] which is denoted as unconstrained linear method.

We consider two different cases: orthogonal junctions and parallel lines. In both cases, a collection of 100 pairs of lines are randomly generated. The length of the lines is 20m. In the case of a junction, the intersection between each pair of lines is located in a cube of side 4 m(see Fig. 6.4 (a)). In the case of parallel lines, two points are randomly located in a cube of side 4 m and then the orientation on the line is randomly computed (see Fig. 6.4 (a)). These lines are projected on an spherical-mirror-based image [Agrawal et al., 2010]. The catadioptric system is composed of a spherical mirror with radius of 1.2 m and a perspective camera located at 1.8 m from the center of the sphere. The resolution of the simulated camera is 1024×768.

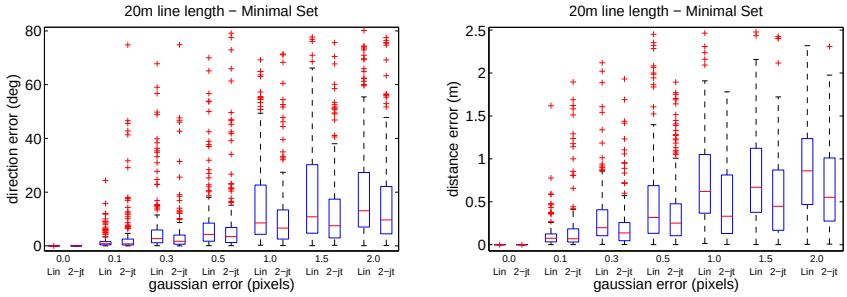
Gaussian noise of a given σ is added to the projected points. The value of σ variates from 0.1 to 2. Then 3D lines are computed from its projection in a single image using the linear Teller approach and using our proposal. For each value of σ we compute 100 pairs of lines. The 100 pairs of lines are the same for each value of σ.

The first simulation, is a comparison between the unconstrained linear approach and our proposals (junction in Fig. 6.5 and parallel lines in Fig. 6.6) considering the minimal set of defining points. The length of the lines in this setup is 20m. The linear approach is computed from 8 points (4 for each line) and the proposal (junction or parallel) is computed from 6 points (3 for each line). The 6 points used for the proposal is a subset of the 8 points used for the unconstrained linear approach to avoid biasing.



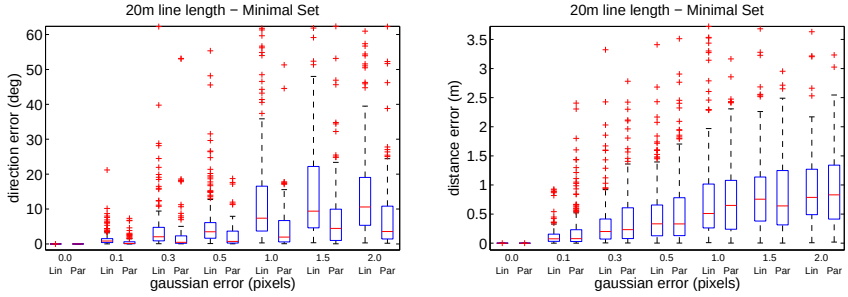
(a) (b)

Figure 6.4: Lines configuration example. (a) Junction. (b) Parallel.



(a) (b)

Figure 6.5: Lines estimation using the minimal set of points comparing the linear approach (Lin) and junction approach (2-jt): (a) Direction error (deg) (b) Distance error (m)



(a) (b)

Figure 6.6: Lines estimation using the minimal set of points comparing the linear approach (Lin) and parallel lines approach (Par): (a) Direction error (deg) (b) Distance error (m)

When using orthogonal junctions we take care of not using the intersection between lines which is a degenerated configuration with only 5 independent constraints.

The second simulation, is a comparison between the linear approach and our proposal considering more than the minimal set of points in the fitting. Considering more points (200 points), we obtain a least square fitting of the null spaces of dimension two in $\mathbb{P}^5$ described by L_m and M_m. In this case the number of points is the same for both linear and our proposal.

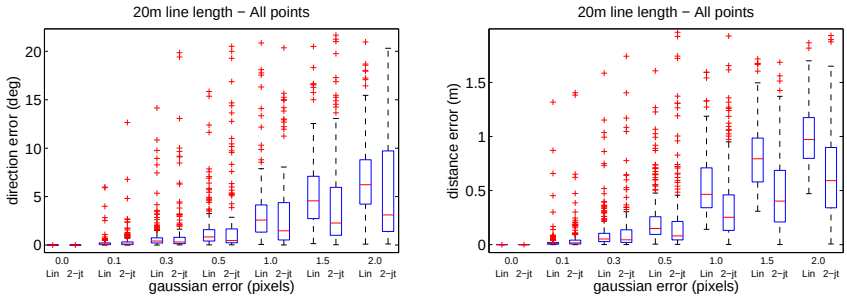
From the results we conclude that there is an improvement in the accuracy of the extracted lines. As expected, the improvement is more evident in the direction of the extracted line than in depth.

6.6 Experiments with real images

A reconstruction has been carried out to evaluate the performance of the method using a catadioptric system composed by a spherical mirror seen by a conventional camera with resolution 1280×1024 pixels.

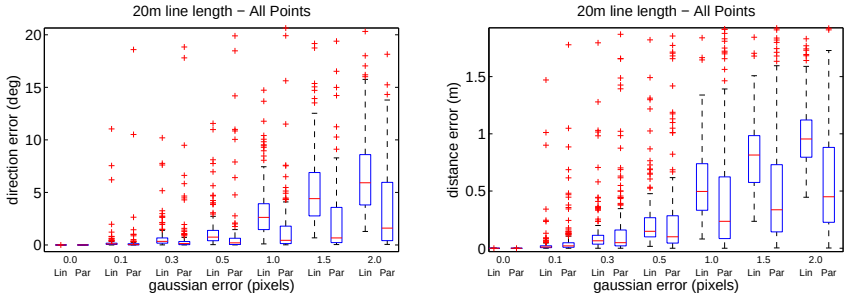
6.6.1 Calibration of the non-central system

First, the perspective camera has been independently calibrated. As the spherical catadioptric system is always axial the extrinsic parameters to calibrate are the tilt in the orientation of the camera, the radius of the sphere R_s and the distance from the center of the sphere to the perspective camera Z_s. Notice that in spherical catadioptric systems Z_s and R_s are coupled in the line projection. Therefore, only $Z_{rel} = \frac{Z_s}{R_s}$ can be recovered from a single line-image.



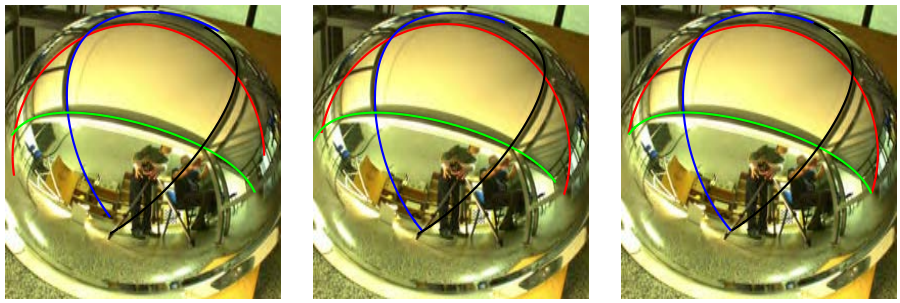
(a) (b)

Figure 6.7: Lines estimation using all points comparing the linear approach (Lin) and junction approach (2-jt): (a) Direction error (deg) in 20 m length lines. (b) Distance error (m) in 20 m length lines.



(a) (b)

Figure 6.8: Lines estimation of parallel lines using all points comparing the linear approach (Lin) and parallel lines approach (Par): (a) Direction error (deg) in 20 m length lines. (b) Distance error (m) in 20 m length lines.



(a) (b) (c)

Figure 6.9: Lines projection after fitting: (a) Linear. (b) junction. (c) Parallel.

In this case we have computed the tilt in the orientation and the relative distance Z_{rel} from the contour projection of the sphere which is a conic in the image plane of the perspective camera and also using the projection of the camera reflection on the image which is related with the direction of the center of the sphere. The radius of the sphere has been estimated from a 3D reconstruction of the mirror obtained from a RGB-D device. This metric can be finally refined using projections of patterns with a known dimension like in the calibration method presented in [Agrawal and Ramalingam, 2013].

6.6.2 3D line reconstruction

We have performed a reconstruction of four lines forming a rectangle. This disposition allows us to first reconstruct two different junctions and then reconstruct two different sets of parallel lines.

In Fig. 6.9 we show the forward projection of these four lines after the fitting using the junction. In Fig. 6.10 we show the reconstructions of these lines from 15 points using the Tellers linear method, our junction approach and our parallel lines approach.

As we can see in Fig. 6.9, line projections are very similar in the three cases. However, the reconstructions of the 3D lines are quite different (see Fig. 6.10). To measure the quality of the results we use the spanned planes by the pairs of lines. In the case of junctions we have compared the planes spanned by the two pairs of intersecting orthogonal lines. Both planes have a deviation of 9.72 degrees. In the case of parallel lines we compare the planes spanned by the two pairs of parallel lines having a deviation of 15.69 degrees. Note that this test is a way to measure quality in orientation but not in depth.

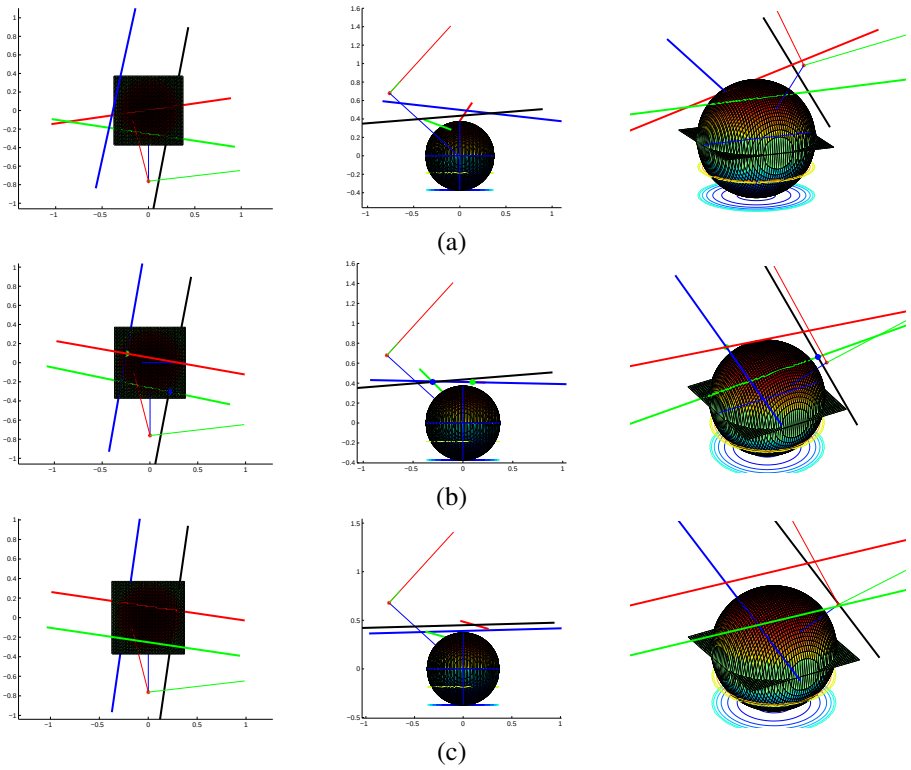


Figure 6.10: Reconstructed lines from 15 points, from left to right: XY-View, YZ-View, Orthographic-View (a) Teller's method, (b) junction proposal. (c) Parallel proposal

6.7 Summary

In this chapter we have presented the minimal solution for computing pairs of intersecting orthogonal lines and parallel lines from single images in non-central systems. This proposal has been tested in the particular case of spherical catadioptric systems. As expected, adding external geometric constraints improve the accuracy of the results. However this improvement does not allow reconstructing lines with lack of effective baseline. Thus, we have not yet solved the impediments for extensively using this kind of systems. The relation between the dimensions of the scene to reconstruct and the dimensions of the system is still too low in practice. Future work inevitably passes through designing new kind of catadioptric or dioptric systems with a bigger effective baseline by construction. Additionally, notice that all the techniques proposed in this chapter could be directly used in combination with these systems.

Chapter 7

Minimal solution for line parallel to plane in non-central cameras

Regarding non-central line-images and the method for recovering 3D from a single projection, in this chapter we continue the previously presented idea of using additional constraints to improve the reconstruction accuracy. In particular, we present a method to recover the geometry of the 3D line from three line-image points when imposing the line is parallel to a given plane. Since we are reducing the number of points involved in the minimal set which defines the line-image we reduce the complexity of robust schemes when extracting line-images. The drawback is that we need additional prior information about the distribution of lines in the scene. Under the Manhattan assumption horizontal lines are orthogonal to the gravity direction, that can be obtained from other sensors and used in our proposal. The formulation based on Plücker lines is integrated in a extraction pipeline which is tested with synthetic and real non-central circular panoramas. In addition we evaluate the accuracy of the proposal in comparison with the unconstrained method and the performance of the robust extractor.

7.1 Introduction

Line fitting in non-central systems is a challenging open topic with several advantages and questions to be solved. On one hand, line projections contain more geometric information because the four degrees of freedom of the line are encapsulated on the line-image. On the other hand, the 3D reconstruction from a single projection is very sensitive to noise. Third the complexity of an automatic extraction in a robust scheme

is high. The accuracy strongly depends on the effective baseline of current systems (which is not enough for practical using in most of cases). The complexity of extraction is related with the minimum required points for fitting, hence reducing the number of required points to fit lines in non-central systems can be very interesting in terms of computational time efficiency.

Regarding the algorithms used in central systems, the use of additional prior like the Manhattan assumption helps to improve the obtained results. In addition, knowing prior information from additional sensors like an accelerometer can reduce the complexity on the extraction of the features. In particular, when having a man-made scenario with dominant directions, vertical direction should be coincident with gravity direction and horizontal directions are orthogonal to this. Imaging a system in any position and orientation, this gravity direction can be measured from a IMU (Inertial Measurement Unit) and defines a plane which is parallel to the whole collection of horizontal lines. In this chapter we present the minimal solution for fitting a line parallel to a plane in non-central systems, which can exploit this extra information to fit lines reducing the minimal number of points from four to three. Notice that this applies for hard and soft Manhattan assumption because horizontal directions are not enforced to be orthogonal between them. The main advantage with respect to the central case is that, when having enough effective baseline, the 3D metric information and the scale of the scene is also recovered.

There exist some previous works dealing with geometric projection in non-central systems. Point back projection models for several non-central catadioptric systems have been studied in [Swaminathan et al., 2006]. In non-central systems, the forward projection has not to be a closed form solution. The forward projection of axial non-central catadioptric systems based on quadric mirrors can be obtained as the solution of a polynomial [Agrawal et al., 2010]. The non axial case is dealt in [Agrawal et al., 2011] and the calibration of a multi-axial non-central systems is studied in [Agrawal and Ramalingam, 2013]. The calibration of conical catadioptric systems and the epipolar geometry is treated in [López-Nicolás and Sagüés, 2014]. Another example of non-central images are the ones obtained from moving cameras without common viewpoint. The case of linear pushbroom camera is extensively studied in [Gupta and Hartley, 1997] and the non-central circular panorama and its epipolar geometry is dealt in [Menem and Pajdla, 2004]. The fundamentals for line fitting in non-central systems are computing the solution for lines intersecting four generic rays [Teller and Hohmeyer, 1999, Kneebone et al., 1998, Griffiths and Harris, 2011]. This theory is exploited in several works [Caglioti and Gasparini, 2005, Caglioti et al., 2007a, Gasparini and Caglioti, 2011] dealing with non-central catadioptric systems. In [Lanman et al., 2006] two non-central catadioptric systems are used for lines reconstruction. In non-central cameras not only lines can recovered from single projections but also other planar curves [Swaminathan et al., 2008]. Some additional restrictions can be used to improve the results: only considering lines orthogonal to the axis of revolution of the system [Pinciroli et al., 2005, Chen et al., 2011]. Cross-ratio properties [Perdigoto and

Araujo, 2012] or using off-axis systems to avoid degeneracies [Caglioti et al., 2007b] can be also used for improving the results. The pose of non-central catadioptric systems is estimated in [Miraldo and Araujo, 2014, Miraldo et al., 2015] using a prior line-based description of the scenario.

In this chapter we present the minimal solution for fitting a line parallel to a plane in non-central systems. This proposal allows to use additional prior information of the vertical direction and exploiting the Manhattan configuration to facilitate the line-image extraction and increasing the accuracy. We evaluate the proposal in terms of accuracy and robustness using extensive simulation. The proposal is integrated in a full extraction pipeline which is tested with synthetic and real images performing 3D reconstruction from single image.

7.2 Background

In this Section we describe the mathematical concepts and notation used in this Chapter. First, we introduce the procedure for 3D line estimation in non-central systems. Second, we present the measure of the effective baseline for a set of four points in a non-central image.

7.2.1 Line estimation in non-central systems

Using Plücker coordinates, the intersection between lines is described by the side operator (see Appendix A). A given line $\mathbf{L} = \left(\mathbf{l}^\mathsf{T}, \bar{\mathbf{l}}^\mathsf{T}\right)^\mathsf{T} \in \mathbb{P}^5$ (where $\mathbf{l} \in \mathbb{R}^3$ and $\bar{\mathbf{l}} \in \mathbb{R}^3$) intersects a ray $\Xi \in \mathbb{P}^5$ if

$$side\left(\mathbf{L}, \Xi\right) = \mathbf{L}^\mathsf{T} \mathsf{W} \Xi = \mathbf{l}^\mathsf{T} \bar{\xi} + \bar{\mathbf{l}}^\mathsf{T} \xi = 0 \tag{7.1}$$

where $\mathsf{W} = \begin{pmatrix} 0_{3\times3} & I_{3\times3} \\ I_{3\times3} & 0_{3\times3} \end{pmatrix}$.

Since a 3D line has four degrees of freedom we need at least 4 equations to solve for $\mathbf{L}$. If four projection rays from a 3D line provide four independent constraints we can compute the 3D line (see Section 5.3 in Chapter 5 for more details). In some degenerated cases four rays do not provide four independent equations [Gasparini and Caglioti, 2011]: when the projection surface is a plane (the line is coplanar with the axis of revolution or coplanar with the plane containing the circle) or when the projection surface is a regulus.

7.2.2 Measuring the effective baseline

When the points defining the line-image have not error (are perfect) the accuracy of the reconstruction is independent of the chosen points. However, in real situations the accuracy of the reconstruction is strongly dependent on the effective baseline of the

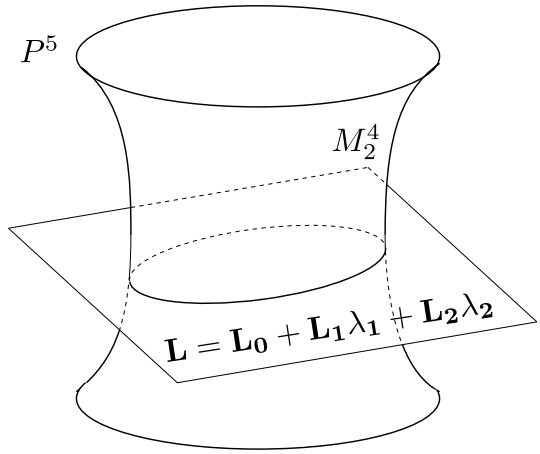


Figure 7.1: Two dimensional subspace in P^5 and Klein Quadric.

chosen set. In central systems all the rays intersect among them. In non-central at least some of the defining rays must be skew to provide enough independent equations. The euclidean distance among rays gives a measure of the quality of the selected set of rays for computing a 3D line. To measure the effective baseline of a set of n rays we use the function z_{bs} (4.49) presented in Chapter 4 which depends on the Euclidean distances between the defining rays.

7.3 Computing a 3D line parallel to a plane

When knowing that a line is parallel to a given plane $\mathbf{U} = \left(u_0, \mathbf{u}^\mathsf{T}\right)^\mathsf{T}$ the minimal solution of defining a line projection in non-central camera reduces from 4 degrees of freedom(DOFs) to 3 DOFs and consequently it can be computed from three independent rays intersecting the line.

Given the line $\mathbf{L} = \left(\mathbf{l}^\mathsf{T}, \bar{\mathbf{l}}^\mathsf{T}\right)^\mathsf{T}$ and considering three skew rays $\Xi_k = \left({\xi_k}^\mathsf{T}, {\bar{\xi}_k}^\mathsf{T}\right)^\mathsf{T}$ intersecting the line, the null space of the under determined linear system

$$\Xi_k^\mathsf{T} \mathsf{W} \mathbf{L} = 0 \qquad {\bar{\xi}_k}^\mathsf{T}\mathbf{l} + {\xi_k}^\mathsf{T}\bar{\mathbf{l}} = 0 \quad k = 1 \ldots 3 \tag{7.2}$$

is a subspace of dimension 2 in P^5. This subspace is not contained a priori in the Klein quadric and intersects the Klein quadric in a one dimensional curve (see Fig. 7.1).

A parametrized description of this 2-dim subspace can be obtained by solving the null space with a singular value decomposition algorithm.

Therefore, the matrix $\mathsf{A} = \begin{bmatrix} \bar{\xi}_1 & \bar{\xi}_2 & \bar{\xi}_3 \\ \xi_1 & \xi_2 & \xi_3 \end{bmatrix}^\mathsf{T}$ can be written as the product of the

three matrices, $\mathsf{A} = \mathsf{U}\Sigma V$ and taking the three last columns of matrix V denoted as $\mathbf{L}_0$, $\mathbf{L}_1$ and $\mathbf{L}_2$ we parametrize the null space with $L : P^2 \mapsto P^5$ defined as

$$\mathbf{L} = \mathbf{L}_0 + \mathbf{L}_1\lambda_1 + \mathbf{L}_2\lambda_2 \sim \mathsf{L}_m\lambda \tag{7.3}$$

where $\mathsf{L}_m = \begin{bmatrix} \mathbf{L}_0 & \mathbf{L}_1 & \mathbf{L}_2 \end{bmatrix}$ and $\lambda \in P^2 \qquad \lambda = \left(\tilde{\lambda}_0, \tilde{\lambda}_1, \tilde{\lambda}_2\right)^\mathsf{T} \sim (1, \lambda_1, \lambda_2)^\mathsf{T}$.

To obtain the two parameters λ_1 and λ_2 we need two independent constraints. First is the constraint of being in the Klein quadric or being a line,

$$\mathbf{L}^\mathsf{T}\mathsf{W}\mathbf{L} = 0, \tag{7.4}$$

and second constraint is the fact being parallel to the a plane which can be expressed with the orthogonality between the direction vector of the line $\mathbf{l}$ and the vector $\mathbf{u}$.

$$\mathbf{u}^\mathsf{T}\mathbf{l} = \mathbf{u}^\mathsf{T}\mathbf{l}_0 + \mathbf{u}^\mathsf{T}\mathbf{l}_1\lambda_1 + \mathbf{u}^\mathsf{T}\mathbf{l}_2\lambda_2 = 0 \tag{7.5}$$

From (7.5) λ_2 is expressed in terms of λ_1

$$\lambda_2 = -\frac{\mathbf{u}^\mathsf{T}\mathbf{l}_0 + \mathbf{u}^\mathsf{T}\mathbf{l}_1\lambda_1}{\mathbf{u}^\mathsf{T}\mathbf{l}_2} \tag{7.6}$$

and substituting equation (7.6) in equation (7.4) we obtain a quadratic equation (7.7) with one unknown λ_1 having two different solutions.

$$b_1\lambda_1^2 + 2b_2\lambda_1 + b_3 = 0 \tag{7.7}$$

where

$$b_1 = \beta_{2211} - 2\beta_{1212} + \beta_{1122} \tag{7.8}$$
$$b_2 = \beta_{0122} - \beta_{0212} - \beta_{1202} + \beta_{2201} \tag{7.9}$$
$$b_3 = \beta_{2200} - 2\beta_{0202} + \beta_{0022} \tag{7.10}$$

and $\beta_{ijkl} = \mathbf{L}_i{}^\mathsf{T}\mathsf{W}\mathbf{L}_j\mathbf{u}^\mathsf{T}\mathbf{l}_k\mathbf{u}^\mathsf{T}\mathbf{l}_l$. From each solution of λ_1 a solution for $\mathbf{L}$ is computed using

$$\mathbf{L} = \left(\mathbf{u}^\mathsf{T}\mathbf{l}_2\mathbf{L}_0 - \mathbf{u}^\mathsf{T}\mathbf{l}_0\mathbf{L}_2\right) + \left(\mathbf{u}^\mathsf{T}\mathbf{l}_2\mathbf{L}_1 - \mathbf{u}^\mathsf{T}\mathbf{l}_1\mathbf{L}_2\right)\lambda_1 \tag{7.11}$$

which is obtained by substituting (7.7) in (7.3).

7.4 Computing a 3D line with known direction

When the complete orientation of a system is known in a Manhattan scenario you can consider the direction $\mathbf{l}$ of the line $\mathbf{L}$ is known (it is one of the three dominant directions). That means that only two of the four degrees of freedom of the line must be computed and only two projecting rays Ξ_k for $k = 1 \dots 2$ are needed.

As we known the direction the intersection between the line $\mathbf{L}$ and a ray can be written as

$$\xi_k{}^{\mathsf{T}} \bar{\mathbf{l}} = -\bar{\xi}_k{}^{\mathsf{T}} \mathbf{l} \quad k = 1 \dots 2 \tag{7.12}$$

and also including the Plucker constraint $\mathbf{l}^{\mathsf{T}} \bar{\mathbf{l}} = 0$ we can compose a linear system of three equations

$$\begin{bmatrix} \xi_1{}^{\mathsf{T}} \\ \xi_2{}^{\mathsf{T}} \\ \mathbf{l}^{\mathsf{T}} \end{bmatrix} \bar{\mathbf{l}} = \begin{pmatrix} -\bar{\xi}_1{}^{\mathsf{T}} \mathbf{l} \\ -\bar{\xi}_2{}^{\mathsf{T}} \mathbf{l} \\ 0 \end{pmatrix} \tag{7.13}$$

and solving it for $\bar{\mathbf{l}}$ we obtain

$$\bar{\mathbf{l}} = \begin{bmatrix} \xi_1{}^{\mathsf{T}} \\ \xi_2{}^{\mathsf{T}} \\ \mathbf{l}^{\mathsf{T}} \end{bmatrix}^{-1} \begin{pmatrix} -\bar{\xi}_1{}^{\mathsf{T}} \mathbf{l} \\ -\bar{\xi}_2{}^{\mathsf{T}} \mathbf{l} \\ 0 \end{pmatrix} . \tag{7.14}$$

If having more than two rays the pseudo inverse can be used but in this case the solution does not perfectly fits the Plücker condition.

7.5 Line-extraction

In this section we present a method for line-image extraction in non-central images. We are assuming the scenario follows a hard-Manhattan (3 orthogonal dominant directions) structure or a soft-Manhattan structure (1 dominant direction and the rest of the lines on the plane orthogonal to this direction). In man-made environments the vertical direction is coincident with the direction of the gravity. We assume that this direction $\mathbf{u}$ is known (e.g. it can be obtained from a IMU system or from other source) but the system is not aligned with this direction (see Fig. 7.2). Notice that this direction is the same for all vertical lines and it is orthogonal to the horizontal plane, hence when we have a line $\mathbf{L}_H$ which is horizontal (parallel to the horizontal plane) we can use the 3-points method presented in Section 7.3 and when the line is vertical we can use the 2-points method presented in Section 7.4. The method is proposed for non-central circular panoramas [Menem and Pajdla, 2004] but it can be adapted to other non-central systems.

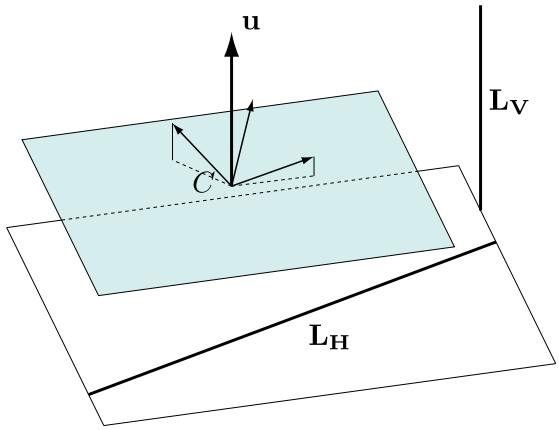


Figure 7.2: Euclidean representation of a reference system C with any orientation and a known direction $\mathbf{u}$ in a Manhattan scenario.

For automatic extraction, first we extract edges from the image using Canny detector. To avoid false edges we first filter the image using a Rolling Guidance Filter [Zhang et al., 2014]. This edges are then stored in connected components. From each component line-images are extracted using a greedy RANSAC approach. After that lines are clustered depending on main directions and finally the segments are merged to improve the extraction results. More in detail:

7.5.1 Extended RANSAC approach

Having a collection of points which are candidates to be supporting points of a line-image, we generate a set of collections of three points. The size of this set is the number of iterations $k_{attempts}$ of the algorithm which is estimated from the ratio of inliers $\omega = \frac{number\ of\ inliers}{total\ number\ of\ points}$, the number of elements defining the minimal set p and the probability P that at least in one random subset p all selected points are inliers with

$$k_{attempts} = \frac{\log(1-P)}{\log(1-\omega^p)}. \tag{7.15}$$

We follow the pre-filter scheme described in PROSAC [Chum and Matas, 2005] and USAC [Raguram et al., 2013] using a measure of the quality of the points, which in this particular case is the measure of effective baseline (see Section 7.2.2). This allows generating hypothesis that are more likely to be valid.

Then, from each valid set of 3 points three hypothesis are computed: two assuming the line is orthogonal to the vertical direction (see Section 7.3) and one assuming the line is parallel to the vertical direction (see Section 7.4). Only one of these solutions

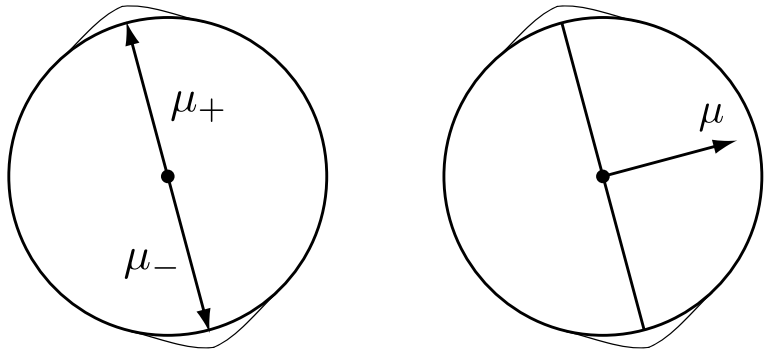


Figure 7.3: (a) Mixture of Gaussians based on vanishing points. (b) Mixture of Gaussians based on direction.

have enough supporting points on the image, hence we can store the three proposal as hypothesis and use the voting scheme to discriminate among them.

Using the distance presented in Chapter 5 we test each hypothesis obtaining a voting. The most voted line-projection is selected, obtaining the 3D Plucker coordinates of the line and the corresponding labelling classifying the line (and the supporting points) as horizontal or vertical.

7.5.2 Clustering

If the connected components are small they do not contain enough baseline to obtain a good reconstruction. For improving the results we perform a merging algorithm which connects different sets of supporting points. If we assume that horizontal lines are arranged in dominant directions we can first cluster the horizontal lines in two classes.

Considering only the directions of lines which in general are elements of an Euclidean space E^3, the direction of horizontal lines are contained in the same horizontal plane. The goal is clustering them in a unidimensional space defined by the unitary circle.

For clustering we use an algorithm based on mixture of Gaussians. Consider, the conditioned probability of the normalized direction vector $\mathbf{x}_i$ given that the label of the direction is c and the distribution parameters $\theta_c = (\mu_c, \sigma_c)$ is described by a Gaussian distribution

$$P(\mathbf{x}_i|c,\theta_c) = \frac{1}{\sqrt{2\pi}\sigma_c} e^{-\frac{d_G(\mathbf{x}_i,\mu_c)^2}{2\sigma_c^2}} \tag{7.16}$$

where $d_G(\mathbf{x}_i,\mu_c)$ is a function measuring the distance between the direction vector $\mathbf{x}_i$ and the mean direction vector μ_c of the distribution.

The problem can be dealt from two points of view: First is using one distribution for each vanishing point meaning that we need to use two Gaussian distributions for each direction (see Fig. 7.3 (a)). In this case the conditioned probability using the normalized direction vectors in E^2 is

$$P(\mathbf{x}_i|c,\theta_c) = \frac{1}{\sqrt{2\pi}\sigma_c} e^{-\frac{\arccos \mu^\mathsf{T} \mathbf{x}_i^{\,2}}{2\sigma_c^2}} \tag{7.17}$$

which uses $d_G(\mathbf{x}_i,\mu_c) = \arccos \mu^\mathsf{T}\mathbf{x}_i$ as measure of distance between the mean parameter of the distribution and each sample.

Instead of this distance, we propose to use the distance to the orthogonal space $d_G(\mathbf{x}_i,\mu_c) = \arcsin \mu^\mathsf{T}\mathbf{x}_i$ defining the distribution

$$P(\mathbf{x}_i|c,\theta_c) = \frac{1}{\sqrt{2\pi}\sigma_c} e^{-\frac{\arcsin \mu^\mathsf{T} \mathbf{x}_i^{\,2}}{2\sigma_c^2}} \tag{7.18}$$

which properly represents the two senses of the direction and both vanishing points simultaneously (see Fig. 7.3 (b)). In addition, with this description we only need two Gaussians for the two dominant directions.

The initial solution is computed from two overlapped histograms. The posterior probability of each sample is estimated using the Bayes rule:

$$P(c|\mathbf{x}_i,\theta_c) = \frac{P(\mathbf{x}_i|c,\theta_c)\,P(c)}{\sum_c P(\mathbf{x}_i|c,\theta_c)\,P(c)} \ . \tag{7.19}$$

For clustering the directions in the two classes we maximize the likelihood described as

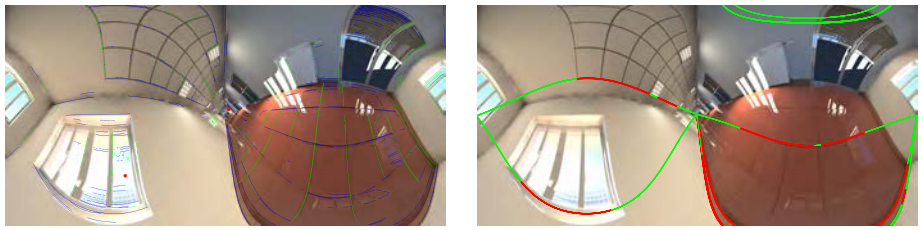
$$\theta = \arg\max_{\theta'} \prod_i \sum_c P\left(\mathbf{x}_i|c,\theta_c'\right) P(c) \ . \tag{7.20}$$

To improve the convexity of the function the logarithm of the likelihood is used in an iterative scheme

$$\theta = \arg\max_{\theta'} \sum_i log \sum_c P\left(\mathbf{x}_i|c,\theta_c'\right) P(c) \tag{7.21}$$

for obtaining the parameters $\theta = (\theta_1,\theta_2)$ of the two dominant directions.

Depending on if the scenario is hard-Manhattan or soft-Manhattan the orthogonality constraint between the two Gaussians can be imposed on the optimization. After the clustering the segments supporting horizontal lines are classified in two dominant directions (see Fig. 7.4).



(a) (b)

Figure 7.4: Extraction pipeline example on synthetic image: (a) Results of the clustering on the image (blue for horizontal dominant direction 1 and green for the horizontal dominant direction 2). (b) Example of extracted line-images on direction 1.

7.5.3 Merging

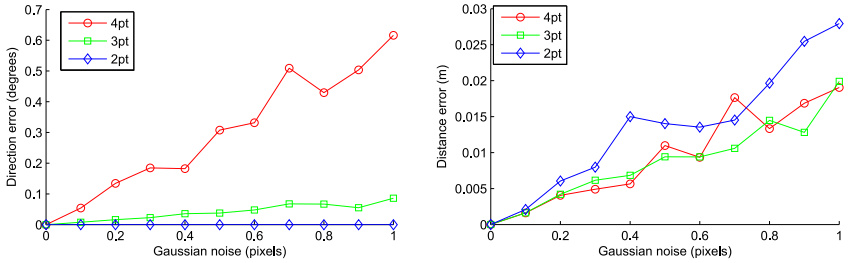
Once the clustering is performed, we can exploit the connectivity of the extracted segments to generate hypothesis of connection. The connectivity map is obtaining composing a set of local Delaunay triangulations computed on the tangent map of the torus defined by the back-projection model.

7.6 Evaluation and experiments

In this Section we present a set of experiments designed for evaluating the performance of the proposed method. First, we evaluate the accuracy in 3D line fitting. Second, we test the robustness of the extraction proposal and finally we show examples of line projections extractions in synthetic and real images.

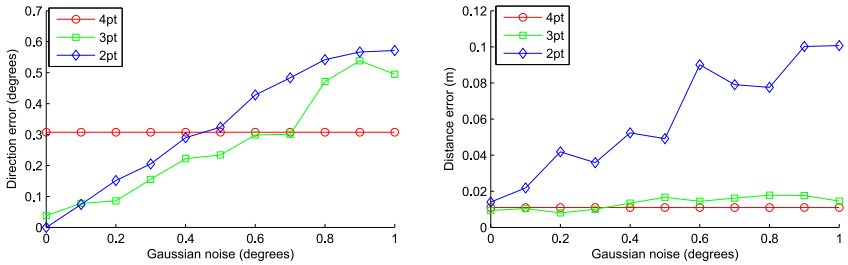
7.6.1 Accuracy evaluation

In this Section we present a comparison in terms of accuracy among the 4 points algorithm (without constraints), the 3 points algorithm (assuming the line is parallel to a plane), and the 2 points algorithm (assuming the direction of the line is known). The used system is a non-central panoramic imaging system with a radius $R_c = 0.5$ and a field of view of 180×360 and a resolution of 2048×4096 pixels. For comparing the accuracy we have simulated the projection of 100 random 3D lines and we have added Gaussian noise to the line-image points. Then the lines are fitted using the three approaches. For the first approach (4pt), 4 points of the line-image are used to fit the line (these points are selected for maximizing the effective baseline). For the second approach (3pt), 3 points of the line-image and the orientation of a plane parallel to the line are used for fitting the line. In this case we also add Gaussian noise to the orientation of the plane and the points of the line-image are selected to maximize the effective



(a) (b)

Figure 7.5: Comparison in terms of the accuracy of line fitting. Gaussian noise in defining points without error in prior information. (a) Direction Error (degrees). (b) Distance error (m).



(a) (b)

Figure 7.6: Comparison in terms of the accuracy of line fitting. Gaussian noise in orientation of prior information with σ from 0 to 1 degrees and a Gaussian noise in image points with $\sigma = 0.5$ pixels (degrees). (a) Direction Error. (b) Distance error (m).

baseline. For the third approach (2pt) the solution is computed from two points of the line-image but assuming that the direction of the line is known and having a Gaussian noise on the orientation of the direction.

In Fig. 7.5 and Fig. 7.6, we represent the estimated accuracy according to two errors:

- the direction error between the estimated 3D line and the ground truth line computed as $\varepsilon_\varphi = \arccos \mathbf{l}^\mathsf{T} \mathbf{l}_{GT}$.
- the depth error between the estimated 3D line and the ground truth line computed as $\varepsilon_d = \left| \|\bar{\mathbf{l}}\| - \|\bar{\mathbf{l}}_{GT}\| \right|$.

Fig. 7.5 shows the influence of points error in accuracy when the prior has no noise. The abscissa axis corresponds to a variation in the Gaussian noise added to the defining points from $\sigma = 0$ pixels to $\sigma = 1$ pixel. As expected, the error is always bigger for the

unconstrained method (4pt). Notice that when the direction is known (2pt) the direction is not really estimated and the plotted error in Fig. 7.5 (a) is always zero.

In Fig. 7.6 a Gaussian fixed error of $\sigma = 0.5$ pixels is added to the defining points and we introduce a variation in the Gaussian error added to the prior information: the prior known plane for the 3-points case and the prior known direction for the 2-points case. The abscissa axis corresponds to a variation in the Gaussian noise added to the prior information from $\sigma = 0$ degrees to $\sigma = 1$ degree. Notice that in this case the error when using the 4-points approach is constant because we are introducing a variation in the prior noise which does not affect the unconstrained method. As expected the worst results are for the 2-points method which is the one using more prior information.

7.6.2 Robustness evaluation

In this Section we present the robustness of the robust method presented in Section 7.5.1. We simulate the projection of 100 random lines with two different kind of noise. The projected points have a Gaussian noise with $\sigma = 0.5 pixels$ representing the projection error. In addition we have added spurious points following a uniform distribution. We denote the original points as inliers and the spurious points as outliers. The rate between the generated inliers and outliers is denoted as ω and its fixed to 0.6. Not all the inlier points have the same influence when fitting the model because we first pre-filter de hypothesis with the baseline criterion. In Fig. 7.7 we show the results of precision and recall, where $precision = \frac{true\ positives}{true\ positives + false\ positives}$ and $recall = \frac{true\ positives}{true\ positives + true\ negatives}$. The vertical lines represent the value for the number of attempts estimated with equation 7.15 with a probability of being successful $P = 0.99$.

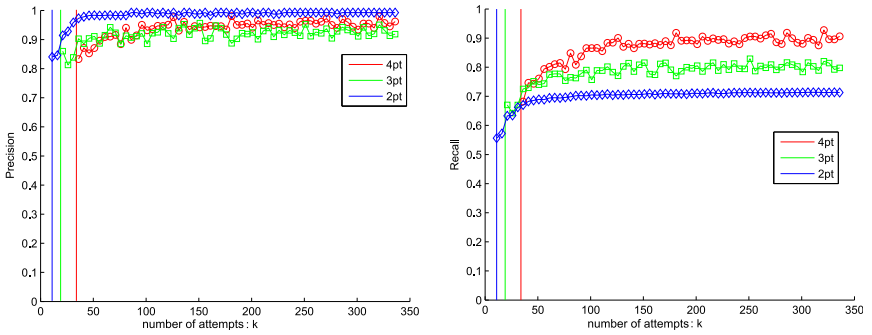
Due to the nature of robust extraction, where a single false positive can condition the whole fitting we are interested on having a high precision to the detriment of the recall. As expected, the number of attempts needed for an acceptable precision is inferior for the constrained methods although a little higher than the theoretical value.

7.6.3 Line extraction with synthetic images and real images

In this section we present the results of the pipeline presented in Section 7.5 obtained from synthetic and real images of a Manhattan scenario. The synthetic images have been generated with the raytracing software MegaPov [1] which allows defining parametric non-central cameras. The scenario is a modification of a publicly available synthetic scenario [2]. The synthetic panoramas have a resolutions of 4096×2048 pixels and the radius of the generation circle is $R_c = 0.5m$. In Fig. 7.8 we show the results of extracted line-images of the first horizontal dominant direction. The line projections

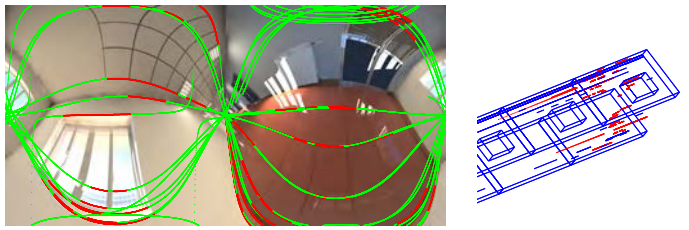
[1] megapov.inetart.net

[2] http://hof.povray.org/office-13.html The Office - Jaime Vives Piqueres, 2004



(a) (b)

Figure 7.7: Precision and recall of the robust approach depending on the number of attempts for a ratio of inliers $\omega = 0.6$. Red for the 4-pt method, green for the 3-pt method and blue for the 2-pt method. (a) Precision, (b) Recall.



(a) (b)

Figure 7.8: Extraction example on synthetic image: (a) Extracted lines following the main direction over a non-central panorama (green for the parametric line and red for supporting points). (b) 3D view of the ground truth (in blue) and the reconstruction of the supporting points (in red)

are depicted in green and the supporting points in red (see Fig. 7.8 (a)). In the 3D view Fig. 7.8 (b) the ground truth is shown in blue and the supporting 3D points in red.

The real images have been generated using the multi-camera approximation presented in Chapter 5. The panorama has been composed using a rotating off-axis fisheye camera (uEye UI-148xSE-C with lens Lensagon CF5M1414) with a radius of $R_c = 0.5m$. The panoramic images have a size of 1570×4096 pixels and the original images have a resolution of 1920×2560 pixels. In Fig.7.9 we show an and example of the extraction in a multi-camera panorama generated from 180 cameras. The line projections are depicted in green and the supporting points in red (see Fig. 7.9 (a)). In Fig.7.9 (b-c) we show the extracted 3D segments in red.

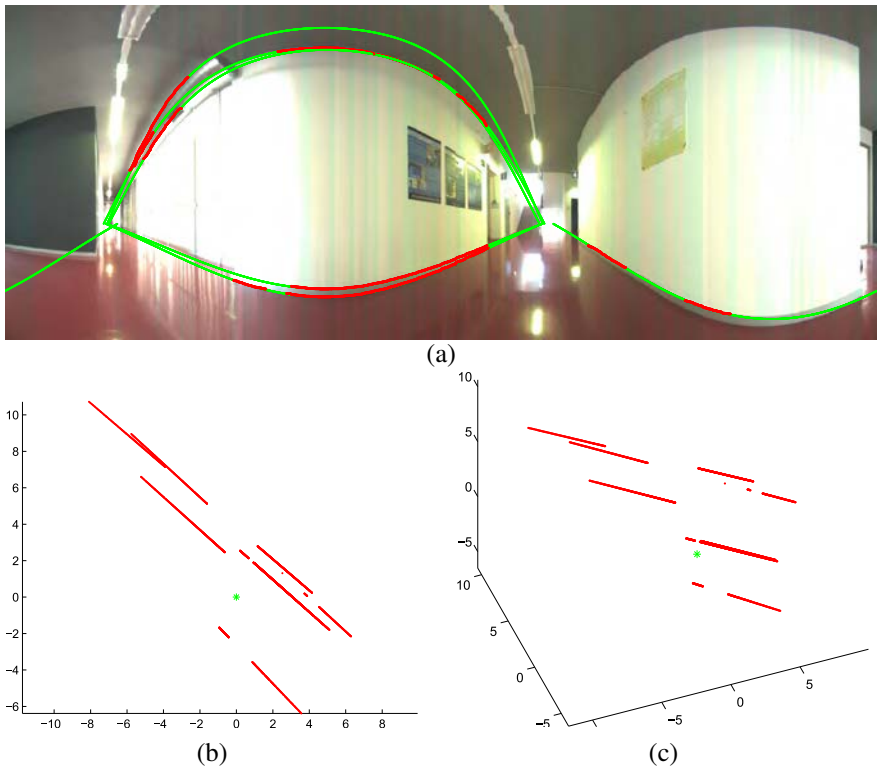


Figure 7.9: Extraction example on real image: (a) Extracted lines following the main direction over a non-central panorama (green for the parametric line and red for supporting points). (b) Top view of reconstructed supporting points. (c) Perspective view of reconstructed supporting points.

7.7 Summary

In this Chapter, we present the minimal solution to compute a line which is parallel to a given plane in non-central images. This approach exploits the configuration of Manhattan scenarios in combination with the prior knowledge of the vertical direction, obtained for example from an accelerometer. The accuracy and robustness of the approach has been evaluated with simulated experiments. Depending on the quality of the prior information the accuracy of the 3D estimation from single projection can be improved despite the lack of effective baseline. The proposal has been integrated in a robust extraction scheme and tested with synthetic and real images.

Part III

Conclusions and appendix

Chapter 8

Conclusions and future work

In this thesis we study the geometry behind line projections in different classes of omnidirectional cameras. First part of the thesis is dedicated to central systems where we focus on line-image extraction and calibration. In particular, we present a method for line-image extraction in hypercatadioptric systems when the calibration is known. We also analyze the uncertainty propagation of the method and particularize the vanishing point computation on hypercatadioptric images developing the self-polar triangle approach. From this, we generalize the line-image description to different classes of catadioptric and dioptric systems. Using the proposed unified framework we exploit the *plumbline* constraint obtaining closed solutions that allow us to simultaneously extract line-images and a calibration parameter which is common to all the different central systems. The *plumbline* constraint allows computing the main calibration parameter and the projection plane of the line from three points. This constraint is integrated in a RANSAC scheme for automatic line-image extraction while obtaining the main calibration parameter. *Plumbline* constraints are then reformulated using gradients as additional input information reducing the number of required points.

Second part of the thesis is devoted to line projections in non-central systems. Line projections have a special interest in non-central systems because it is possible to recover the complete 3D information of the line from a single projection. In particular, 4 points on the line-image directly define the line-image, the 3D line and the projection surface. Regarding this idea, we generalize the description used in central systems developing a representation of line-image for non-central systems with revolution symmetry. This representation is particularized for the cases of conical and spherical catadioptric systems obtaining quartic polynomial expressions and, in the case of the conical catadioptric line-image, a simpler expression which allows fitting the line-image from 5 points recovering the 3D line and the geometry of the mirror. Main problem when recovering a 3D line from a single projection is that results are very noise sensitive. This prevents using this technique in real applications because in practice, it is only possible to reconstruct lines in regions which are close to the imaging system. The

uncertainty in fitting edges into the 4 dof curves makes the extraction of line-images a challenging task. To deal with this problem we have explored different approaches. To measure the quality of a set of points as a function of the non-centrality of the rays we introduce the concept of effective baseline which strongly depends on the suitability of the selected points and the class of non-central system. This measure allows making a comparison among different kind of systems. It also permits to determine if a particular selection is adequate allowing a pre-evaluation step in the extraction process. To achieve the line-image extraction we also use a measure of distance determining if a point belongs to a line-image or not. Up to our knowledge this thesis is the first work where automatic line-extraction without prior information about the orientation or location of the 3D lines is achieved in non-central systems.

Besides non-central catadioptric systems we study the non-central circular panorama showing different ways of implementing it. Our conclusion is that this kind of system provides greater effective baseline than non-central catadioptric systems obtaining better accuracy and better performance in line-image extraction. Another way for increasing the accuracy in 3D line reconstruction is exploiting geometric constraints for improving the accuracy and stability in line reconstruction. We develop minimal solutions when computing pairs of lines for non-central systems. In particular, we develop the cases of intersecting orthogonal lines and pair of parallel lines. Finally we present the minimal solution for computing a line which is parallel to a given plane which can be exploited using the prior information of the vertical direction in a Manhattan scenario.

Regarding minimal solutions and the number of points needed for fitting the proposed models, we use robust approaches based on RANSAC for fitting these models. Notice that there is an increasing difficulty in the models and their fitting in the different approaches presented along this thesis. In Chapter 2 we are fitting models defined by 2 elements. In Chapter 3 the number of elements is 3 if using the closed form solutions for the plumbline constraint or 2 if using also the gradient approaches (2 points location and 2 gradients on these points). In Chapters 4 and 5 we need at least 4 elements since we are computing the 4 DOFs of the 3D line from a single projection. Because of the elevate number of attempts needed, we use the pre-filtering step based on the effective-baseline measure to previously remove bad-conditioned attempts. Finally, in Chapters 6 and 7 we use additional information in the fitting and we only have to use 3 elements. In Chapters 2, 3 and 4 we propose a calculus which is clearly focused on the image representation. In fact, the line-image model presented in Chapter 4 is a generalization of the previous models used in Chapters 2 and 3. By contrast, in Chapters 5, 6 and 7 the fitting is done in the $\mathbb{P}^5$ space dealing with projecting rays instead of points. This is a more general approach because the back projection model is encapsulated in the definition of the projecting rays. This approach also allows exploiting the properties of Plücker coordinates and fitting lines in non-central systems when the system has not symmetry of revolution. Nevertheless, working directly on the image allows us to perform the calibration estimations presented in Chapters 3 and 4 because the calibration

parameters are involved in the expression of the line-image.

In conclusion, increasing the field of view of the cameras has a main drawback consisting of inducing high distortions on the images. As a result of these distortions line projections become curves which are more difficult to deal with. However, during the development of this thesis we state that these drawbacks can be converted in advantages. On the one hand, the distortion of the projected curve encodes the geometry of the system allowing the estimation of the calibration of the system. On the other hand, the non-central case is even more interesting because the complete 3D line is mapped on the line-image and it is possible to perform 3D reconstructions from single images without imposing constraints. The second part of the thesis is conducted around this idea which supposes an important advantage with respect to conventional cameras.

In the past, despite the theory behind reconstructing lines from single projections in non-central systems was very promising, the approach was not yet integrated in a real system and the issue is still open because of the extraction difficulties and the lack of accuracy. In this thesis we study the underlying problems with new proposals. As result of our work we propose a magnitude we call effective-baseline. This effective-baseline is related with the accuracy, which has been insufficient in current non-central systems and, depends on the size and the design of the non-central system. We have found that the best configuration among current non-central systems is the non-central circular panoramic system. In fact, up to our knowledge, our proposal is the first work achieving 3D reconstruction of a real scenario using as input a single non-central image and not assuming restrictions about the structure of the scene. In particular, we have reconstructed a corridor using a system whose dimensions are in the typical order of a mobile robot dimensions. Notice that for obtaining the same accuracy from a single image with the non-central catadioptric systems the dimension of the mirrors must be similar or even greater than the size of the corridor being not practical for a real application.

The immediate future work is a direct consequence of these conclusions. First of all, any effort in designing new non-central systems with more effective baseline directly will improve the results of our proposals which are adaptable to new systems. Considering the current technology, the non-central circular panorama is a feasible system for 3D reconstruction. There can be different ways of achieving this configuration. In particular, a rotating camera is the easiest configuration for validating the theoretical approaches and methods we propose, however in practice this system is not capable of acquiring coherent non-central panoramas when the system is moving. A feasible solution to solve this problem is manufacturing the multi-camera system proposed in Chapter 5. This configuration can be achieved for example by installing an array of line-cameras (camera with only a column of pixels) around a circular robot. The advantage of this configuration is that the non-central panorama can be acquired in a single shot but also that we conserve free space inside the defining circle that can be used for storing the rest of the parts of the robot. The synchronization problems can be easily solved using a trigger signal for the simultaneous acquisition. Notice

that each camera only has to acquire a single column. The stitching of the non-central panorama can be addressed by using parallel computing with field programmable gate arrays (FPGAs).

Of course this configuration directly suggests a question about the feasibility of the system: Why rejecting most of the information of the array of cameras for stitching a panorama? Why not directly using the information of the whole collection of cameras for performing a reconstruction? Our answers to these questions are the following: First of all, our proposal only has to manage a small of amount of information from each camera reducing the needed memory for storing it. But the main reason is that with our proposal we avoid the costly and prone to error matching step in the 3D reconstruction pipeline. In conventional cameras, the geometric discipline about 3D multi-view reconstruction is practically closed and solved, and the big effort is focused on the artificial intelligence part where the system has to identify features which have to be invariant to transformations. Our proposal is changing the paradigm by avoiding this matching step and solving the 3D reconstruction with a pure geometrical solution.

With such a multi-camera device, and once that it is possible to acquire non-central panoramic sequences in a moving system, next step is developing the uncertainty propagation of the estimated 3D lines for performing a Simultaneous Localization and Mapping (SLAM) system. Notice that our proposal can take advantage of the metric reconstruction of lines for avoiding the scale drift. On the other hand, some of the horizontal line features can be shown during long parts of the sequence improving the reconstruction.

Some research line of future work taking into account the work presented in this thesis are discussed next. Notice that we are estimating 3D information in non-central systems by imposing some structure prior information about the scene. In particular, we are imposing that the edges of the scene are arranged in straight lines. It is possible to consider other arranges like circles, or planar curves and if the projection surface conserves the degrees of freedom of the original curve the problem could be addressed in a similar way. Besides, we can explore additional geometric constraints between lines for improving the 3D reconstruction like corners formed by three orthogonal incident lines.

Appendix A

Homogeneous space and Grassmann-Cayley algebra

Consider a projective coordinate system in $\mathbb{P}^3$ which is equivalent to choosing a basis $\mathbf{e}_0$,$\mathbf{e}_1$,$\mathbf{e}_2$,$\mathbf{e}_3$ in $\mathbb{R}^4$, we define the point x

$$\mathsf{x} = \sum_{i=0}^{3} x_i \mathbf{e}_i = x_0 \mathbf{e}_0 + [\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3]\,\mathbf{x} = [\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_0] \begin{pmatrix} \mathbf{x} \\ x_0 \end{pmatrix} \tag{A.1}$$

with coordinates $\mathbf{X} = \left(\mathbf{x}^\mathsf{T}, x_0\right)^\mathsf{T}$ and basis $[\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_0]$.

The coordinates of the same point in a different basis $\mathbf{e}'_0$,$\mathbf{e}'_1$,$\mathbf{e}'_2$,$\mathbf{e}'_3$ in $\mathbb{R}^4$

$$\mathsf{x} = [\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_0]\,\mathbf{X} = \left[\mathbf{e}'_1, \mathbf{e}'_2, \mathbf{e}'_3, \mathbf{e}'_0\right]\mathbf{X}' \tag{A.2}$$

can be expressed by the endomorphism $\mathbf{X} = \mathsf{H}\mathbf{X}'$. If H is a Euclidean transformation it can be written as

$$\mathbf{X} = \begin{pmatrix} \mathsf{R} & \mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix} \mathbf{X}' \tag{A.3}$$

A.1 Exterior product of homogeneous vectors

The exterior (or outer) product [Kanatani, 2015, Pottmann and Wallner, 2001] between two homogeneous vectors assigns to two vectors $\mathbf{x}$ and $\mathbf{y}$ of $\mathbb{R}^4$ a new bivector, which is contained in a new vector space, denoted by $\Lambda^2\mathbb{R}^4$. The operation has the following properties,

antisymmetry: $\mathsf{x}\wedge\mathsf{y}=-\mathsf{y}\wedge\mathsf{x}$. In particular $\mathsf{x}\wedge\mathsf{x}=0$.
bi-linearity: $\mathsf{x}\wedge(\lambda\mathsf{y}+\mu\mathsf{z})=\lambda\mathsf{x}\wedge+\mu\mathsf{x}\wedge\mathsf{z}$,
associativity: $\mathsf{x}\wedge(\mathsf{y}\wedge\mathsf{z})=(\mathsf{x}\wedge\mathsf{y})\wedge\mathsf{z}$
scalar operation: $\lambda\wedge\mu=\lambda\mu\ \lambda\wedge\mathsf{x}=\lambda x$

In particular for the basis $\mathbf{e}_0$,$\mathbf{e}_1$,$\mathbf{e}_2$,$\mathbf{e}_3$

$$\mathbf{e}_i\wedge\mathbf{e}_j=-\mathbf{e}_j\wedge\mathbf{e}_i \quad i,j=0,1,2,3 \tag{A.4}$$

$$\mathbf{e}_i\wedge\mathbf{e}_i=0 \quad i=0,1,2,3 \tag{A.5}$$

A.2 Plücker coordinates of lines

A straight 3D line is a geometric arrange of 3D points having four degrees of freedom (DOFs) which is defined by the span of two 3D points or the intersection of two planes. Considering projective geometry for describing the geometry of an scene, the set of straight 3D lines forms a four-dimensional manifold induced by the underlying geometry of the projective space $\mathbb{P}^3$.

The Plücker description of a 3D line is a $\mathbb{P}^5$ representation of a line obtained from the null space of any pair of points x and y

$$\begin{pmatrix} x_0 & x_1 & x_2 & x_3 \\ y_0 & y_1 & y_2 & y_3 \end{pmatrix} \tag{A.6}$$

that can be computed using the exterior product

$$L=\mathsf{x}\wedge\mathsf{y}=\sum_{i=1}^{3}\sum_{j=1}^{3}x_iy_j\mathbf{e}_i\wedge\mathbf{e}_j \tag{A.7}$$

applying the definition (A.4) and (A.5) the exterior product becomes

$$L=\mathsf{x}\wedge\mathsf{y}=\sum_{i,j\in I}(x_iy_j-x_jy_i)\,\mathbf{e}_i\wedge\mathbf{e}_j \tag{A.8}$$

obtaining the homogeneous vector the $\mathbf{L}\in\mathbb{R}^6$ defined in a basis $\mathbf{e}_0\wedge\mathbf{e}_1,\mathbf{e}_0\wedge\mathbf{e}_2,\mathbf{e}_0\wedge\mathbf{e}_3,\mathbf{e}_2\wedge\mathbf{e}_3,\mathbf{e}_3\wedge\mathbf{e}_1,\mathbf{e}_1\wedge\mathbf{e}_2$

$$L=\mathsf{x}\wedge\mathsf{y}\ =[\mathbf{e}_0\wedge\mathbf{e}_1,\mathbf{e}_0\wedge\mathbf{e}_2,\mathbf{e}_0\wedge\mathbf{e}_3,\mathbf{e}_2\wedge\mathbf{e}_3,\mathbf{e}_3\wedge\mathbf{e}_1,\mathbf{e}_1\wedge\mathbf{e}_2]\,\mathbf{L} \tag{A.9}$$

defined by $\mathbf{L}=(l_{01},l_{02},l_{03},l_{23},l_{31},l_{12})^\mathsf{T}$ where $l_{ij}=x_iy_j-x_jy_i$.

The vector $\mathbf{L}$ is decomposed in two vectors of $\mathbb{R}^3$, $\mathbf{l}=(l_{01},l_{02},l_{03})^\mathsf{T}$ and $\bar{\mathbf{l}}=(l_{23},l_{31},l_{12})^\mathsf{T}$ such that $\mathbf{L}=\left(\mathbf{l}^\mathsf{T},\bar{\mathbf{l}}^\mathsf{T}\right)^\mathsf{T}$ [1]. Instead of using the tensor l_ij the definition can be expressed in terms of the cross product and matrix operations

[1] Notation detail: The vectors $\mathbf{l}$ and $\bar{\mathbf{l}}$ are defined by the tensor l_{ij} however in some parts of this thesis we refer to the components of this vectors using this notation; $\mathbf{l}=(l_1,l_2,l_3)^\mathsf{T}$ and $\bar{\mathbf{l}}=\left(\bar{l}_1,\bar{l}_2,\bar{l}_3\right)^\mathsf{T}$.

$$
\begin{aligned}
L = \mathsf{x} \wedge \mathsf{y} &= (x_0\mathbf{e}_0 + [\mathbf{e}_1,\mathbf{e}_2,\mathbf{e}_3]\,\mathbf{x}) \wedge (y_0\mathbf{e}_0 + [\mathbf{e}_1,\mathbf{e}_2,\mathbf{e}_3]\,\mathbf{y}) \\
&= x_0\mathbf{e}_0 \wedge [\mathbf{e}_1,\mathbf{e}_2,\mathbf{e}_3]\,\mathbf{y} + [\mathbf{e}_1,\mathbf{e}_2,\mathbf{e}_3]\,\mathbf{x} \wedge y_0\mathbf{e}_0 + [\mathbf{e}_1,\mathbf{e}_2,\mathbf{e}_3]\,\mathbf{x} \wedge [\mathbf{e}_1,\mathbf{e}_2,\mathbf{e}_3]\,\mathbf{y} \\
&= \mathbf{e}_0 \wedge [\mathbf{e}_1,\mathbf{e}_2,\mathbf{e}_3]\,(x_0\mathbf{y} - y_0\mathbf{x}) + [\mathbf{e}_2 \wedge \mathbf{e}_3, \mathbf{e}_3 \wedge \mathbf{e}_1, \mathbf{e}_1 \wedge \mathbf{e}_2]\,\mathbf{x} \times \mathbf{y}
\end{aligned} \tag{A.10}
$$

defining the span operation in terms of the coordinates as

$$
\mathbf{L} = \begin{pmatrix} \mathbf{l} \\ \bar{\mathbf{l}} \end{pmatrix} = \begin{pmatrix} x_0\mathbf{y} - y_0\mathbf{x} \\ \mathbf{x} \times \mathbf{y} \end{pmatrix} \tag{A.11}
$$

Notice that not all the elements of the $\mathbb{P}^5$ space correspond to 3D lines. The points of $\mathbb{P}^5$ corresponding to lines in $\mathbb{P}^3$ must hold $\mathbf{l}^{\mathsf{T}}\bar{\mathbf{l}} = 0$ which is known as Plücker identity. This identity is a quadratic constraint which defines a two dimensional subspace in $\mathbb{P}^5$ called the Klein quadric M_2^4. Plücker representation has a geometric interpretation in Euclidean geometry. Vector $\mathbf{l}$ represents the direction of $\mathbf{L}$ and $\bar{\mathbf{l}}$ is the moment vector which can be seen as the normal to a plane passing through the 3D line and the origin of the reference system. The Plücker identity expresses the orthogonality of $\mathbf{l}$ and $\bar{\mathbf{l}}$ given that the direction vector must be contained in the plane defined by $\bar{\mathbf{l}}$. The minimum distance from the origin O to the 3D line can be computed as $d_l = \frac{\|\bar{\mathbf{l}}\|}{\|\mathbf{l}\|}$. The Euclidean vector from the origin O to the point of the 3D line with minimum distance is $\mathbf{x}_l = \frac{\mathbf{l}\times\bar{\mathbf{l}}}{\|\mathbf{l}\|}$. Notice that $\left[\mathbf{x}_l, \mathbf{l}, \bar{\mathbf{l}}\right]$ defines an orthogonal basis on the Euclidean space. When $\mathbf{x}_l$ is known the line can be defined as

$$
\mathbf{L} = \begin{pmatrix} \mathbf{l} \\ \mathbf{x}_l \times \mathbf{l} \end{pmatrix} \tag{A.12}
$$

A.2.1 Line equation

A point $\mathsf{x} \in \mathbb{P}^3$ is contained in the line $L = \mathsf{y} \wedge \mathsf{z}$ if and only if $\mathsf{x} \wedge y \wedge z = 0$,namely

$$
\mathsf{x} \wedge L = 0 \tag{A.13}
$$

which represents the equation of the line L.

Consider the matrix $\mathsf{E} = [\mathbf{e}_1,\mathbf{e}_2,\mathbf{e}_3]$ representing the basis of the original Euclidean space and the the matrix $\bar{\mathsf{E}} = [\mathbf{e}_2 \wedge \mathbf{e}_3, \mathbf{e}_3 \wedge \mathbf{e}_1, \mathbf{e}_1 \wedge \mathbf{e}_2]$ representing the momentum space, then

$$
L = \mathbf{e}_0 \wedge [\mathbf{e}_1,\mathbf{e}_2,\mathbf{e}_3]\,\mathbf{l} + [\mathbf{e}_2 \wedge \mathbf{e}_3, \mathbf{e}_3 \wedge \mathbf{e}_1, \mathbf{e}_1 \wedge \mathbf{e}_2]\,\bar{\mathbf{l}} = \mathbf{e}_0 \wedge \mathsf{E}\mathbf{l} + \bar{\mathsf{E}}\bar{\mathbf{l}} = \mathbf{e}_0 \wedge \mathsf{l} - \bar{\mathsf{l}}^* \tag{A.14}
$$

where $\mathsf{l} = \mathsf{E}\mathbf{l} = [\mathbf{e}_1,\mathbf{e}_2,\mathbf{e}_3]\,\mathbf{l}$ is the direction vector in the euclidean space and $\bar{\mathsf{l}}^* = -\bar{\mathsf{E}}\bar{\mathbf{l}} = -[\mathbf{e}_2 \wedge \mathbf{e}_3, \mathbf{e}_3 \wedge \mathbf{e}_1, \mathbf{e}_1 \wedge \mathbf{e}_2]\,\bar{\mathbf{l}}$ is the moment vector in the "dual" space, such that

$$\bar{\mathsf{l}}^* = -\bar{\mathsf{l}}\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3 \tag{A.15}$$

where $\Pi_\infty = \mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3$ is a trivector representing the volume element and $\bar{\mathsf{l}} = [\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3]\bar{\mathbf{l}}$ is the moment vector in the Euclidean space. Therefore, the equation of the line is

$$\mathsf{x} \wedge L = (x_0\mathbf{e}_0 + [\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3]\,\mathbf{x}) \wedge \left(\mathbf{e}_0 \wedge [\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3]\,\mathbf{l} + [\mathbf{e}_2 \wedge \mathbf{e}_3, \mathbf{e}_3 \wedge \mathbf{e}_1, \mathbf{e}_1 \wedge \mathbf{e}_2]\,\bar{\mathbf{l}}\right) \tag{A.16}$$

$$\begin{aligned} \mathsf{x} \wedge L = (x_0\mathbf{e}_0 + \mathsf{E}\mathbf{x}) \wedge \left(\mathbf{e}_0 \wedge \mathsf{l} - \bar{\mathsf{l}}^*\right) &= -x_0\mathbf{e}_0 \wedge \bar{\mathsf{l}}^* + \mathsf{E}\mathbf{x} \wedge x_0\mathbf{e}_0 \wedge \mathsf{l} - \mathsf{E}\mathbf{x} \wedge \bar{\mathsf{l}}^* \\ &= -x_0\mathbf{e}_0 \wedge \left(\mathsf{E}\mathbf{x} \wedge \mathsf{l} + \bar{\mathsf{l}}^*\right) - \mathsf{E}\mathbf{x} \wedge \bar{\mathsf{l}}^* \end{aligned} \tag{A.17}$$

so the equation $\mathsf{x} \wedge L = 0$ is equivalent to

$$\mathsf{E}\mathbf{x} \wedge \mathsf{l} + \bar{\mathsf{l}}^* = 0, \quad \mathsf{E}\mathbf{x} \wedge \bar{\mathsf{l}}^* = 0 \tag{A.18}$$

which expressed in terms of the cross product and matrix operations is

$$\mathbf{x} \times \mathbf{l} - x_0\bar{\mathbf{l}} = 0, \quad \mathbf{x}^\mathsf{T}\bar{\mathbf{l}} = 0\,. \tag{A.19}$$

The line equation defines an Euclidean distance from point to line such that

$$d = \frac{\|\mathbf{x} \times \mathbf{l} - x_0\bar{\mathbf{l}}\|}{x_0\|\mathbf{l}\|}\,. \tag{A.20}$$

A.2.2 Change of reference

Consider a projective transformation $\mathsf{H} = \begin{pmatrix} \mathsf{A} & \mathbf{t} \\ \mathbf{o}^\mathsf{T} & s \end{pmatrix}$ with $A = [\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3]$ such that $\mathbf{X} = \mathsf{H}\mathbf{X}'$. The equivalent transformation for Plucker lines is the matrix G such that

$$\mathbf{L} = \mathsf{G}\mathbf{L}' \tag{A.21}$$

where $\mathsf{G} = \begin{pmatrix} \mathsf{G}_{11} & \mathsf{G}_12 \\ \mathsf{G}_{21} & \mathsf{G}_2 \end{pmatrix}$ with

$$\mathsf{G}_{11} = s\mathsf{A} - \mathbf{t}\mathbf{o}^\mathsf{T} \tag{A.22}$$

$$\mathsf{G}_{12} = \mathsf{A}[\mathbf{o}]_\times{}^\mathsf{T} \tag{A.23}$$

$$\mathsf{G}_{21} = [\mathbf{t}]_\times \mathsf{A} \tag{A.24}$$

$$\mathsf{G}_{22} = (\mathbf{a}_2 \times \mathbf{a}_3, \mathbf{a}_3 \times \mathbf{a}_1, \mathbf{a}_1 \times \mathbf{a}_2) \tag{A.25}$$

If H is a Euclidean transformation then

$$\mathsf{G} = \begin{pmatrix} \mathsf{R} & \mathbf{0} \\ [\mathbf{t}]_\times \mathsf{R} & \mathsf{R} \end{pmatrix} \tag{A.26}$$

A.2.3 The side operator

Given two 3D lines expressed in homogeneous Plücker coordinates $\mathbf{L} = \left(\mathbf{l}^\mathsf{T}, \bar{\mathbf{l}}^\mathsf{T}\right)^\mathsf{T}$ and $\mathbf{M} = \left(\mathbf{m}^\mathsf{T}, \bar{\mathbf{m}}^\mathsf{T}\right)^\mathsf{T}$ the side operator between them is defined as

$$\Omega(\mathbf{L}, \mathbf{M}) = \mathbf{l}^\mathsf{T}\bar{\mathbf{m}} + \mathbf{m}^\mathsf{T}\bar{\mathbf{l}} \tag{A.27}$$

The operator Ω is a signed distance whom sign defines the side of the line M with respect the line L. Consider a plane Π defined by the two direction vectors $\mathbf{l}$ and $\mathbf{m}$ but passing by line L and splitting the space in two regions, then $sign(\Omega(\mathbf{L}, \mathbf{M}))$ determines if the line M is passing through the right side or the left side of this plane Π.

The norm of the side operator is $\|\Omega(\mathbf{L}, \mathbf{M})\| = d\|\mathbf{l}\|\|\mathbf{m}\||\sin\alpha|$ where α the angle between $\mathbf{l}$ and $\mathbf{m}$ and d is the Euclidean distance between the closest point of both lines which can defined using the side operator as

$$d = \frac{|\mathbf{l}^\mathsf{T}\bar{\mathbf{m}} + \mathbf{m}^\mathsf{T}\bar{\mathbf{l}}|}{\|\mathbf{l} \times \mathbf{m}\|} \,. \tag{A.28}$$

A.3 Representation of planes

Consider three points at x, y, z, the corresponding plane defined by these points Π is the exterior product of the three homogeneous points

$$\Pi = \mathsf{x} \wedge \mathsf{y} \wedge \mathsf{z} \,. \tag{A.29}$$

The result of the product of three vectors of R^4 is an element of ΛR^4 with the basis

$$\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3, \quad \mathbf{e}_0 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3, \quad \mathbf{e}_0 \wedge \mathbf{e}_3 \wedge \mathbf{e}_1, \quad \mathbf{e}_0 \wedge \mathbf{e}_1 \wedge \mathbf{e}_2 \tag{A.30}$$

and described with the homogeneous vector $\mathbf{U} \in P^3$ such that $\mathbf{U} = \left(\mathbf{u}^\mathsf{T}, u_0\right)^\mathsf{T}$ by

$$\Pi = \mathbf{e}_0 \wedge [\mathbf{e}_2 \wedge \mathbf{e}_3, \mathbf{e}_3 \wedge \mathbf{e}_1, \mathbf{e}_1 \wedge \mathbf{e}_2]\,\mathbf{u} - [\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3]\,u_0 = \left[e_0 \wedge \bar{\mathsf{E}}, -\Pi_\infty\right]\mathbf{U} \tag{A.31}$$

where

$$\mathbf{u} = x_0\mathbf{y} \times z + y_0\mathbf{z} \times \mathbf{x} + z_0\mathbf{x} \times \mathbf{y} \tag{A.32}$$

$$u_0 = -|\mathbf{x}, \mathbf{y}, \mathbf{z}| \,. \tag{A.33}$$

A.3.1 Equation of a plane

A point x is on the plane $\Pi = \mathsf{p}_1 \wedge \mathsf{p}_2 \wedge \mathsf{p}_3$ if and only if $\mathsf{x} \wedge \mathsf{p}_1 \wedge \mathsf{p}_2 \wedge \mathsf{p}_3 = 0$, namely

$$\mathsf{x} \wedge \Pi = 0 \,. \tag{A.34}$$

Consider $\mathsf{u} = [\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3]\,\mathbf{u}$ and the dual $\mathsf{u}^* = -\mathsf{u}\mathbf{e}_1 \wedge \mathbf{e}_2 \wedge \mathbf{e}_3 = -\mathsf{u}\Pi_\infty$, then equation (A.31) can be written as

$$\Pi = -\mathbf{e}_0 \wedge \mathsf{u}^* - u_0 \Pi_\infty \tag{A.35}$$

and the exterior product of four points can be expanded

$$\begin{aligned} \mathsf{x} \wedge \Pi = (x_0\mathbf{e}_0 + \mathsf{E}\mathbf{x}) \wedge (-\mathbf{e}_0 \wedge \mathsf{u}^* - u_0\Pi_\infty) = -\mathsf{E}\mathbf{x} \wedge x_0\mathbf{e}_0 \wedge \mathsf{u}^* - x_0u_0\mathbf{e}_0 \wedge \Pi_\infty \\ = \mathbf{e}_0 \wedge (\mathsf{E}\mathbf{x} \wedge \mathsf{u}^* - x_0u_0\Pi_\infty) = -\left(x_0u_0 + \mathbf{u}^\mathsf{T}\mathbf{x}\right) e_0 \wedge \Pi_\infty \end{aligned} \tag{A.36}$$

such that equation (A.34) becomes

$$u_0x_0 + \mathbf{u}^\mathsf{T}\mathbf{x} = \mathbf{U}^\mathsf{T}\mathbf{X} = 0 \,. \tag{A.37}$$

The plane equation allows defining an Euclidean distance between point and plane such that

$$d = \frac{\|\mathbf{u}^\mathsf{T}\mathbf{x} + u_0x_0\|}{x_0\|u\|} \,. \tag{A.38}$$

A.4 Dual representation: span and intersection

The dual representation of a line $L = \mathbf{e}_0 \wedge \mathsf{l} - \bar{\mathsf{l}}^*$ is the expression $L^* = -\mathbf{e}_0 \wedge \bar{\mathsf{l}} + \mathsf{l}^*$ with where the corresponding coordinates are

$$\mathbf{L}^* = \begin{pmatrix} \mathbf{l}^* \\ \bar{\mathbf{l}}^* \end{pmatrix} \,. \tag{A.39}$$

The line L spanned by points x and y, $L = \mathsf{x} \cup \mathsf{y} = \mathsf{x} \wedge \mathsf{y}$ is given by

$$\mathbf{L} = \begin{pmatrix} \mathbf{l} \\ \bar{\mathbf{l}} \end{pmatrix} = \begin{pmatrix} x_0\mathbf{y} - y_0\mathbf{x} \\ \mathbf{x} \times \mathbf{y} \end{pmatrix} \tag{A.40}$$

Dual to (A.40), the dual Plücker coordinates of the line L contained on planes Π and Υ and represented by vectors $\mathbf{U}$ and $\mathbf{V}$ is $\mathbf{L}^{*\mathsf{T}} = \left((u_0\mathbf{v} - v_0\mathbf{u})^\mathsf{T}, (\mathbf{u} \times \mathbf{v})^\mathsf{T}\right)^\mathsf{T}$. The intersection $L = \Pi \cap \Upsilon$ between the planes Π and Υ represented by vectors $\mathbf{U}$ and $\mathbf{V}$ is

$$\mathbf{L} = \begin{pmatrix} \mathbf{l} \\ \bar{\mathbf{l}} \end{pmatrix} = \begin{pmatrix} \mathbf{u} \times \mathbf{v} \\ u_0\mathbf{v} - v_0\mathbf{u} \end{pmatrix} \,. \tag{A.41}$$

The span of a line L and a point x non incident to the line is a plane Π , such that $\Pi = \mathsf{x} \cup L = \mathsf{x} \wedge L$. The coordinates $\mathbf{U}$ of plane Π are given by

$$\mathbf{U} = \begin{pmatrix} \mathbf{u} \\ u_0 \end{pmatrix} = \begin{pmatrix} -x_0\bar{\mathbf{l}} + \mathbf{x} \times \mathbf{l} \\ \mathbf{x}^\mathsf{T}\bar{\mathbf{l}} \end{pmatrix} \tag{A.42}$$

The dual operation computes the intersection point x with coordinates $\mathbf{X}$ between the plane Π and the line L such that $\mathsf{x} = \Pi \cap L$ obtaining

$$\mathbf{X} = \begin{pmatrix} \mathbf{x} \\ x_0 \end{pmatrix} = \begin{pmatrix} -u_0\mathbf{l} + \mathbf{u} \times \bar{\mathbf{l}} \\ \mathbf{u}^\mathsf{T}\mathbf{l} \end{pmatrix} \tag{A.43}$$

If two lines L and M intersect such that $\Omega(\mathbf{L}, \mathbf{M}) = 0$ the spanned plane Π with coordinates $\mathbf{U}$ such that $\Pi = L \cup M$ is

$$\mathbf{U} = \begin{pmatrix} \mathbf{u} \\ u_0 \end{pmatrix} = \begin{pmatrix} -\mathbf{l} \times \mathbf{m} \\ \bar{\mathbf{l}}^\mathsf{T}\mathbf{m} \end{pmatrix} \tag{A.44}$$

The dual version of this expression computes the intersection point such that $\mathsf{x} = L \cap M$ by

$$\mathbf{X} = \begin{pmatrix} \mathbf{x} \\ x_0 \end{pmatrix} = \begin{pmatrix} -\bar{\mathbf{l}} \times \bar{\mathbf{m}} \\ \mathbf{l}^\mathsf{T}\bar{\mathbf{m}} \end{pmatrix} \tag{A.45}$$

when $\Omega(\mathbf{L}, \mathbf{M}) = 0$.

A.5 Plücker matrices

The antisymmetric rules of the exterior product operation (see Section A.1) can be explicitly imposed defining the antisymmetric matrix

$$\mathsf{N} = ab^\mathsf{T} - ba^\mathsf{T} \ . \tag{A.46}$$

Consider two points of euclidean space $\mathbf{x}, \mathbf{y} \in \mathbb{R}^3$ and the cross-product $\mathbf{z} = \mathbf{x} \times \mathbf{y}$. The exterior product of $\mathbf{x}$ and $\mathbf{y}$ is related with the cross product by

$$\mathsf{x} \wedge \mathsf{y} = [\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3]\mathbf{x} \wedge [\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3]\mathbf{y} \wedge = [\mathbf{e}_2 \wedge \mathbf{e}_3, \mathbf{e}_3 \wedge \mathbf{e}_1, \mathbf{e}_1 \wedge \mathbf{e}_2]\mathbf{z} \tag{A.47}$$

Notice that cross product and exterior production are not exactly the same because the exterior product is a bivector and the cross product is a vector of the Euclidean space but the computing of the components is equivalent. The antisymmetric matrix defined by equation (A.46)

$$\mathsf{N} = \mathbf{x}\mathbf{y}^\mathsf{T} - \mathbf{y}\mathbf{x}^\mathsf{T} = \begin{pmatrix} 0 & z_3 & -z_2 \\ -z_3 & 0 & z_1 \\ z_2 & -z_1 & 0 \end{pmatrix} \tag{A.48}$$

contains the coordinates of the exterior product (or cross-product).

Consider two points in $\mathbb{P}^3$ with homogeneous coordinates $\mathbf{X} \in \mathbb{R}^4$ and $\mathbf{Y} \in \mathbb{R}^4$. The result of imposing the antisymmetric rules (equations (A.4) and (A.5)) of the exterior product using (A.46) is known as Plücker matrix and its relation with plucker coordinates is

$$\mathsf{L} = \mathbf{X}\mathbf{Y}^\mathsf{T} - \mathbf{Y}\mathbf{X}^\mathsf{T} = \begin{pmatrix} 0 & l_{12} & -l_{31} & -l_{01} \\ -l_{12} & 0 & l_{23} & -l_{02} \\ l_{31} & -l_{23} & 0 & -l_{03} \\ l_{01} & l_{02} & l_{03} & 0 \end{pmatrix} = \begin{pmatrix} -[\bar{\mathbf{l}}]_\times & -\mathbf{l} \\ \mathbf{l}^\mathsf{T} & 0 \end{pmatrix} \quad \text{(A.49)}$$

where $[\mathbf{x}]_\times = \begin{pmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{pmatrix}$ is the antisymmetric operator (do not confuse with (A.48)). Notice that in general equation (A.46) is a way of computing the exterior product between two vectors using matrix operations but the result is a matrix instead of a bivector.

The advantage of this description is that allows easily computing the meet operation between a line and plane. Given a line L represented by the Plücker matrix L and the plane Π represented by the vector $\mathbf{U}$ the point x with homogeneous coordinates $\mathbf{X}$ resulting from the intersection between line and plane $\mathsf{x} = L \cap \pi$ is the pole of L and it is computed by

$$\mathbf{X} = \mathsf{L}\mathbf{U} \,. \quad \text{(A.50)}$$

The duality between points and planes allows using (A.46) with the vectors $\mathbf{U}$ and $\mathbf{V}$, which represent the planes Π and Υ, obtaining the dual Plücker matrix

$$\mathsf{L}^* = \mathbf{U}\mathbf{V}^\mathsf{T} - \mathbf{V}\mathbf{U}^\mathsf{T} = \begin{pmatrix} 0 & l_{03} & -l_{02} & -l_{23} \\ -l_{03} & 0 & l_{01} & -l_{31} \\ l_{02} & -l_{03} & 0 & -l_{12} \\ l_{23} & l_{31} & l_{12} & 0 \end{pmatrix} = \begin{pmatrix} -[\mathbf{l}]_\times & -\bar{\mathbf{l}} \\ \bar{\mathbf{l}}^\mathsf{T} & 0 \end{pmatrix} . \quad \text{(A.51)}$$

With the dual Plücker matrix it is easy to define the span operation. Given a line L represented by the dual Plücker matrix L^* and the point x represented by the vector $\mathbf{X}$ the plane Π with homogeneous coordinates $\mathbf{U}$ resulting from the span between line and plane $\mathsf{x} = L \cap \pi$ is the polar plane of L and it is computed by

$$\mathbf{U} = \mathsf{L}^*\mathbf{X} \,. \quad \text{(A.52)}$$

A.6 Quadrics, polarities, null-polarities and complexes

In the Cayley homogeneous space polarities are described by symmetric matrices and represent quadrics. Null polarities are described by anti-symmetric matrices and represent complexes.

A.6.1 Quadrics

If Q is a symmetric matrix then the endomorphism defined by $\mathbf{U} = \mathsf{Q}\mathbf{X}$ is called a polarity. In particular, given a matrix $\mathsf{Q} \in \mathbb{R}^{4\times4}$ such that $\mathsf{Q}^{\mathsf{T}} = \mathsf{Q}$ and an homogeneous point of the projective Cayley-Grassmann space $\mathbf{X} \in \mathbb{P}^3$ the equation

$$\mathbf{X}^{\mathsf{T}}\mathsf{Q}\mathbf{X} = 0 \tag{A.53}$$

represents a quadric in $\mathbb{P}^3$. A quadric defines a polarity between a point and a plane. The plane $\mathbf{U} = \mathsf{Q}\mathbf{X}$ is the polar plane of $\mathbf{X}$ with respect to Q.

The dual of a non-degenerate quadric represented by the full rank matrix, Q is another quadric $\mathsf{Q}^* = \mathsf{adjoint}\,(Q) = \det(\mathsf{Q})\,\mathsf{Q}^{-\mathsf{T}}$ which defining the equation of the tangent planes $\mathbf{U}$ to the quadric Q such that

$$\mathbf{U}^{\mathsf{T}}\mathsf{Q}^*\mathbf{U} = 0\,. \tag{A.54}$$

The point $\mathbf{X} = \mathsf{Q}^*\mathbf{U}$ is the pole plane of $\mathbf{U}$ with respect to Q. Notice that, since $\mathsf{Q}^{\mathsf{T}} = \mathsf{Q}$, then $\mathsf{Q}^{-\mathsf{T}} = \mathsf{Q}^{-1}$ and due to $\mathbf{X}$ is up to scale then $\mathbf{X} = \mathsf{Q}^{-1}\mathbf{U}$.

Under the point transformation $\mathbf{X} = \mathsf{H}\mathbf{X}'$ between two reference systems a quadric transforms as

$$\mathsf{Q} = \mathsf{H}^{-\mathsf{T}}\mathsf{Q}'\mathsf{H}^{-1}\,. \tag{A.55}$$

A.6.2 Intersection of a line with a quadric

Consider two points $\mathbf{X}_0$ and $\mathbf{X}_1$, the corresponding parametric line definition

$$\mathbf{X}(\lambda) = \mathbf{X}_0 + \lambda\mathbf{X}_1 \tag{A.56}$$

and the corresponding Plücker matrix such that $\mathsf{L} = \mathbf{X}_0{\mathbf{X}_1}^{\mathsf{T}} - \mathbf{X}_1{\mathbf{X}_0}^{\mathsf{T}}$.

The plane $\mathbf{U}_0 = \mathsf{Q}X_0$ is the plane polar to the quadric Q respect to the pole $\mathbf{X}_0$ which is a plane intersecting the quadric such that any tangent plane to the quadric passing through a point of this intersection contains the pole $\mathbf{X}_0$. Consider the point $\mathbf{X}_u$ which is the intersection between the line with the polar plane $\mathbf{U}_0$. It is called the conjugate point of $\mathbf{X}_0$ and it can be computed as

$$\mathbf{X}_u = \mathsf{L}\mathbf{U}_0 = \mathsf{L}\mathsf{Q}\mathbf{X}_0\,, \tag{A.57}$$

and we can express the line in terms of $\mathbf{X}_0$ and $\mathbf{X}_u$

$$\mathbf{X}(\lambda) = \mathbf{X}_u + \lambda \mathbf{X}_0 \,. \tag{A.58}$$

The line intersects the quadric in two points $\mathbf{X}_I^+$ and $\mathbf{X}_I^-$ which are two different real points if the ray intersects the quadric, one double point if the ray is tangent to the quadric and two complex points if not.

Consider the polar plane $\mathbf{U}$ of a generic point $\mathbf{X}(\lambda)$ of the line

$$\mathbf{U} = \mathsf{Q}\mathbf{X}(\lambda) = \mathsf{Q}\mathbf{X}_u + \lambda \mathsf{Q}\mathbf{X}_0 \tag{A.59}$$

If the point $\mathbf{X}(\lambda)$ is in the quadric its corresponding polar plane $\mathbf{U}$ is tangent to the quadric and passing through the pole.

Imposing the point $\mathbf{X}(\lambda)$ lying on plane $\mathbf{U}$ we can solve the intersection.

$$\mathbf{X}^\mathsf{T}(\lambda)\mathbf{U} = \mathbf{X}_u{}^\mathsf{T}\mathsf{Q}\mathbf{X}_u + \lambda\left(\mathbf{X}_u{}^\mathsf{T}\mathsf{Q}\mathbf{X}_0 + \mathbf{X}_0{}^\mathsf{T}\mathsf{Q}\mathbf{X}_u\right) + \lambda^2\mathbf{X}_0{}^\mathsf{T}\mathsf{Q}\mathbf{X}_0 = 0 \tag{A.60}$$

Since $\mathbf{X}_0$ and $\mathbf{X}_u$ are conjugate points $\mathbf{X}_u{}^\mathsf{T}\mathsf{Q}\mathbf{X}_0 = 0$ and $\mathbf{X}_0{}^\mathsf{T}\mathsf{Q}\mathbf{X}_u = 0$ and the last equation remains

$$\mathbf{X}^\mathsf{T}(\lambda)\mathbf{U} = \mathbf{X}_u{}^\mathsf{T}\mathsf{Q}\mathbf{X}_u + \lambda^2\mathbf{X}_0{}^\mathsf{T}\mathsf{Q}\mathbf{X}_0 = 0 \tag{A.61}$$

From these we obtain the parameter λ and the corresponding expressions for the intersections and for the polar planes in these points (which are tangent to these).

$$\lambda^2 = -\frac{\mathbf{X}_u{}^\mathsf{T}\mathsf{Q}P_u}{\mathbf{X}_0{}^\mathsf{T}\mathsf{Q}\mathbf{X}_0} = -\frac{\mathbf{X}_0{}^\mathsf{T}\mathsf{Q}\mathsf{L}^\mathsf{T}\mathsf{Q}\mathsf{L}\mathsf{Q}\mathbf{X}_0}{\mathbf{X}_0{}^\mathsf{T}\mathsf{Q}\mathbf{X}_0} \tag{A.62}$$

$$\mathbf{X}_I^\pm = (\mathsf{L}\mathsf{Q} + \lambda I)\,\mathbf{X}_0 \tag{A.63}$$

$$\mathbf{U}_I^\pm = (\mathsf{Q}\mathsf{L} + \lambda I)\,\mathsf{Q}\mathbf{X}_0 \tag{A.64}$$

A.6.3 Null-Polarities and Complexes

If N is a antisymmetric matrix then the endomorphism defined by $\mathbf{U} = \mathsf{N}\mathbf{X}$ is called a null-polarity. In particular, given a matrix $\mathsf{N} \in \mathbb{R}^{4\times4}$ such that $\mathsf{N}^\mathsf{T} = -\mathsf{N}$ and an homogeneous point of the projective Cayley-Grassmann space $\mathbf{X} \in \mathbb{P}^3$ the equation

$$\mathbf{X}^\mathsf{T}\mathsf{N}\mathbf{X} = 0 \tag{A.65}$$

means that $\mathbf{X}$ is contained in the plane $\mathsf{N}\mathbf{X}$ and N represents a complex. If following a notation similar to the one used in Plücker matrices

$$\mathsf{N} = \begin{pmatrix} 0 & n_{03} & -n_{02} & -n_{23} \\ -n_{03} & 0 & n_{01} & -n_{31} \\ n_{02} & -n_{03} & 0 & -n_{12} \\ n_{23} & n_{31} & n_{12} & 0 \end{pmatrix} = \begin{pmatrix} -[\bar{\mathbf{n}}]_\times & -\mathbf{n} \\ \mathbf{n}^\mathsf{T} & 0 \end{pmatrix} \tag{A.66}$$

where $\bar{\mathbf{n}} = (n_{01}, n_{02}, n_{03})^\mathsf{T}$ and $\mathbf{n} = (n_{23}, n_{31}, n_{12})^\mathsf{T}$, if the relation $\mathbf{n}^\mathsf{T}\ \bar{\mathbf{n}} = 0$ is satisfied the complex is called a singular complex.

Bibliography

[Achtelik et al., 2011] Achtelik, M., Achtelik, M., Weiss, S., and Siegwart, R. (2011). Onboard imu and monocular vision based control for mavs in unknown in-and outdoor environments. In *IEEE international conference on Robotics and automation (ICRA 2011)*, pages 3056–3063.

[Adelson and Wang, 1992] Adelson, E. H. and Wang, J. Y. A. (1992). Single lens stereo with a plenoptic camera. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (2):99–106.

[Agrawal and Ramalingam, 2013] Agrawal, A. and Ramalingam, S. (2013). Single image calibration of multi-axial imaging systems. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2013)*, pages 1399–1406.

[Agrawal et al., 2010] Agrawal, A., Taguchi, Y., and Ramalingam, S. (2010). Analytical forward projection for axial non-central dioptric and catadioptric cameras. In *European Conference on Computer Vision (ECCV 2010)*, pages 129–143.

[Agrawal et al., 2011] Agrawal, A., Taguchi, Y., and Ramalingam, S. (2011). Beyond alhazen's problem: Analytical projection model for non-central catadioptric cameras with quadric mirrors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011)*, pages 2993–3000.

[Alemán-Flores et al., 2014] Alemán-Flores, M., Alvarez, L., Gomez, L., and Santana-Cedrés, D. (2014). Line detection in images showing significant lens distortion and application to distortion correction. *Pattern Recognition Letters*, 36:261–271.

[Alvarez et al., 2009] Alvarez, L., Gómez, L., and Sendra, J. (2009). An algebraic approach to lens distortion by line rectification. *Journal of Mathematical Imaging and Vision*, 35(1):36–50.

[Antone and Teller, 2000] Antone, M. and Teller, S. (2000). Automatic recovery of relative camera rotations for urban scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2000)*, volume 2, pages 282 –289.

[Baker and Nayar, 1999] Baker, S. and Nayar, S. K. (1999). A theory of single-viewpoint catadioptric image formation. *International Journal of Computer Vision*, 35(2):175–196.

[Baker and Nayar, 2001] Baker, S. and Nayar, S. K. (2001). *Single viewpoint catadioptric cameras*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[Barreto, 2003] Barreto, J. (2003). *General Central Projection Systems: Modeling, Calibration and Visual Servoing*. PhD thesis.

[Barreto and Araujo, 2001] Barreto, J. P. and Araujo, H. (2001). Issues on the geometry of central catadioptric image formation. *IEEE Conference On Computer Vision and Pattern Recognition (CVPR 2001)*, 2:422.

[Barreto and Araujo, 2005] Barreto, J. P. and Araujo, H. (2005). Geometric properties of central catadioptric line images and their application in calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1327–1333.

[Barreto and Araujo, 2006a] Barreto, J. P. and Araujo, H. (2006a). Fitting conics to paracatadioptric projections of lines. *Computer Vision and Image Understanding*, 101(3):151 – 165.

[Barreto and Araujo, 2006b] Barreto, J. P. and Araujo, H. (2006b). Fitting conics to paracatadioptric projections of lines. *Computer Vision and Image Understanding*, 101(3):151–165.

[Bay et al., 2006] Bay, H., Tuytelaars, T., and Van Gool, L. J. (2006). SURF: Speeded up robust features. In *9th European Conference on Computer Vision (ECCV 2006)*.

[Bazin et al., 2007] Bazin, J. C., Demonceaux, C., and Vasseur, P. (2007). Fast central catadioptric line extraction. In *IbPRIA '07: Proceedings of the 3rd Iberian conference on Pattern Recognition and Image Analysis, Part II*, pages 25–32.

[Bazin et al., 2010] Bazin, J. C., Demonceaux, C., Vasseur, P., and Kweon, I. (2010). Motion estimation by decoupling rotation and translation in catadioptric vision. *Computer Vision and Image Understanding*, 114(2):254–273.

[Bermudez et al., 2010] Bermudez, J., Puig, L., and Guerrero, J. J. (2010). Line extraction in central hyper-catadioptric systems. *OMNIVIS - 10th Workshop on Omnidirectional Vision, Camera Networks and Nonclassical Cameras , pp8:1-7,June, 2010*.

[Bermudez-Cameo et al., 2015a] Bermudez-Cameo, J., , Lopez-Nicolas, G., and Guerrero, J. (2015a). Fitting line projections in non-central catadioptric omnidirectional systems. Technical report.

[Bermudez-Cameo et al., 2014a] Bermudez-Cameo, J., Barreto, J. P., Lopez-Nicolas, G., and Guerrero, J. J. (2014a). Minimal solution for computing pairs of lines in non-central cameras. In *The 12th Asian Conference on Computer Vision (ACCV 2014). Lecture Notes in Computer Science*, volume 9003, pages 585–597.

[Bermudez-Cameo et al., 2015b] Bermudez-Cameo, J., Demonceaux, C., Lopez-Nicolas, G., and J.J., G. (2015b). Minimal solution for line parallel to a plane in non-central systems. Technical report.

[Bermudez-Cameo et al., 2015c] Bermudez-Cameo, J., Lopez-Nicolas, G., and Guerrero, J. (2015c). Automatic line extraction in uncalibrated omnidirectional cameras with revolution symmetry. *International Journal of Computer Vision*, 114(1):16–37.

[Bermudez-Cameo et al., 2012a] Bermudez-Cameo, J., Lopez-Nicolas, G., and Guerrero, J. J. (2012a). A unified framework for line extraction in dioptric and catadioptric cameras. In *11th Asian Conference on Computer Vision (ACCV 2012). Lecture Notes in Computer Science.*, volume 7727.

[Bermudez-Cameo et al., 2013] Bermudez-Cameo, J., Lopez-Nicolas, G., and Guerrero, J. J. (2013). Line extraction in uncalibrated central images with revolution symmetry. In *24th British Machine Vision Conference (BMVC 2013).*

[Bermudez-Cameo et al., 2014b] Bermudez-Cameo, J., Lopez-Nicolas, G., and Guerrero, J. J. (2014b). Line-images in cone mirror catadioptric systems. In *22th International Conference on Pattern Recognition (ICPR 2014)*, pages 2083–2088.

[Bermudez-Cameo et al., 2012b] Bermudez-Cameo, J., Puig, L., and Guerrero, J. J. (2012b). Hypercatadioptric line images for 3D orientation and image rectification. *Robotics and Autonomous Systems*, 60(6):755–768.

[Bermudez-Cameo et al., 2015d] Bermudez-Cameo, J., Saurer, O., Lopez-Nicolas, G., Guerrero, J., and Pollefeys, M. (2015d). Exploiting line metric reconstruction from non-central circular panoramas. Technical report.

[Bosse et al., 2002] Bosse, M., Rikoski, R., Leonard, J., and Teller, S. (2002). Vanishing points and 3D lines from omnidirectional video. In *International Conference on Image Processing (ICIP 2002)*, volume 3, pages 513 – 516.

[Brown, 1971] Brown, D. (1971). Close-range camera calibration. *Photogrammetric engineering*, 37(8):855–866.

[Bukhari and Dailey, 2013] Bukhari, F. and Dailey, M. N. (2013). Automatic radial distortion estimation from a single image. *Journal of Mathematical Imaging and Vision*, 45(1):31–45.

[Caglioti and Gasparini, 2005] Caglioti, V. and Gasparini, S. (2005). On the localization of straight lines in 3D space from single 2D images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, volume 1, pages 1129–1134.

[Caglioti et al., 2007a] Caglioti, V., Gasparini, S., and Taddei, P. (2007a). Methods for space line localization from single catadioptric images: new proposals and comparisons. In *IEEE 11th International Conference on Computer Vision (ICCV 2007)*, pages 1–6.

[Caglioti et al., 2007b] Caglioti, V., Taddei, P., Boracchi, G., Gasparini, S., and Giusti, A. (2007b). Single-image calibration of off-axis catadioptric cameras using lines. In *11th International Conference on Computer Vision (ICCV 2007)*, pages 1–6.

[Carrasco et al., 2012] Carrasco, J.-P., de la Escalera, A. d. l. E., Armingol, J. M., et al. (2012). Recognition stage for a speed supervisor based on road sign detection. *Sensors*, 12(9):12153–12168.

[Chen et al., 2011] Chen, W., Cheng, I., Xiong, Z., Basu, A., and Zhang, M. (2011). A 2-point algorithm for 3D reconstruction of horizontal lines from a single omnidirectional image. *Pattern Recognition Letters*, 32(3):524–531.

[Chum and Matas, 2005] Chum, O. and Matas, J. (2005). Matching with prosac-progressive sample consensus. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, volume 1, pages 220–226.

[Courbon et al., 2007] Courbon, J., Mezouar, Y., Eck, L., and Martinet, P. (2007). A generic fisheye camera model for robotic applications. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007)*, pages 1683–1688.

[Cucchiara et al., 2003] Cucchiara, R., Grana, C., Prati, A., and Vezzani, R. (2003). A hough transform-based method for radial lens distortion correction. In *12th International Conference on Image Analysis and Processing (ICIAP 2003)*, pages 182–187.

[Dalal and Triggs, 2005] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, volume 1, pages 886–893.

[Davison et al., 2007] Davison, A. J., Reid, I. D., Molton, N. D., and Stasse, O. (2007). Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067.

[Devernay and Faugeras, 2001] Devernay, F. and Faugeras, O. (2001). Straight lines have to be straight. *Machine Vision and Applications*, 13(1):14–24.

[Fischler and Bolles, 1981] Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.

[Fitzgibbon, 2001] Fitzgibbon, A. W. (2001). Simultaneous linear estimation of multiple view geometry and lens distortion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, volume 1, pages I–125.

[Gasparini and Caglioti, 2011] Gasparini, S. and Caglioti, V. (2011). Line localization from single catadioptric images. *International journal of computer vision*, 94(3):361–374.

[Gasparini et al., 2009] Gasparini, S., Sturm, P., and Barreto, J. P. (2009). Plane-based calibration of central catadioptric cameras. In *IEEE 12th International Conference on Computer Vision (ICCV 2009)*, pages 1195–1202.

[Geyer and Daniilidis, 2000] Geyer, C. and Daniilidis, K. (2000). A unifying theory for central panoramic systems and practical applications. In *6th European Conference on Computer Vision, (ECCV 2000)*, volume 2, pages 445–461.

[Geyer and Daniilidis, 2001] Geyer, C. and Daniilidis, K. (2001). Catadioptric projective geometry. *International Journal of Computer Vision*, 45(3):223–243.

[Gonçalves, 2010] Gonçalves, N. (2010). On the reflection point where light reflects to a known destination on quadratic surfaces. *Optics letters*, 35(2):100–102.

[Gonçalves and Nogueira, 2009] Gonçalves, N. and Nogueira, A. C. (2009). Projection through quadric mirrors made faster. In *IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops 2009)*, pages 2141–2148.

[Griffiths and Harris, 2011] Griffiths, P. and Harris, J. (2011). *Principles of algebraic geometry*, volume 52. John Wiley & Sons.

[Guerrero et al., 2008] Guerrero, J. J., Murillo, A. C., and Sagües, C. (2008). Localization and matching using the planar trifocal tensor with bearing-only data. *IEEE Transactions on Robotics*, 24(2):494–501.

[Gupta and Hartley, 1997] Gupta, R. and Hartley, R. I. (1997). Linear pushbroom cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(9):963–975.

[Hartley and Zisserman, 2000] Hartley, R. I. and Zisserman, A. (2000). *Multiple View Geometry in Computer Vision*.

[Kanatani, 2015] Kanatani, K. (2015). *Understanding geometric algebra: Hamilton, Grassmann and Clifford for Computer Vision and Graphics*.

[Kannala and Brandt, 2006] Kannala, J. and Brandt, S. (2006). A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8):1335–1340.

[Kannala et al., 2008a] Kannala, J., Brandt, S. S., and Heikkilä, J. (2008a). Self-calibration of central cameras by minimizing angular error. In *3rd International Conference on Computer Vision Theory and Applications (VISAPP 2008)*, pages 28–35.

[Kannala et al., 2008b] Kannala, J., Heikkilä, J., and Brandt, S. S. (2008b). Geometric camera calibration. *Wiley Encyclopedia of Computer Science and Engineering*.

[Kim et al., 2008] Kim, J.-H., Li, H., and Hartley, R. (2008). Motion estimation for multi-camera systems using global optimization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, pages 1–8.

[Kingslake, 1989] Kingslake, R. (1989). *A history of the photographic lens*. Academic Press.

[Klein and Murray, 2007] Klein, G. and Murray, D. (2007). Parallel tracking and mapping for small ar workspaces. In *6th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2007)*, pages 225–234.

[Kneebone et al., 1998] Kneebone, G. et al. (1998). *Algebraic projective geometry*. Oxford University Press, USA.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

[Kümmerle et al., 2011] Kümmerle, R., Grisetti, G., Strasdat, H., Konolige, K., and Burgard, W. (2011). g 2 o: A general framework for graph optimization. In *IEEE International Conference on Robotics and Automation (ICRA 2011)*, pages 3607–3613.

[Lanman et al., 2006] Lanman, D., Wachs, M., Taubin, G., and Cukierman, F. (2006). Reconstructing a 3D line from a single catadioptric image. In *Third International Symposium on 3D Data Processing, Visualization, and Transmission*, pages 89–96.

[Lee et al., 2009] Lee, D. C., Hebert, M., and Kanade, T. (2009). Geometric reasoning for single image structure recovery. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, pages 2136–2143.

[Lee et al., 2015] Lee, G. H., Li, B., Pollefeys, M., and Fraundorfer, F. (2015). Minimal solutions for the multi-camera pose estimation problem. *The International Journal of Robotics Research*.

[López-Nicolás et al., 2014] López-Nicolás, G., Omedes, J., and Guerrero, J. (2014). Spatial layout recovery from a single omnidirectional image and its matching-free sequential propagation. *Robotics and Autonomous Systems*, 62(9):1271–1281.

[López-Nicolás and Sagüés, 2010] López-Nicolás, G. and Sagüés, C. (2010). Catadioptric camera model with conic mirror. In *British Machine Vision Conference (BMVC 2010)*.

[López-Nicolás and Sagüés, 2014] López-Nicolás, G. and Sagüés, C. (2014). Unitary torus model for conical mirror based catadioptric system. *Computer Vision and Image Understanding*, 126:67–79.

[Lowe, 2004] Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, volume 20, pages 91–110.

[Lumsdaine and Georgiev, 2009] Lumsdaine, A. and Georgiev, T. (2009). The focused plenoptic camera. In *IEEE International Conference on Computational Photography (ICCP 2009)*, pages 1–8.

[Magnier et al., 2010] Magnier, B., Comby, F., Strauss, O., Triboulet, J., and Demonceaux, C. (2010). Highly specific pose estimation with a catadioptric omnidirectional camera. In *In IEEE Int. Conf. on Imaging Systems and Techniques (IST'10)*, Thessaloniki, Grece.

[Mahony et al., 2012] Mahony, R., Kumar, V., and Corke, P. (2012). Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. *IEEE Robotics & amp amp Automation Magazine*, (19):20–32.

[Mei and Malis, 2006] Mei, C. and Malis, E. (2006). Fast central catadioptric line extraction, estimation, tracking and structure from motion. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2006)*, pages 4774–4779.

[Mei and Rives, 2007] Mei, C. and Rives, P. (2007). Single viewpoint omnidirectional camera calibration from planar grids. In *IEEE International Conference on Robotics and Automation (ICRA 2007)*, pages 3945–3950.

[Meier et al., 2012] Meier, L., Tanskanen, P., Heng, L., Lee, G. H., Fraundorfer, F., and Pollefeys, M. (2012). Pixhawk: A micro aerial vehicle design for autonomous flight using onboard computer vision. *Autonomous Robots*, 33(1-2):21–39.

[Melo et al., 2013] Melo, R., Antunes, M., Barreto, J., Falco, G., and Gonalves, N. (2013). Unsupervised intrinsic calibration from a single frame using a "plumb-line" approach. In *IEEE 14th International Conference on Computer Vision (ICCV 2013)*, pages 1–6.

[Menem and Pajdla, 2004] Menem, M. and Pajdla, T. (2004). Constraints on perspective images and circular panoramas. In *5th British Machine Vision Conference (BMVC 2004)*, pages 1–10.

[Mezouar et al., 2004] Mezouar, Y., Abdelkader, H., Martinet, P., and Chaumette, F. (2004). Central catadioptric visual servoing from 3D straight lines. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, volume 1, pages 343–348.

[Miraldo and Araujo, 2014] Miraldo, P. and Araujo, H. (2014). Planar pose estimation for general cameras using known 3d lines. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)*, pages 4234–4240.

[Miraldo et al., 2015] Miraldo, P., Araujo, H., and Goncalves, N. (2015). Pose estimation for general cameras using lines. *IEEE Transactions on Cybernetics*, (99):1–1.

[Newcombe et al., 2011] Newcombe, R. A., Lovegrove, S. J., and Davison, A. J. (2011). Dtam: Dense tracking and mapping in real-time. In *IEEE International Conference on Computer Vision (ICCV 2011)*, pages 2320–2327.

[Ng et al., 2005] Ng, R., Levoy, M., Brédif, M., Duval, G., Horowitz, M., and Hanrahan, P. (2005). Light field photography with a hand-held plenoptic camera. *Computer Science Technical Report CSTR*, 2(11).

[of Vision Pty. Ltd. (2004),] of Vision Pty. Ltd. (2004), P. Persistence of vision raytracer (version 3.6).

[Perdigoto and Araujo, 2012] Perdigoto, L. and Araujo, H. (2012). Reconstruction of 3D lines from a single axial catadioptric image using cross-ratio. In *21th International Conference on Pattern Recognition (ICPR 2012)*, pages 857–860.

[Pinciroli et al., 2005] Pinciroli, C., Bonarini, A., and Matteucci, M. (2005). Robust detection of 3D scene horizontal and vertical lines in conical catadioptric sensors. In *Proc. 6th Workshop on Omnidirectional Vision*.

[Pottmann and Wallner, 2001] Pottmann, H. and Wallner, J. (2001). *Computational line geometry*. Springer.

[Puig et al., 2011] Puig, L., Bastanlar, Y., Sturm, P., Guerrero, J. J., and Barreto, J. (2011). Calibration of central catadioptric cameras using a dlt-like approach. *International Journal of Computer Vision*, 93(1):101–114.

[Puig et al., 2010] Puig, L., Bermudez, J., and Guerrero, J. J. (2010). Self-orientation of a hand-held catadioptric system in man-made environments. *IEEE International Conference on Robotics and Automation (ICRA 2010) pp:2549-2555, Anchorage, Alaska May*.

[Puig et al., 2012] Puig, L., Bermudez-Cameo, J., Sturm, P., and Guerrero, J. J. (2012). Calibration of omnidirectional cameras in practice. a comparison of methods. *Computer Vision and Image Understanding*, 116:120–137.

[Raguram et al., 2013] Raguram, R., Chum, O., Pollefeys, M., Matas, J., and Frahm, J. (2013). Usac: a universal framework for random sample consensus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):2022–2038.

[Ramalingam and Brand, 2013] Ramalingam, S. and Brand, M. (2013). Lifting 3D manhattan lines from a single image. In *IEEE International Conference on Computer Vision (ICCV 2013)*, pages 497–504.

[Ray, 2002] Ray, S. (2002). *Applied photographic optics: Lenses and optical systems for photography, film, video, electronic and digital imaging*. Focal Press.

[Ros et al., 2015] Ros, G., Ramos, S., Granados, M., Bakhtiary, A., Vazquez, D., and Lopez, A. M. (2015). Vision-based offline-online perception paradigm for autonomous driving. In *IEEE Winter Conference on Applications of Computer Vision (WACV 2015)*, pages 231–238.

[Rosten and Loveland, 2011] Rosten, E. and Loveland, R. (2011). Camera distortion self-calibration using the plumb-line constraint and minimal hough entropy. *Machine Vision and Applications*, 22(1):77–85.

[Rublee et al., 2011] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). Orb: an efficient alternative to sift or surf. In *IEEE International Conference on Computer Vision (ICCV 2011)*, pages 2564–2571.

[Scaramuzza et al., 2006] Scaramuzza, D., Martinelli, A., and Siegwart, R. (2006). A toolbox for easily calibrating omnidirectional cameras. In *International Conference on Ingelligent Robots and Systems (IROS 2006)*.

[Scaramuzza et al., 2009] Scaramuzza, D., Martinelli, A., and Siegwart, R. (2009). A robust descriptor for tracking vertical lines in omnidirectional images and its use in mobile robotics. *International Journal of Robotics Research*, 28(2):149–171. Special Issue on Field and Service Robotics.

[Schneider et al., 2009] Schneider, D., Schwalbe, E., and Maas, H.-G. (2009). Validation of geometric models for fisheye lenses. *Journal of Photogrammetry and Remote Sensing*, 64(3):259 – 266.

[Serrano-Gotarredona et al., 2009] Serrano-Gotarredona, R., Oster, M., Lichtsteiner, P., Linares-Barranco, A., Paz-Vicente, R., Gómez-Rodríguez, F., Camuñas-Mesa, L., Berner, R., Rivas-Pérez, M., Delbrück, T., et al. (2009). Caviar: A 45k neuron, 5m synapse, 12g connects/s aer hardware sensory–processing–learning–actuating system for high-speed visual object recognition and tracking. *IEEE Transactions on Neural Networks*, 20(9):1417–1438.

[Stevenson and Fleck, 1996] Stevenson, D. and Fleck, M. (1996). Nonparametric correction of distortion. In *3rd IEEE Workshop on Applications of Computer Vision (WACV 96)*, pages 214–219.

[Strand and Hayman, 2005] Strand, R. and Hayman, E. (2005). Correcting radial distortion by circle fitting. In *16th British Machine Vision Conference (BMVC 2005)*.

[Strelow et al., 2001] Strelow, D., Mishler, J., Koes, D., and Singh, S. (2001). Precise omnidirectional camera calibration. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2001).*, volume 1, pages I–689.

[Sturm and Gargallo, 2007] Sturm, P. and Gargallo, P. (2007). Conic fitting using the geometric distance. In *Asian Conference on Computer Vision (ACCV 2007)*.

[Sturm et al., 2011] Sturm, P., Ramalingam, S., Tardif, J.-P., Gasparini, S., and Barreto, J. (2011). Camera models and fundamental concepts used in geometric computer vision. *Foundations and Trends in Computer Graphics and Vision*, 6(1–2):1–183.

[Svoboda and Pajdla, 2002] Svoboda, T. and Pajdla, T. (2002). Epipolar geometry for central catadioptric cameras. *International Journal of Computer Vision*, 49(1):23–37.

[Swaminathan et al., 2006] Swaminathan, R., Grossberg, M. D., and Nayar, S. K. (2006). Non-single viewpoint catadioptric cameras: Geometry and analysis. *International Journal of Computer Vision*, 66(3):211–229.

[Swaminathan and Nayar, 2000] Swaminathan, R. and Nayar, S. K. (2000). Nonmetric calibration of wide-angle lenses and polycameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1172–1178.

[Swaminathan et al., 2008] Swaminathan, R., Wu, A., Dong, H., et al. (2008). Depth from distortions. In *The 8th Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras-OMNIVIS*.

[Tardif et al., 2006] Tardif, J., Sturm, P., and Roy, S. (2006). Self-calibration of a general radially symmetric distortion model. *9th European Conference on Computer Vision (ECCV 2006)*, pages 186–199.

[Teller and Hohmeyer, 1999] Teller, S. and Hohmeyer, M. (1999). Determining the lines through four lines. *Journal of graphics tools*, 4(3):11–22.

[Thormählen et al., 2003] Thormählen, T., Broszio, H., and Wassermann, I. (2003). Robust line-based calibration of lens distortion from a single view. In *Proceedings of MIRAGE*, pages 105–112.

[Torralba et al., 2008] Torralba, A., Fergus, R., and Freeman, W. T. (2008). 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1958–1970.

[Vasseur and Mouaddib, 2004] Vasseur, P. and Mouaddib, E. M. (2004). Central catadioptric line detection. In *British Machine Vision Conference*.

[Viola and Jones, 2001] Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, volume 1, pages I–511.

[Wang et al., 2009] Wang, A., Qiu, T., and Shao, L. (2009). A simple method of radial distortion correction with centre of distortion estimation. *Journal of Mathematical Imaging and Vision*, 35(3):165–172.

[Wu and Hu, 2005] Wu, Y. and Hu, Z. (2005). Geometric invariants and applications under catadioptric camera model. In *10th IEEE International Conference on Computer Vision (ICCV 2005)*, volume 2, pages 1547–1554.

[Ying and Hu, 2004a] Ying, X. and Hu, Z. (2004a). Can we consider central catadioptric cameras and fisheye cameras within a unified imaging model? In *8th European Conference on Computer Vision (ECCV 2204)*.

[Ying and Hu, 2004b] Ying, X. and Hu, Z. (2004b). Catadioptric camera calibration using geometric invariants. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1260–1271.

[Ying and Hu, 2004c] Ying, X. and Hu, Z. (2004c). Catadioptric line features detection using hough transform. In *17th International Conference on Pattern Recognition (ICPR 2004)*, volume 4, pages 839–842.

[Ying and Zha, 2005] Ying, X. and Zha, H. (2005). Simultaneously calibrating catadioptric camera and detecting line features using hough transform. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)*, pages 412–417.

[Zhang et al., 2014] Zhang, Q., Shen, X., Xu, L., and Jia, J. (2014). Rolling guidance filter. In *13th European Conference on Computer Vision (ECCV 2014).*, pages 815–830.

[Ziegler et al., 2014] Ziegler, J., Bender, P., Schreiber, M., Lategahn, H., Strauss, T., Stiller, C., Dang, T., Franke, U., Appenrodt, N., Keller, C. G., et al. (2014). Making bertha drive–an autonomous journey on a historic route. *Intelligent Transportation Systems Magazine, IEEE*, 6(2):8–20.