UNIVERSITE DE BOURGOGNE

U.F.R. SCIENCES ET TECHNIQUES

THÈSE

Pour obtenir le grade de

Docteur de l'Université de Bourgogne

Spécialité : Instrumentation et Informatique de l'Image

par

Saleh Mosaddegh

le 17 Octobre 2011

Two View Line-Based Matching, Motion Estimation and Reconstruction for Central Imaging Systems

Directeur de thèse:

Professeur David Fofi (Le2i, Université de Bourgogne)

Co-directeur de thèse:

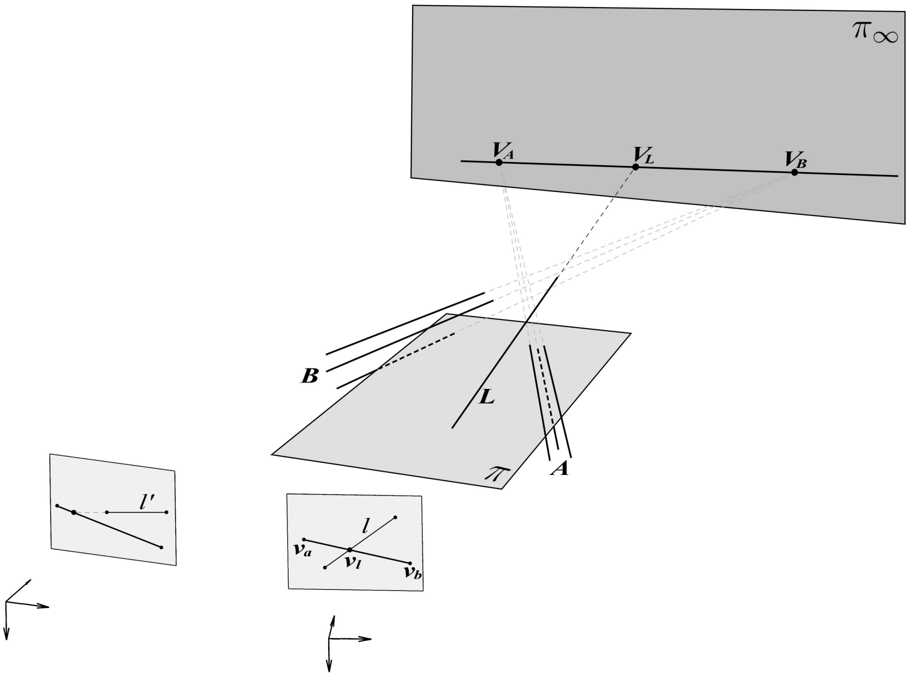
Professeur Pascal Vasseur (LITIS, INSA Rouen)

Jury:

Lacroix, Simon	Directeur de recherches, LAAS, CNRS	Rapporteur
Morin, Luce	Professeur, IETR, INSA Rennes	Rapporteur
Truchetet, Fredéric	Professeur, Le2i, Université de Bourgogne	Examinateur
Habed, Adlane	MCF, Le2i, Université de Bourgogne	Examinateur

TWO VIEW LINE-BASED MATCHING, MOTION ESTIMATION AND RECONSTRUCTION FOR CENTRAL IMAGING SYSTEMS

SALEH MOSADDEGH



A thesis submitted for the degree of Doctor of Philosophy of the University of Bourgogne

SUPERVISORS:
Prof. David Fofi
Prof. Pascal Vasseur

Le2i
Computer Vision Departement
Universite de Bourgogne

Le Creusot, France
July 2011

Saleh Mosaddegh: *Two View Line-based Matching, Motion Estimation and Reconstruction for Central Imaging Systems* , © July 2011

SUPERVISORS:
Prof. David Fofi
Prof. Pascal Vasseur

Le Creusot, France

Dedicated to pictures because
“A picture is worth a thousand words”!

ABSTRACT

The primary goal of this thesis is to develop generic motion and structure algorithms for images taken from constructed scenes by various types of central imaging systems including perspective, fish-eye and catadioptric systems. Assuming that the mapping between the image pixels and their 3D rays in space is known, instead of image planes, we work on image spheres (projection of the images on a unit sphere) which enable us to present points over the entire view sphere suitable for presenting omnidirectional images.

In the first part of this thesis, we develop a generic and simple line matching approach for images taken from constructed scenes under a short baseline motion as well as a fast and original geometric constraint for matching lines in planar constructed scenes insensible to the motion of the camera for all types of central images including omnidirectional images.

Next, we introduce a unique and efficient way of computing overlap between two segments on perspective images which considerably decreases the overall computational time of a segment-based motion estimation and reconstruction algorithm. Finally in last part of this thesis, we develop a simple motion estimation and surface reconstruction algorithm for piecewise planar scenes applicable to all kinds of central images which uses only two images and is based on minimum line correspondences.

To demonstrate the performance of these algorithms we experiment with various real images taken by a simple perspective camera, a fish-eye lens, and two different kinds of paracatadioptric sensors, the first one is a folded catadioptric camera and the second one is a classic paracatadioptric system composed of a parabolic mirror in front of a telecentric lens.

RÉSUMÉ

L'objectif principal de cette thèse est de développer des algorithmes génériques d'estimation du mouvement et de la structure à partir d'images de scènes prises par différents types de systèmes d'acquisition centrale : caméra perspective, fish-eye et systèmes catadioptriques, notamment. En supposant que la correspondance entre les pixels de l'image et les lignes de vue dans l'espace est connue, nous travaillons sur des images sphériques, plutôt que sur des images planes (projection des images sur la sphère unitaire), ce qui nous permet de considérer des points sur une vue mieux adaptée aux images omnidirectionnelles et d'utiliser un modèle générique valable pour tous les capteurs centraux.

Dans la première partie de cette thèse, nous développons une approche générique de mise en correspondance simple de lignes à partir d'images de scènes urbaines ou péri-urbaines sous la contrainte d'un faible déplacement du capteur, ainsi qu'une contrainte rapide et originale pour apparier des lignes d'un environnement plan par morceaux, indépendante du mouvement de la caméra centrale.

Ensuite, nous introduisons une méthode unique et efficace pour estimer le recouvrement entre deux segments sur des images perspectives, diminuant considérablement le temps global de calcul par rapport aux algorithmes connus. Enfin, dans la dernière partie de cette thèse, nous développons un algorithme d'estimation du mouvement et de reconstruction de surfaces pour les scènes planes par morceaux applicable à toutes sortes d'images centrales, à partir de deux vues uniquement et ne nécessitant qu'un nombre minime de correspondances de ligne.

Pour demontrer les performances de ces algorithmes, nous les avons expérimentés avec diverses images réelles acquises à partir d'une caméra perspective, une lentille fish-eye, et deux différents types de capteurs paracatadioptriques (l'un est composé d'un miroir simple, et l'autre d'un miroir double).

KEYWORDS: Structure and Motion, central omnidirectional image, catadioptric, line segments matching, wide baseline, constructed scene.

PUBLICATIONS

Most ideas and some figures have appeared previously in the following publications:

A. S. Mosaddegh, D. Fofi, P. Vasseur, "Fast Line Matching between Uncalibrated Disparate Views of Planar Surfaces", Twelfth IAPR Conference on Machine Vision Applications (MVA), June 2011.

B. S. Mosaddegh, D. Fofi, P. Vasseur, "Line Segment Based Structure and Motion from Two Views: a Practical Issue", 6th International Conference on Computer Vision Theory and Applications (VISIGRAPP), March 2011.

C. S. Mosaddegh, A. Fazlollahi, D. Fofi, P. Vasseur, "Line segment based structure and motion from two views", IS&T/SPIE Electronic Imaging - Image Processing: Machine Vision Applications IV, January 2011.

D. S. Mosaddegh, D. Fofi, P. Vasseur, "Line Based Motion Estimation and Reconstruction of Piece-Wise Planar Scenes", IEEE Computer Society's Workshop on Applications of Computer Vision (WACV), January 2011.

E. S. Mosaddegh, D. Fofi, P. Vasseur, "Motion Estimation and Reconstruction of Piecewise Planar Scenes from Two Views", 25th International Conference of Image and Vision Computing (IVCNZ), November 2010.

F. S. Mosaddegh, D. Fofi, P. Vasseur, "Ego-translation estimation from one straight edge in constructed scenes", IS&T/SPIE Electronic Imaging - Image Processing: Machine Vision Applications II, January 2010.

G. S. Mosaddegh, D. Fofi, P. Vasseur, "A Generic Method of Line Matching for Central Imaging Systems under Short-Baseline Motion", Springer-Verlag LNCS, Vol. 5716, Image Analysis and Processing (ICIAP), pp. 939-948, September 2009.

H. S. Mosaddegh, D. Fofi, P. Vasseur, S. Ainouz, "Line Matching across Catadioptric Images under Short Baseline Motion", OMNIVIS in conjunction with ECCV, October 2008.

Other Publications not Forming Part of the Thesis:

A. S. Ainouz, O. Morel, S. Mosaddegh, D. Fofi, "Adapted Processing of Catadioptric Images Using Polarization Imaging", International Conference on Image Processing (ICIP), November 2009.

ACKNOWLEDGMENTS

My first, and most earnest, acknowledgment must go to Professor David Fofi and Professor Pascal Vasseur for their valuable helps and supports in their capacities of my supervisors. I wish to express my deep and sincere thanks to David for providing me with a very pleasant working environment in Le2i, and for his important support throughout this work (If it was not because of me being almost fully rubbed in Sicily! I would have said, with confidence, that Italians are among the most wonderful people in the world!). Equally, I owe my most sincere gratitude to Pascal for his detailed and constructive comments whenever we had the chance to meet. Both of these gentlemen patiently tolerated my passion for international conferences! and lack of interest in journals! and their understanding and encouraging was, I believe, the key to successful determination of this thesis! I would like also to thank Dr. Cedric Demonceaux and Dr. Dro Desire Sidibe for many interesting discussions and Professor Simon Lacroix and Professor Luce Morin, for their generous contribution of time and expertise during the reviewing this thesis.

My sincere thanks are also due to all the people who by different ways are contributing to Le2i, particulièrement, notre secrétaire gentille et sympa, Nathalie! It was a pleasure for me to work with all the wonderful people in our lab. I like to remember the weekends and BBQs and all the people who participated!

A big thank-you goes to my wonderful family for always being there when I needed them most and for their continuous and unconditional support. My mother, my father, my only brother, Hamid, my two sisters Salehe and Safiye and my only nephew, Farzad, never even once complained about how infrequently I visit them and they deserve far more credit than I can ever give them.

Finally, many thanks to Conseil Regional de Bourgogne for providing me with financial support to pursue my academic career in Europe.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

Part I

BACKGROUND

Here we give some backgrounds on material presented in this thesis. We have also given some backgrounds inside each chapter wherever it was more appropriate.

Chapter 1: we briefly talk about structure and motion from lines followed by the objectives, the contributions and the layout of this document.

Chapter 2: Basic pinhole camera and its image formation process, a classification of the existing central imaging systems based on their fabrication technologies and also with respect to the mirror geometry and finally some common calibration techniques are material presented in this chapter.

1 INTRODUCTION

Structure from Motion, or SfM, has been the subject of many researches. The SfM problem, as it is handled by human stereo vision system, was formally investigated by Ullman [136]. He investigated the process by which the visual system constructs descriptions of the surrounding objects in the scene, their 3D shapes and their motions. The structure from motion problem, as defined in computer vision, is a similar problem where the aim is to find the correspondence between images and the reconstruction of 3D scene or objects.

This field can be seen as a collection of tools and techniques for recovering the geometry of 3D scenes from its projection on flat 2D images. The need for estimating unknown internal parameters of the imaging system in addition to the fact that during image formation process, depth information is lost makes this field very challenging and additional information is inevitable in order to solve the reconstruction problem. One way is to exploit prior knowledge about the scene to reduce the complexity of the problem. For example, using only one image from architectural scenes, one can reconstruct simple 3D line segments and planar surfaces using parallelism and coplanarity constraints [123] or recover a 3D texture-mapped architecture model by employing constraints derived from shape symmetries, which are prevalent in architecture [65]. Another possibility is to use more than one image taken from different locations and to find corresponding image features across these images. 3D pre-images of these correspondences then can be reconstructed by triangulation. However, calibration parameters and location with respect to the scene coordinate system of each camera are necessary for triangulation step to work. Assuming enough accurate feature correspondences are established, these unknowns can also be estimated from the correspondences between two or more views.

1.1 STRUCTURE AND MOTION FROM LINES

In computer vision, straight line segments are particularly useful features from images of man-made environments thanks to existence of numerous straight edges in such scenes. Similar to other features of interest, they are also used to perform motion estimation and/or 3D reconstruction between two or multiple views of the same scene. Among these lines, if the scene is man-made, there exist generally two or more groups of parallel lines which define vanishing directions. Vanishing points in the image corresponding to these vanishing directions can be used to derive useful parameters of the imaging system and find feature correspondences between images. This is the approach considered in the first part of this thesis for camera calibration and line matching. In the second part of the thesis, these vanishing points are further exploited to estimate the unknown rotation between two views and facilitate the development of a practical interactive system for recovering motion between two images of piece-wise planar scenes and at the same time reconstructing the surfaces.

Line segments are generally less numerous than interest points but richer in information. Moreover, their detection is very reliable according to their orientation. Most keypoints hardly capture geometrical and structural information of the scene since they are not localized at edges while lines are directly connected to geometrical information about the scene. Despite these advantages, structure and motion using lines is a particular difficult field and only a few works have been proposed in literature.

As for matching stage, besides the traditional challenges in point matching, these difficulties proceed from some other different reasons such as the inaccuracy of the endpoint extraction, fragmented segments, the poor geometric disambiguity constraint, lack of significant photometric information in the local neighborhood of segments and no global geometric constraint such as the epipolar constraint. Up to now, only a few methods are reported in the literature for automatic line segment matching [115, 130, 45, 12, 139, 138].

As for motion estimation stage, these difficulties proceed mainly from the fact that two views of lines are not enough to estimate motion [55, 96, 153, 141, 60, 9, 27]. Once again, besides our proposed algorithm in chapter 7, the algorithm introduced in Zhang [153], are, so far, the only works on motion estimation based on only two views of only line segments.

1.2 THE LINE CORRESPONDENCE PROBLEM

Feature matching is a fundamental problem in computer vision for a wide variety of vision tasks such as image registration, motion estimation, object recognition, etc. It is defined as the task of determining the correspondences between two sets of image features extracted from two or more views of the same scene. To find correspondence between images, the trajectories of features such as corner points (edges with gradients in multiple directions) need to be found from one image to the next. These trajectories over time are then used to reconstruct their 3D positions and the cameras motion. The most used features of interest are generally points and infinite lines, while line segments, contours and regions are rarely exploited.

Line Matching is simply finding the corresponding images of the same 3D line across two or multiple images of a scene. It is often the first step in the reconstruction of scenes such as an urban scene. As argued by Ayache [1], we may prefer to match segments instead of points because:

A. we can reduce the complexity of the matching by reducing the number of matches which should be carried out since there are always less segments than points

b. segments can provide stronger matching constraints since geometric attribute measured on contour segments are more richer and discriminant than points.

c. the position and orientation of a segment are usually extracted more precisely than position and orientation of an isolated point (a point can also have a consistent orientation based on local image properties, refer to SIFT for more details [72]).

However, the problem of matching segments is more complex than points due to the reasons mentioned earlier, especially if only two views are considered. One solution is to identify corresponding segments interactively in each view, having the advantage that surfaces can be defined simultaneously with correspondences, *e.g.* the user can also identify geometric primitives such as 2D rectangles and 3D cubes. However, this interactive approach has the disadvantage that it is time consuming and the accuracy of the resulting motion estimation and reconstruction will depend seriously on how carefully the user positions the image segments.

As long as automatic approaches are concerned, years of research have shown that, in general, the correspondence problem is difficult to solve automatically. Automatic algorithms work by computing some geometric and photometric properties of line segments such as length, orientation, average gray-level intensity along the segment and the coordinates of the midpoint or endpoints based on some assumptions (sometimes very naive) such as the transformation between two images is either a similarity or an affine transformation.

Line matching techniques may be divided into two categories: narrow-baseline and wide-baseline.

1.2.1 *Narrow baseline line matching*

Under the assumption that the change in camera position and orientation with respect to the distance of the camera to the scene is small, the neighborhood of line segments will look similar in two views and simple similarity measuring function such as ZNCC[1] from pixel intensity values sampled from a rectangular window around the line segment can be efficient in matching the segments. Since depth computation is quite sensitive to image coordinate measurement noise for closely spaced viewpoints, structure and motion parameters can not be accurately recovered. However, by considering multiple views and tracking segment correspondences throughout consecutive images of this type, it is possible to recover structure and motion parameters accurately.

[1] Zero-mean Normalized Cross Correlation

1.2.2 *Wide baseline line matching*

When the baseline is large, the segment translation in the two images may be considerable and random, thus narrow-baseline matching algorithms can easily fail and automatically motion and structure estimation becomes much more difficult. Until now, only a few methods for automatic line segment matching for wide baseline stereo exist. These methods are either based on known epipolar geometry or based on some quantities that are invariant under perspective viewing. Even if the epipolar geometry is available, it can not be used on line segments that have a similar orientation as the epipolar lines and, therefore, other properties have to be used for robust wide baseline line segment matching.

1.3 PRINCIPAL OBJECTIVES

In this thesis, our main objective is to develop some tools for motion and structure between two images of lines regardless of the type of the imaging system (perspective, fish-eye or catadioptric) assuming that the mapping between the image pixels and their 3D rays in space is known.

In the first part, we investigate some methods of matching lines with no dependency on the photometric information around the segments or their endpoints, therefore looking for a geometric constraint. Matching lines is already a challenging task in general and in constructed scenes the problem is more challenging due to the ambiguities caused either by large homogeneous regions of texture or repeated patterns in images.

In the second part, assuming the rotation part of the motion between two images can be estimated using matched vanishing points, we investigate how this can be in our advantages for developing a simple algorithm for reconstruction of piece-wise planar scenes from only two views and based on minimum line correspondences.

1.4 KEY CONTRIBUTIONS

The main contributions of the thesis are:

- A generic and simple line matching approach for all types of central images including omnidirectional images in constructed scenes under a short baseline motion.

- A fast and original geometric constraint for matching lines for central images including omnidirectional images in planar constructed scenes insensible to the motion of the camera.

- A unique and efficient way of computing overlap between two segments on perspective images.

- A simple motion estimation and surface reconstruction algorithm for piecewise planar scenes applicable to all kinds of central images including omnidirectional images.

The above two line matching methods are based merely on images of lines on the unitary sphere and no other constraints such as the epipolar geometry are neither known nor used.

1.5 OUTLINE OF DISSERTATION

This work is divided into three parts. In the first part of our work, following this introduction, the work starts by reviewing image formation process for different types of imaging systems followed by a brief examination and classification of large field-of-view (FOV) cameras and some of their examples, the unitary sphere and some common calibration techniques for central imaging systems.

The second and third parts of the thesis divide our main contributions into automatic structure from motion methods and algorithms for constructed scenes especially applicable to omnidirectional images. The conclusion and some remarks on possible future work can be found at the end of each chapter.

Part I - Background

Chapter 1: A brief talk about structure and motion from lines followed by the objectives, the contributions and the layout of this document.
Chapter 2: A classification of the existing central imaging systems based on their fabrication technologies and also with respect to the mirror geometry and some common calibration techniques.

Part II - Line matching for constructed scenes using two views

Chapter 3: A state of the art on various approaches of line matching along with their classification based on the kind of motion which can be handled by each method.
Chapter 4: Line matching across images taken by a central imaging system with focus on short baseline motion of the system is proposed.
Chapter 5: We address the problem of matching randomly oriented lines on a scene plane by finding the intersection of each line on the image plane with the line at infinity of the plane followed by recovering the direction of the line in scene.

Part III - Motion estimation and reconstruction for constructed scenes using two views

Chapter 6: We introduce a unique and efficient way of computing overlap between two segments which considerably decreases the overall computational time of a segment-based motion estimation and reconstruction algorithm.
Chapter 7: And finally, we present an algorithm for reconstruction of piece-wise planar scenes from only two views and based on minimum line correspondences.

A general summary is the last chapter of the manuscript where we put together a summary of the methods plus the contributions and future works.

2 CENTRAL IMAGING SYSTEMS AND THEIR GEOMETRY OF IMAGE FORMATION

Cameras are, generally speaking, the most important elements of each and any image-based computer vision task. They are devices that map points in the real world onto pixels in the image. Therefore, we dedicate this separate chapter to the geometry of how this image formation process occurs for different types of imaging systems. We do so by considering basic pinhole camera and its geometry of forming images followed by a brief examination of large field-of-view (FOV) cameras and some of their examples. We then conclude with the unitary sphere model and some common calibration techniques. Since the focus of our work is central imaging systems therefore we do not investigate non-central imaging systems and their properties and applications.

2.1 PINHOLE CAMERA MODEL

A basic pinhole camera, the simplest camera model, works on the same principal as *camera obscura* built by medieval scientists for investigating the properties of light and optics [97]. A camera obscura is a dark chamber with a small hole in one of its walls and an image screen opposite of it (Figure 1). Early models were large; comprising either a whole darkened room or a tent but later on, more easily portable models became available (Figure 2). Such cameras were later adapted by Nicephore Niepce (1765 – 1833) for creating the first photographs.

Consider the geometry of a pinhole camera as shown in Figure 3. Let the projection centre of the camera, C, coincide with the origin of the world coordinate system, XYZ, and the image plane be at a distance f from C and the centre of the image coordinate system, xyz, coincides with the intersection of the Z_{axis} and the image plane. This intersection point is called principal point.

Following properties of similar triangles, it is easy to show that a point in Euclidean 3-space with coordinates $X = (X, Y, Z)^T$ is mapped to the point $(f\frac{X}{Z}, f\frac{Y}{Z})^T$ on the image plane where a line joining the point X to the centre of projection meets the image plane. This projection maps a 3D point in Euclidean space $\mathbb{R}^3$ to a 2D image point in $\mathbb{R}^2$. The projection can be conveniently represented in homogeneous coordinates by a linear mapping in terms of matrix multiplication:

Figure 1: Camera obscura, from a manuscript of military designs. 17th century. Courtesy of The Library of Congress, Washington (http://loc.gov/)

Figure 2: Joshua Reynolds' camera obsecura (1723 - 1792): (left) open camera and (right) disguised as a book. Courtesy of the Science Museum, London (http://sciencemuseum.org.uk/)

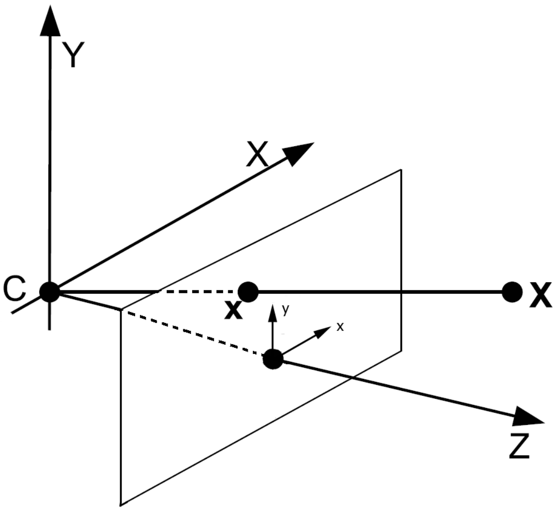


Figure 3: The Pinhole camera geometry.

$$\begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

To derive the above expression, we assumed that the origin of coordinates in the image plane is at the principal point. In practice, it is common to assume the origin of the image coordinate system at one of the corners of the image. In this case, the coordinates of the principal point are not anymore $(0,0)$. We also assumed the camera and world coordinate systems to be the same which is not always the case. These two coordinate systems are generally related via a rotation and a translation. Furthermore, we assumed that the image coordinates are metric coordinates while in practice they are almost always measured in pixels. Relaxing these assumptions and by following [52], the above equation can be rewritten as:

$$x = KR\,[I| - C]\,X$$

where X is now in a world coordinate system, C represents the coordinates of the camera centre in the world coordinate system, and R is a 3 x 3 rotation matrix representing the orientation of the camera coordinate system. K is called calibration matrix and contains five internal camera parameters:

$$K = \begin{pmatrix} \alpha_x & s & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{pmatrix}$$

where α_x and α_y represent the focal lengths of the camera in terms of pixel dimensions in the x and y directions respectively; (u_0, v_0) is the principal point on the image plane and s is the skew parameter. For more details on camera matrix, we refer the reader to Hartley and Zisserman [52] or Faugeras and Luong [28].

2.2 OMNIDIRECTIONAL VISION CAMERAS

An off the-shelf traditional perspective cameras is an advanced variation of the same old pinhole camera designed over a century ago. Soon it was realized that these cameras are too limited and highly restrictive for many of the tasks such as robot navigation, pattern recognition, tracking or surveillance tasks. One place to look for the inspiration for designing new vision sensors is nature. Thanks to the massive computational power offered by the brain, a human being is able to smoothly perform navigation and recognition tasks despite a very limited field of view compared to the other visual organs found in nature which have much less computational capacity for processing visual information (for example insects brain has 10^5 to 10^6 neurons compared to 10^{11} of a human brain [35]). Despite this, we as humans are not capable to build similar view systems and it is logical to assume that the performances of these perfect flying systems are improved by the special construction of their eyes, mainly from a wide field of view given by their compound eyes [99]. This might have inspired the design of some of omnidirectional vision sensors such as the general imaging model proposed by Grossberg and Nayar [48] which is composed of a set of light sensors on a sphere and which is capable of representing any arbitrary imaging system.

Having said that, one can easily conclude that among the vision sensors, the ones that replicate nature models are the most versatile and among the vision sensors inspired by nature the omnidirectional ones are, theoretically, the most suitable for 3D reconstruction and navigation tasks. In the rest of this section we will give a classification of the most common omnidirectional sensors together with a description of their characteristics.

Omnidirectional cameras are commonly classified based on their fabrication technologies, see Figure 4.

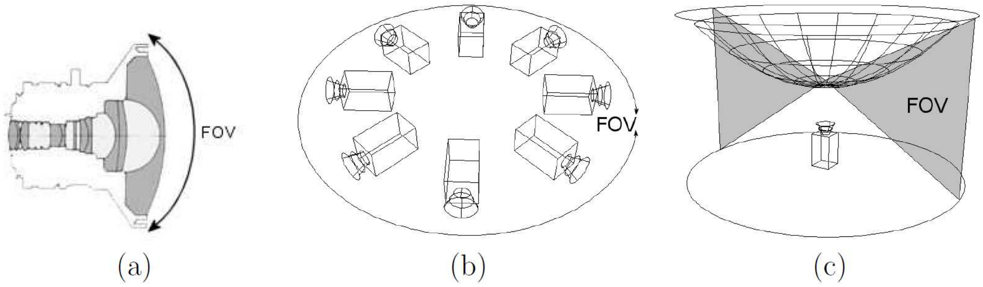


Figure 4: Examples of non-perspective imaging systems. (a) a dioptric wide-angle system such a fish-eye lens, (b) an imaging system made of a camera cluster and (c) a catadioptric system. The images are courtesy of R. Orghidan [99].

2.2.1 *Special lenses*

Special lenses such as fish-eye lenses are imaging systems with a very short focal length which can produce large FOV images (e.g. [10, 104]). However, the modeling process of these cameras is very complicated due to their specific drawbacks such as the radial distortion, the varying resolution (high in the middle, low in peripherals) and the lack of a unique view point. Despite these shortcomings, a fish-eye lens can provide images suitable for many applications and many researchers have studied its projection function and how to remove the distortion and calibrate the lens [29, 144, 20]. A particular configuration of the panoramic annular lens (PAL) with mirror surfaces is also used by Greguss [46] in order to achieve large FOV images.

2.2.2 *Multiple image acquisition systems*

These systems form panoramic images by stitching images taken from:

- a spinning single camera [23, 68, 77, 156] where the camera rotates at constant angular velocity , taking narrow, vertical scan lines from different images and joins them to form a panoramic views;
- or from a single camera attached to a rotating arm[101, 88];
- or from two cameras attached to a rotating plate[17, 18]. An example of their latest configuration is shown in Figure 5;
- or from multiple cameras oriented in different directions [4, 29]. The configuration of several cameras looking in different directions is especially more applicable where a full spherical field of view is not necessary for instance for video-conferencing [25, 29].

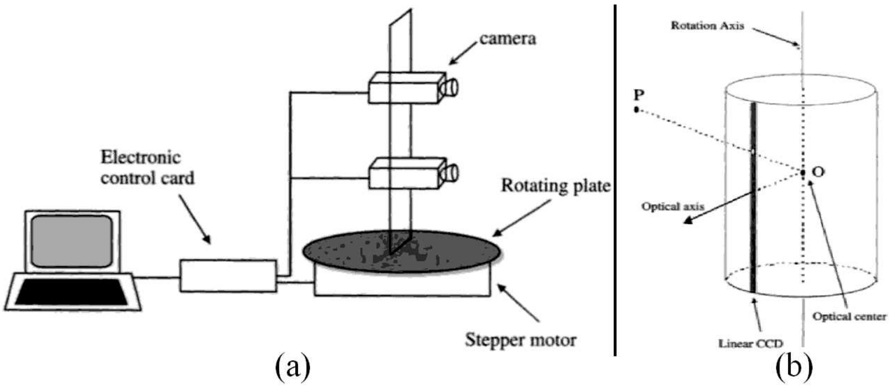


Figure 5: Examples of multiple image acquisition systems proposed by Benosman et al. (a) The panoramic sensor architecture, (b) Vertical scan line CCD camera equivalent.

2.2.3 *Catadioptrics*

Considering the main drawbacks of above two classes of omnidirectional cameras (e.g. complexity of modeling special lenses such as fish-eye lenses or difficult fabrication, setup and calibration of multiple image acquisition systems), the question is whether it is possible to create a simpler and faster imaging system with still a spherical view field? To achieve this goal, Baker and Nayar [5, 93] investigated the incorporation of a dioptric or refractive element with a catoptric or reflective element, i.e. a combination of a lens and a mirror. They refer to this approach as catadioptric image formation. They classified these sensors in two respective categories: central and non-central sensors.

The first group is sensors with a single viewpoint which are made of either parabolic mirror associated to an orthographic camera or hyperbolic, elliptic or planar mirrors placed in front of a perspective camera. Figure 6 shows the entire class of feasible central catadioptric systems. Table 1 provides the equation of the 3D surface assuming the origin of the coordinate system coincident with the focus of the mirror and the z_{axis} is aligned with the mirror axis. The length of latus rectum[1] of the conic mirrors is $4p$.

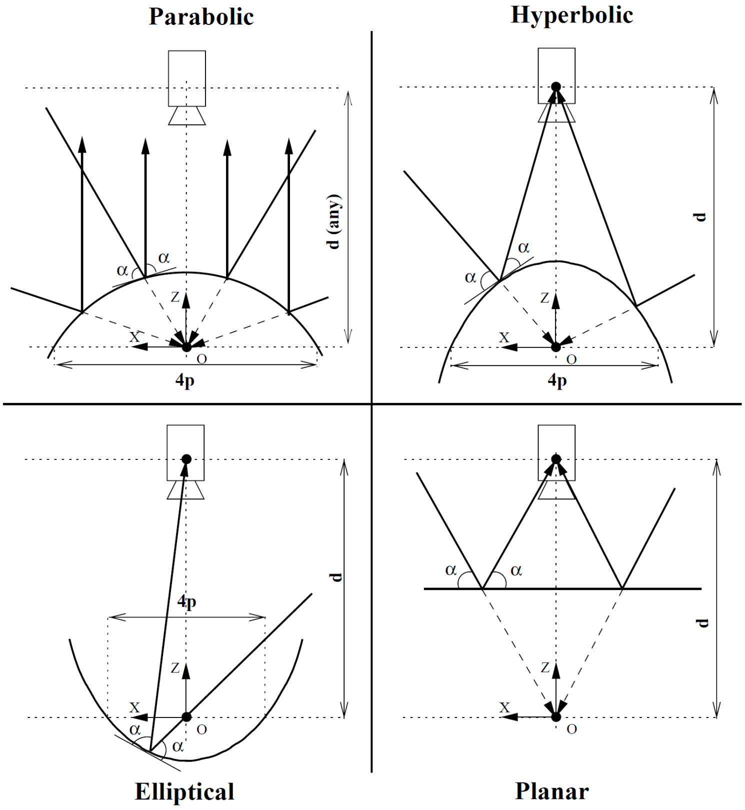


Figure 6: The entire class of feasible central catadioptric systems. Courtesy of Joao P. Barreto

[1] The line segment through a focus of a conic section, perpendicular to the major axis, which has both endpoints on the curve.

Camera	Mirror surface	
Parabolic	$\sqrt{x^2+y^2+z^2} = z+2p$	
Hyperbolic	$\frac{(z+\frac{d}{2})^2}{a^2}$-$\frac{x^2}{b^2}$-$\frac{y^2}{b^2} = 1$	$a = \frac{\sqrt{d^2+4p^2}-2p}{2}$ $b = \sqrt{p(\sqrt{d^2+4p^2}-2p)}$
Ellipse	$\frac{(z+\frac{d}{2})^2}{a^2}+\frac{x^2}{b^2}+\frac{y^2}{b^2} = 1$	$a = \frac{\sqrt{d^2+4p^2}+2p}{2}$ $b = \sqrt{p(\sqrt{d^2+4p^2}+2p)}$
Plane	$z = -\frac{d}{2}$	

Table 1: All central catadioptric mirror equations

The design of these systems ensure that the camera only measures the intensity of light passing through a single point in 3D space which is the projection point. Uniqueness of an effective viewpoint is desirable because it allows the mapping of any part of the scene to a perspective plane without creating any parallax. In this sense, a central catadioptric system has the same effect as a camera rotating about its focus, though without the necessity of rotating the camera (see Figure 7 for an example). The resulting perspective images can be processed by traditional computer vision techniques though their resolution is not as good as a traditional perspective image.

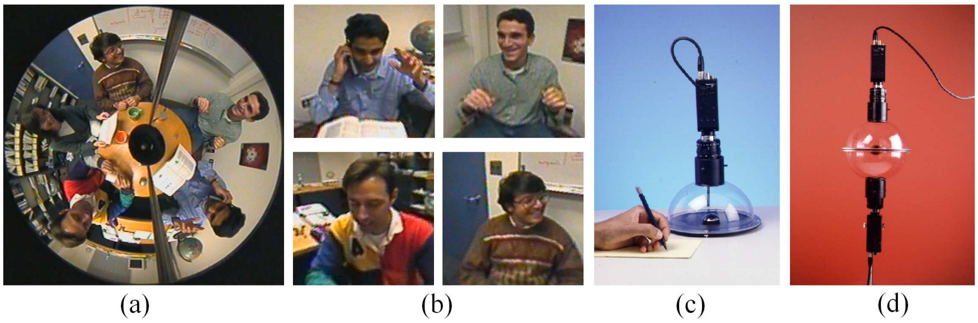


Figure 7: Generation of perspective images (b) from an omnidirectional image (a) made possible thanks to unique effective viewpoint of the system. On the right are two implementations of catadioptric omnidirectional cameras with paraboloidal mirrors: (c) Hemispherical field of view. (d) Full sphere field of view. Courtesy of S.K.Nayar.

Catadioptric systems can be composed of a single camera with a single mirror [146, 102, 119, 22, 86, 26, 70, 148, 149, 140, 98, 93, 41, 122, 155, 56, 142, 38], a single camera with multiple mirrors [95, 42, 43, 94, 87, 120, 31, 32] or multiple cameras with multiple mirrors [91, 67, 58, 116, 121, 145, 147, 44]. The mirror type can be flat, conic, elliptic, hyperbolic, parabolic or spherical among which only flat, hyperbolic and parabolic mirrors exhibit a practical single view point (SVP) property. The conical mirrors have the single view point at the apex of the cone which means that the only rays entering the pinhole are those which gaze the cone and therefore they do not come from the scene. This is a degenerate case and therefore cones can not be used to construct a SVP catadioptric camera. Similarly for a spherical mirror, the viewpoint and pinhole coincide at the center of the sphere which means the observer can only see itself. Even though elliptical mirrors satisfy SVP, their practical application is limited because they decrease the field of view instead of increasing it.

2.3 SVP OMNIDIRECTIONAL SYSTEM EXAMPLES

Since our subjects of interest are central imaging systems, in the next section we will present some representative examples of related works on SVP complying imaging systems. We refer the reader to sources such as [126, 99] for an in depth treatment of the non-SVP catadioptric configurations including catadioptric stereo systems, plenoptic cameras (where a lenticular array is placed in front of a camera's sensor plane forming many tiny pinhole cameras), along with several common characteristics.

2.3.1 *Single camera with single hyperbolic mirror*

HyperOmni, an early prototype of this configuration was built by Yamazawa et al. [148] in 1993 for robot localization. It was used later for obstacle detection [149] and later in 1998 for map building [140] and video surveillance [98] applications.

2.3.2 *Single camera with single parabolic mirror*

S.K. Nayar was the first one to use a parabolic mirror with a telecentric (orthographic) camera having a hemispherical field of view [93], see Figure 7. Similar sensors were used by Gluckman and Nayar [41] and Sturm[122] for ego-motion estimation and interactive 3D reconstruction respectively.

2.3.3 *Single camera with multiple conic-shaped mirrors*

Nayar and Peri [94] used a combination of two mirrors with conic cross-sections for building what was termed "folded catadioptric camera" based on the fact that SVP is possible by positioning two conic-shaped mirrors such that foci of successive mirrors in a sequence of reflections coincide. As can be seen in Figure 8, they used different combinations of planar (PL), hyperboloidal (HYP), ellipsoidal (ELL) and paraboloidal (PAR) mirror in their design. This type of design leads to a more compact sensor and in return the removal of undesirable optical effects due to the curvatures of large mirrors. Recently, Nagahara et al. [89] proposed an omnidirectional vision sensor which has a single viewpoint and a constant angular resolution. The proposed omnidirectional sensor uses two mirrors, which improves the degree of freedom of the design for satisfying each property.

2.3.4 *Multiple cameras with multiple Planar mirrors*

Using a single perspective camera and a planar mirror is of no practical use since this configuration does not enhance the view field. However, several synchronized perspective cameras looking at several planar mirrors arranged as a pyramid can give a SVP wide field of view with high resolution images. This setup which is often called the "Nalwa pyramid" was first patented by Iwerks [64] in 1964 and later a basically same system was proposed by Nalwa in 1996 [91]. As shown in Figure 9, the main idea is to arrange all camera–mirror pairs such that the effective viewpoints coincide. Some other works that use different variants of Nalwa pyramid are [67, 73, 58, 116, 59]. For example, Hua et al. [59] achieved a wider vertical field of view by arranging six cameras and a hexagonal pyramidal mirror as a component and placing two such sensor components symmetrically back to back such that the effective viewpoints of the two pyramids coincide (see figure 9(c)).

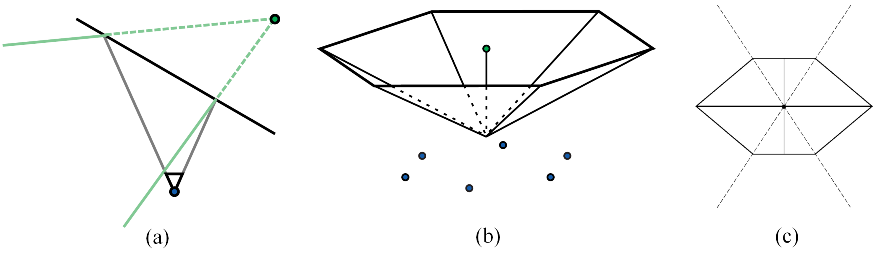


Figure 9: The Nalwa pyramid concept. (a) The image captured a perspective camera looking at a planar mirror is the same image as the one captured by the same camera located behind the mirror, at the position obtained by reflecting the original camera. (b) a pyramidal layout of several such camera–mirror pairs gives a SVP wide field of view with high resolution images. (c) Double vertical FOV design: Two hexagonal pyramidal mirrors component are put together symmetrically such that the effective viewpoints of the two pyramids coincide.

To have a hemispherical field of view, recently Gao et al. [37] proposed a similar design consisting of multiple imaging sensors and a hexagonal prism mirror with six cameras plus a real camera located inside the pyramid with its view point coincident with effective viewpoint of side cameras (see figure 10).

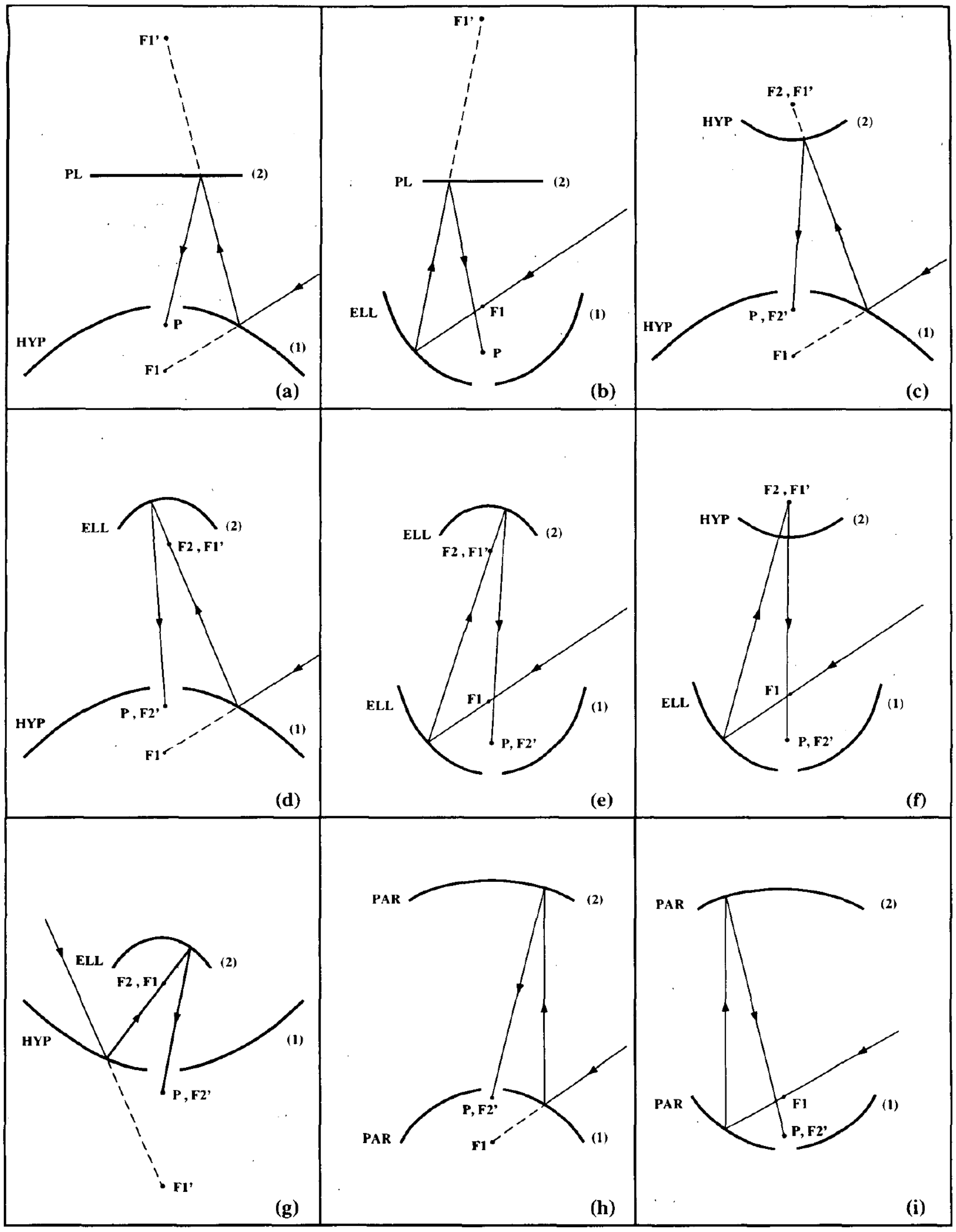


Figure 8: A dictionary of two-mirror folded catadioptric camera designs that satisfy the single viewpoint assumption.

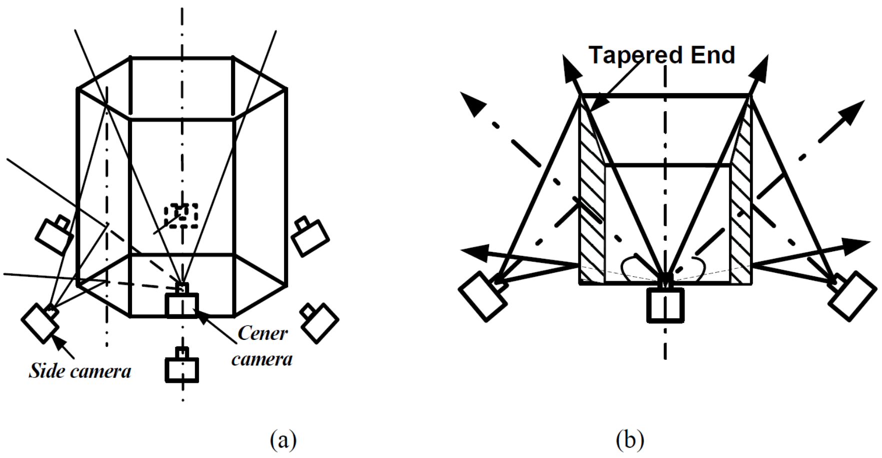


Figure 10: The imaging system with hemispherical field of view proposed by Gao et al.. (a) A hexagonal prism mirror with six cameras plus a real camera located inside the pyramid. (b) A cross-section showing the hemispherical field of view coverage.

2.3.5 *Multiple cameras with multiple parabolic mirrors*

S.K. Nayar also used a back-to-back configuration of two parabolic mirrors with two telecentric cameras to achieve a full spherical field of view [93]. To do so, each paraboloid is cut by the horizontal plane that passes through its focus resulting in a field of view exactly equal to a hemisphere. Symmetrically placing two such paraboloid back-to-back ensure that their viewpoints coincide and therefore creating a SVP omnidirectional sensor with entire spherical field of view. The catadioptric cameras with full sphere field of view in figure 7(d) is a back-to-back configuration of two identical sensor in 7(c).

2.4 CAMERA CALIBRATION

Calibration can be defined for every image pixel as the determination of the 3D ray along which light travels to reach the image plane in some common coordinate system. Regardless of the steps involved, all the calibration technique try to find and remove the effects of refractions, radial distortions, reflections (for the case of catadioptric cameras) and such on the light rays entering the camera. This enables us to map any given image pixel to a ray in space which pass through the camera centre and the corresponding pre-image of the pixel in 3D world. In other words, camera calibration enables us to back-project any image pixel to its corresponding ray in 3D.

The calibration method either use a calibration object such as planar grids, spheres or any other suitable 3D object with some known metric measures or point correspondence across several images from an unknown solid scene (self-calibration or auto-calibration).

In this section, for the sake of completeness, we will briefly mention some recent calibration methods for SVP systems. We skip few available tailor-made methods for multiple image acquisition systems

2.4.1 *Single perspective camera*

The simplest approach for perspective camera calibration based on pinhole camera model is to find a linear mapping in projective space in order to estimate the internal camera parameters, mainly focal length and principal point. However, this approach cannot fully model many real camera systems with nonlinear distortion lenses. Perspective camera calibration is now a mature field and there exist many techniques that take into consideration the radial and tangential nonlinear distortion effects using simple parametric models in early works (Tsai [135], Heikkila and Silven [54], Zhang [154], Sturm and Maybank [124], Heikkila [53]) to high order parametric models capable of handling very large field of views and fish-eye lenses (Shah and Aggarwal [118], Kannala and Brandt [66]).

2.4.2 *SVP catadioptric systems*

Here, we will briefly mention some representative techniques for calibrating SVP systems. For a full coverage on both central and non-central catadioptric calibration techniques found in literature and their common and most used ideas, refer to Shabayek [117], chapters 2-7.

Micusik and Pajdla [82] use point correspondences and catadioptric epipolar geometry constraints to generalize the geometric distortion model and the self calibration method described by Fitzgibbon in [34].

Few methods use geometric invariants for the calibration procedure. Barreto and Araujo [7] have shown that the central catadioptric sensor can be fully calibrated from one image of three or more lines and Ying and Hu [150] use images of lines and spheres as geometric invariants.

Scaramuzza et al. [112] proposed a very fast and completely automatic procedure based on the assumption that the imaging function can be described by a Taylor series expansion whose coefficients are the calibration parameters to be estimated.

Ramalingam and Sturm [105] adapt the generic approach of Sturm and Ramalingam [125] which uses three or more views of a calibration grid, acquired from unknown viewpoints, to calibrate a general imaging system. They see the calibration problem as a motion estimation problem between these calibration grids and they propose a four point calibration algorithm that computes the motions between these grids using triplets of 3D points (lying on the three grids) for four pixels in the image which in return allow recovering the projection rays.

Later, by relaxing the assumption of using a calibration grid, Ramalingam et al. [107] consider the self calibration problem using a central variant of a generic imaging model which assigns projection rays to pixels without a parametric mapping and the calibration is purely performed from image matches. However, the motion of the camera is assumed to be pure translation and/or rotation. These motion considerations together with image matches form geometric constraints on the projection rays.

The fact that central catadioptric cameras can be considered as a camera with general distortion is exploit by some researcher (Thirthala and Pollefeys [131], Ramalingam et al. [106], Tardif et al. [128]) for modeling catadioptric systems. In this sense, the projection model can be seen as a projection from a perspective camera followed by a non-parametric displacement of the imaged point in the distortion centre direction (i.e. the image is non-linearly distorted). They also assume the radial symmetry (i.e. the displacement of a point is a function of its distance from the distortion centre) and the coincidence of the distortion centre with the principal point.

2.5 homogeneous presentation

For the rest of this thesis, we use homogeneous presentation of points and lines in order to benefit from simple mathematical tools for defining lines and their intersections on the image plane [52]. For a point $p = [x, y]^T$ in the image plane, its homogeneous coordinates are $\tilde{p} = [x, y, 1]^T$. The infinite line supporting the line segment passing through points p_1 and p_2 is represented by cross product of these points: $l = \tilde{p}_1 \times \tilde{p}_2$ and the intersection point of two lines l_1 and l_2 is represented by $I = l_1 \times l_2$. The Euclidean coordinates of the intersection point in the image plane are simply the first two elements of I divided by the third element.

2.6 unified projection model

Geyer and Daniilidis [40] introduced an unifying theory for all central catadioptric systems, covering elliptic, parabolic and hyperbolic projections as well as perspective projection. They showed that the image formation of these systems can be modeled as a two step projecting first from the scene to a sphere centered at the mirror focus point and then projective mapping from the sphere to the image plane with a projection center on the symmetry axis of the sphere perpendicular to the plane. The position of the point on the axis depends on the mirror shape. Later on, Barreto and Araujo [6] introduced a modified version of this unifying model based on three steps. This model was further extended by Mei and Rives [81] to include optical distortion and the misalignment between the sphere axis and the images plane and it was used for developing a general calibration toolbox. Besides the obvious advantage of unifying the geometrical model for this range of SVP omnidirectional sensors, the unified model reduces the number of independent unknown parameters to be estimated during the calibration.

Since, in this thesis, we use the calibration algorithm of Mei and Rives (which is also capable of calibrating fish-eye images) to calibrate our sensors, we mention, in the followings section, the necessary equations for lifting a pixel on the image onto the unitary sphere using the calibration results of this toolbox.

2.6.1 *A full omnidirectional projection model*

The necessary equations for lifting a pixel on the image onto the unitary sphere (pixel point to metric ray) are already derived in several slightly different formulations in literature. As was mentioned above, we use the projection model of Mei and Rives [81] which is an extension of the model proposed by Barreto and Araujo [6] and Geyer and Daniilidis [40]. Figure 11 shows the steps involved in the projection of a 3D point in the scene to a pixel point on the omnidirectional image plane.

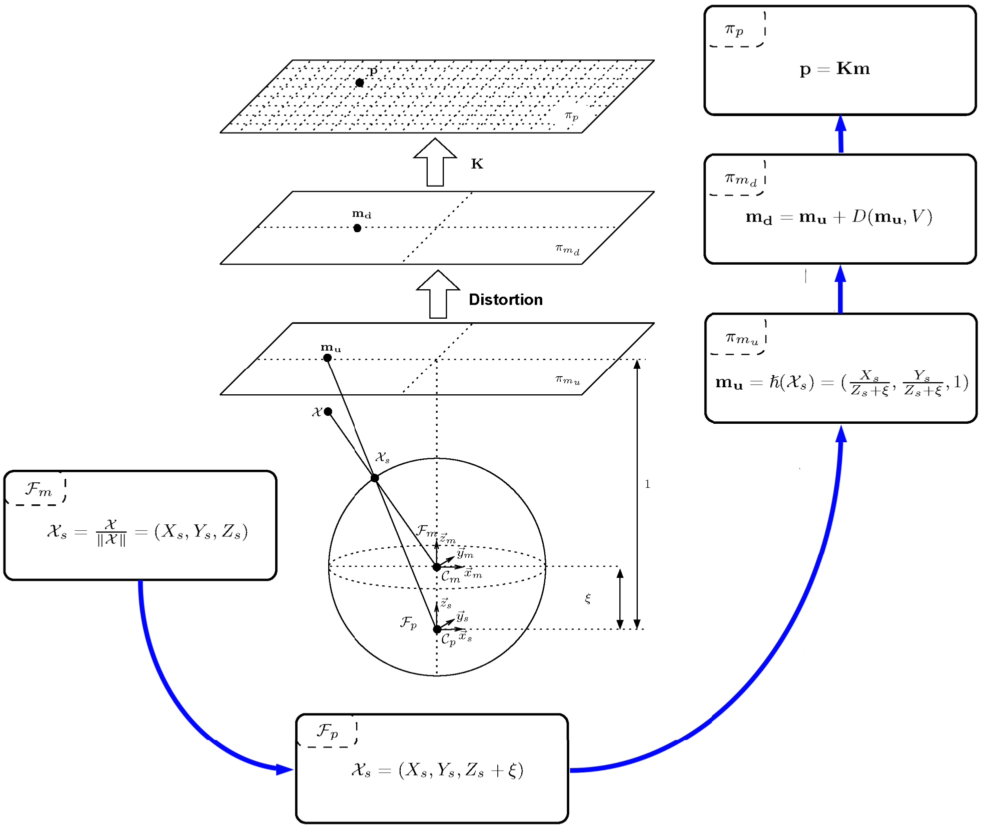


Figure 11: A full model for projection of a 3D point in the scene to a pixel point on the omnidirectional image plane.

By normalization, a 3D point $\chi = (X, Y, Z)$, in the mirror/sphere frame is projected onto the surface of the unit sphere: $\chi_s = (X_s, Y_s, Z_s) = \frac{\chi}{||\chi||}$. Mapping $\hbar$ then projects the point from the surface of the unit sphere onto a normalized plane located at unit distance from the projection center, $(0, 0, \xi)$, defined by

Camera	ξ	η
Parabolic	1	$-2p$
Hyperbolic	$\frac{d}{\sqrt{d^2+4p^2}}$	$\frac{-2d}{\sqrt{d^2+4p^2}}$
Ellipse	$\frac{d}{\sqrt{d^2+4p^2}}$	$\frac{2d}{\sqrt{d^2+4p^2}}$
Plane	0	-1

Table 2: Unified model parameters

unified projection model (Table 2): $\hbar\left(\chi_s\right) = m_u = \left(\frac{X_s}{Z_s+\xi}, \frac{Y_s}{Z_s+\xi}, 1\right)$. The radial and tangential distortions are then added using the distortion function $D\left(m_u, V\right)$ where $V = [k_1\, k_2\, k_3\, k_4\, k_5]$ includes the distortion coefficients.

$$m_d \quad = \quad m_u + D\left(m_u, V\right)$$

The radial and tangential distortion components of the distortion function for the point $m_u = (x, y, 1)$ on the normalized plane are defined through the following equations:

$$\mathcal{L}(\rho) \quad = \quad 1 + k_1\rho^2 + k_2\rho^4 + k_5\rho^6 \tag{2.1}$$

$$d_{tan} \quad = \quad \begin{bmatrix} 2k_3xy + k_4(\rho^2 + 2x^2) \\ k_3(\rho^2 + 2y^2) + 2k_4xy \end{bmatrix} \tag{2.2}$$

where $\rho = \sqrt{x^2 + y^2}$. Finally a generalized camera projection matrix K, projects the distorted point m_d on the normalized plane to the pixel p on the omnidirectional image.:

$$p \quad = \quad K\, m_d = \begin{bmatrix} \gamma_1 & \gamma_1 s & u_0 \\ 0 & \gamma_2 r & v_0 \\ 0 & 0 & 1 \end{bmatrix} m_d$$

where $(u_0, v_0)$is the principal point, s is the skew, r is the aspect ratio, $\gamma_1 = f_1\eta$ and $\gamma_2 = f_2\eta$. The camera focal lengths f_1 and f_2 and the mirror parameter η (Table 2) that depends on the mirror shape cannot be estimated independently. All these parameters along with five distortion coefficients are available as the result of calibration.

2.6.1.1 *The inverse mapping*

The same above steps can be used in reverse to back project a pixel from image plane to a 3D point on unitary sphere. However, because of the high degree distortion model $D\,(m_u,\,V)$, there does not exist any general analytic expression for the inverse mapping and one should take an iterative numerical approach.

2.6.2 *The image sphere representation*

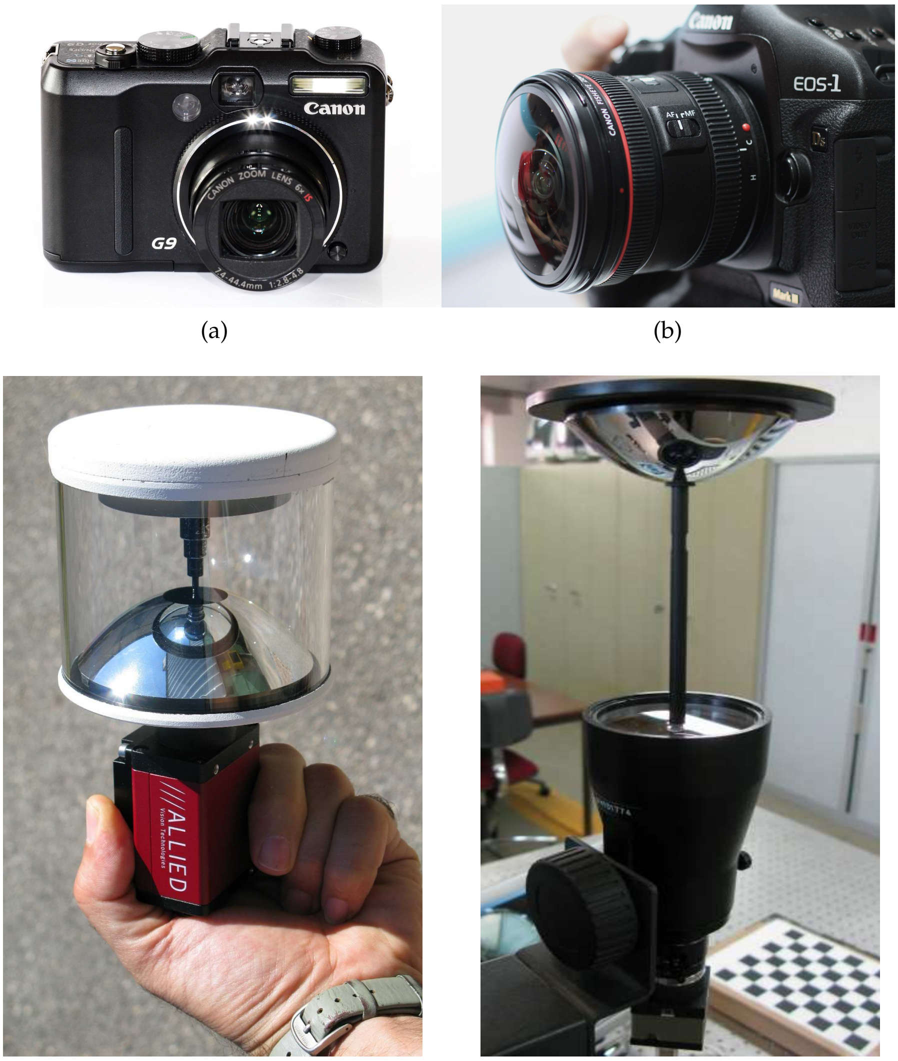
After calibration of the imaging system, we are able to back-project any image pixel to its corresponding ray in 3D. For reconstructing an undistorted image, it is then common to re-project these rays onto some canonical image plane using also some interpolation techniques. Even though choosing a planar image for this purpose seems natural (since CCD arrays and photographic films are planar), choice of a sphere is more advantageous. Unlike the image plane, **image sphere** enables us to present points over the entire view sphere suitable for presenting omnidirectional images. Furthermore, as will be shown in following chapters, the projection of line segment images onto the image sphere gives us some nice properties useful for matching lines.

2.7 OUR CENTRAL IMAGING SYSTEMS

The experiments of this thesis were carried out using images taken by four different categories of sensors as presented in the figure 12(a-d). The perspective sensor was a canon G9 and we employed the calibration technique using planar grid developed by Mei and Rives [81] to accurately calibrate the camera. A Canon DS126081 with a Canon EF 8-15mm fisheye lens was used for taking fisheye images. As for catadioptric sensors, we experimented with two different kinds of paracatadioptric sensors, provided by RemoteReality. The first one was a folded catadioptric camera with configuration *h* (figure 8) and the second one was a classic paracatadioptric system composed of a parabolic mirror in front of a telecentric lens. All these wide field of view sensors were calibrated using the generic toolbox of Scaramuzza et al. [113].

2.8 SUMMARY

In this chapter, beginning with basic pinhole camera and its image formation process, a classification of the existing central imaging systems with some representative examples for their most common configurations were presented. Based on their fabrication technologies, omnidirectional cameras were classified to special lenses (such as fish-eye lenses), multiple image acquisition systems (such as multiple cameras oriented in different directions) and catadioptrics (obtained by combining mirrors and conventional cameras). The catadioptric systems were further investigated with respect to the mirror geometry. Finally, the unitary



(a)

(b)

(c)

(d)

Figure 12: Four different categories of sensors used in this thesis. (a) Perspective camera.(b) Camera with fisheye lens.(c) Camera with folded optics.(d) Paracatadioptric system composed of a parabolic mirror in front of a telecentric lens.

sphere and some common calibration techniques for central imaging systems concluded the chapter.

Part II

LINE MATCHING FOR CONSTRUCTED SCENES USING TWO VIEWS

In this part, line matching for constructed scenes using two views is investigated and some efficient algorithms are proposed.
Chapter 3: We present a state of the art on line matching.
Chapter 4: Line matching across images taken by a central imaging system with focus on short baseline motion of the system is proposed.
Chapter 5: Here, a generic method of matching randomly oriented (however parallel to a scene plane) lines between two views of a constructed scene is addressed.

3 STATE OF THE ART ON LINE MATCHING

The most used features of interest are generally points and there exist a number of novel approaches to wide baseline matching of interest points, some of them such as SIFT (Scale-invariant feature transform) [72] and SURF (Speeded Up Robust Features) [11] are now even considered as reference ones. Less work has considered matching of other features of interest such as line segments. One of the main reasons for this lack of interest in research on lines might be that a set of corresponding infinite lines does not constrain the motion of the camera in two images. For this reason, at least three views are needed to perform 3D reconstruction using lines, whereas two images are enough for points. However, as we will show in following chapter, man-made environments do contain lots of linear structures which in return provide us with some interesting properties such as vanishing point features which we use to develop some simple algorithms for line matching. Furthermore, these properties plus some scene constraints such as planarity of constructed scenes, can be used for motion estimation and 3D reconstruction from line correspondences in two views (Part iii).

Here we will present some various approaches of line matching. A more specific mention of related line matching techniques for man-made scenes will be presented in their corresponding chapters.

3.1 PERSPECTIVE LINE MATCHING

The earliest works involving line segments date back to 1970's where researchers were interested in grouping line segments belonging to the same solid object in the scene and to extract its position and 3D structure based on the topological configuration of its segments in the image. Roberts (1963) [108] assumed that a scene could be decomposed into a number of primitive polyhedra. These primitives then were recognized as a specific polyhedron by looking iteratively for expected transformed versions of the primitives in the image. After finding the most likely primitive polyhedron, the process was restarted until all objects are recovered. Similarly, Guzman-Arenas and Guzmaan [50] and later both Huffman (1971) [61] and Clowes (1971) [24] proposed a similar system where the system looked for instances of polyhedra primitives based on the type of junctions in the image such as L junctions,T junctions, arrow-shaped junctions or based on whether the segment was the image of a concave or convex edge, or whether it was an occluding contour. Though originally working with only one image, this 3D pattern recognition technique can eventually be used for matching primitive polyhedron objects (in other words, a set of line segments) between two or more views by simply interpreting and grouping the same line segments belonging to the same object in different views. The method, however, works only for scenes with only solid polyhedron objects.

In line segment-based algorithm introduced by Medioni and Nevatia [78, 79], each line segment descriptor consists of its orientation, the average gray-level intensity along the orientation and the coordinates of its endpoints. An iterative matching procedure is employed where in each iteration, a hypothesized match between two segments is accepted if the match can help matching many of the other segments. Their algorithm allows for the possibility of fragmented segments by considering sets of matches together. They compute iteratively so called the "minimum differential disparity" evaluation function applied over neighboring edge segments to determine the power of each match. The lower is this criterion for a putative match, the stronger is the match. The minimum differential disparity value for each possible pair is computed based on the overlapped length of the matching segments along the epipolar lines. The method can work only with very small view changes between the images therefore it is more suitable for short baseline motion.

Ayache and Faverjon [2] describe each edge segment using the coordinates of its midpoint, its length, and its orientation. They first use local constraints to find a set of initial matches. A pair of line segments is considered a potential match if the midpoints of the two segments satisfy the epipolar constraint near an expected disparity value and their length ratio and orientation difference lie below a preset threshold. Then a global correspondence search is applied on these potential matches consisting of a prediction and recursive propagation process. For their method to work, the cameras need to be fully calibrated.

In their work, McIntosh and Mutch (1988) [75] match lines based on geometric and photometric properties of line segments such as length, orientation and contrast. Therefore their approach is sensitive to illumination changes and considerable camera motions. Similarly, Gros et al. (1998) [47] used angles and length ratios between line segments to match them based on the naive assumptions that the transformation between two images is either a similarity or an affine transformation.

Gu et al. (1987) [49] presented each polygon by using a unique string which encodes the concavity or convexity of each vertex of the polygon. Polygons are then matched based on the similarity of their string values.

In [57], Horaud and Skordas proposed a matching algorithm based on graphs . For each image, a graph, so called a relational graph, is built which encapsulates the lines of the image as its nodes and edges in the graph include the relational information between line segments such as whether they are collinear or which side (left or right) one segment is located with respect to another one. Based on these two relational graphs, a third graph, so called a correspondence graph, is then constructed which encapsulates, as its nodes, a set of potential assignments for each segment in the other image and edges in this graph are established on the basis of segment relationships. Finally matching is carried on by searching for sets of mutually compatible nodes in this graph by looking for the maximal clique which maximizes a benefit function.

Tsai [133, 134, 132] used the geometric hashing idea introduced by Wolfson and Lamdan [143]. Considering each 3 line segments in the first image, an affine invariant value can be computed from any forth line. A hash table indexed by the invariant value is then used to store all possible triplets of lines in the first image. Putative matches are then recognized by forming triplets of line segments from the second image followed by computing their invariants and searching the hash table for the triplets from the first image which have the same invariant value. Finally, a voting scheme is used to find possible correspondences.

In [152], Zhang tries to solve the matching problem by taking long sequences of images over short time interval. Since the time interval is small, the correspondence of a token at the following image must be in the same neighborhood as in the previous image. Consequently the (extended) Kalman filter is used for matching. This technique, however, does not apply in cases where the assumption of short time interval does not hold such as images taken with large baseline.

Schmid and Zisserman (1997) [115] used the epipolar geometry between the two images, which is assumed to be known, and present a matching algorithm based on correlation of the neighborhood around the line segments of potential line matches. They also use the trifocal tensor as a tool for verification or rejection of matches between three views. However, the assumption of known epipolar geometry limits the usefulness of these methods.

Tell and Carlsson [130] look at line segments between two Harris corners and use Fourier coefficients of the intensity values along the segments as descriptors. The large number of corner pair combinations that may have to be considered is a main drawback of this method.

Goedeme et al. [45] use invariant column segments as the local image features and therefore their method deals only with vertical lines. For each vertical segment, a descriptor vector is computed based on geometrical, color and intensity information. The motion is restricted and the camera may translate only in the horizontal plane and rotate only around a vertical axis (such as a mobile robot). Under this camera motion constraint, a vertical line in the world always projects to a vertical line in the image plane which will only be scaled around the point where it intersects the horizon. This provides a geometrical invariant which is the ratio between the vertical segment length and the distance of the midpoint of the segment to the horizon.

Bay et al. [12] proposed to match line segments, first based on the histograms of the neighboring color profiles and then to use the topological relations between all line segments to remove false matches as well as to find more matches. Their method has two main drawbacks. Firstly, it is computationally expensive since the matching propagation is an iterative process. Secondly, due to using the color histogram for finding the initial matches, it is less robust to illumination and other image changes.

	References
Short base-line	Medioni [78], Medioni and Nevatia [79], Ayache and Faverjon [2], McIntosh and Mutch [75], Gros et al. [47], Gu et al. [49], Horaud and Skordas [57], Tsai [134], Zhang [152]
Short base-line/large rotation	The work presented in chapter 4
Long base-line	Schmid and Zisserman [115], Tell and Carlsson [130], Goedeme et al. [45], Bay et al. [12], Wang et al. [139, 138]

Table 3: Classification of perspective line matching methods

MSLD proposed by Z.H. Wang et al. [139] is a descriptor for line matching analogous to SIFT for point matching. It is based on defining a pixel support region for each pixel of a line and then accumulating a histogram of image gradient for this region. The mean and standard deviation of these histograms form the final MSLD descriptor. This descriptor relies on photometric information around the segments which is not usually enough rich and the method may also fails when encountering repeated textures, not to mention that it can not handle the problem of scale changes.

L. Wang et al. [138] proposed to use angles and length ratios between lines computed by their endpoints to describe a pair of line segments (named Line Signature). The line matching is then done on the basis of pairs of line segments. Since Line Signatures rely on the endpoints of line segments, the method may fail when the endpoints are not accurate enough.

3.1.1 *Classification*

The above methods can be classified based on the amount of projective distortion between two views (see table 3). The projective distortion is directly connected to translation of the system with respect to the distance of the camera to the scene and the rotation of the camera. We classified the methods into three groups: short base-line (where neither the translation nor the rotation is considerable), long base-line (where the translation and/or rotation are large) and a middle class where the translation is small but the rotation can be arbitrary and large. The following table classifies the above methods into these three groups. Obviously, the methods which work for long base-line motion also work for the other two kinds of motion but not vice versa. Similarly, the methods which work for short base-line/large rotation motion also work for the short base-line motions but not vice versa.

Line matching methods can also be categorized into two major groups. One group includes those methods which match each line separately without taking into account the rest of the lines at the same time [1, 80, 152, 2, 45, 139]. The other group consists of the methods which try to solve the correspondence problem by considering spatial relationships among all the segments [100, 47, 51, 115].

3.2 omnidirectional line matching

Because of the non-linear distortions introduced by the large field of view of omnidirectional cameras, the above perspective line matching methods can not be directly applied to images taken by these imaging devices. Provided that the imaging model is known and that the omnidirectional camera is central, one solution to apply these methods is to generate a perspective view out of the omnidirectional image[74]. It is obvious that this approach is computationally very expensive and the performance of the algorithms greatly decreases since the unwrapped perspective images have non-uniform image quality.

The algorithms which directly work on omnidirectional images are faster, though they still have to deal with unavoidable problems from non-uniform resolution of the image. Scaramuzza et al. [111, 114] developed a stable descriptor based on the image gradients which is unique, distinctive and invariant to rotation but it can only describe vertical lines. Assuming that the optical axis of the camera is also vertical, all world vertical lines project into radial lines on the image plane. Brassart et al. [19] also propose a catadioptric line matching method which can deal only with 3D lines parallel to the optical axis of the camera.

More recently, Vasseur and Demonceaux [137] have proposed a method for catadioptric line matching across multiple images. They present catadioptric lines by their normals in sphere space and use only these normals and their relative positions in order to perform the matching. If the motion of the camera is a pure rotation, then the corresponding lines are related by the rotation and they propose a voting method to find an initial set of two matched lines which are enough to estimate the rotation.

As for a generic motion, they then show that for a translation up to the scene depth, the angle difference between two normals of any couple of 3D lines in two image spheres is less than 20 degrees on average and therefore suggest to use the same voting approach than in the pure rotation case in order to find the initial set of two matched normals. They proceed by constructing hashing tables based on bases defined by few couples of normals associated to the longest lines in the first image followed by a voting scheme in the second image to select the best corresponding bases and subsequently to match the rest of the lines.

3.3 summary

This chapter reviewed main existing methods of line matching for central imaging systems including their classification based on the kind of motion which can be handled by each method. In the following two chapters, we investigate constraints and algorithms for matching lines under short baseline motion and wide baseline motion.

4 SHORT BASE-LINE LINE MATCHING AND IMAGE STITCHING FOR CENTRAL IMAGING SYSTEMS

Following our objective for developing a generic line matching method for constructed scenes especially applicable to omnidirectional images, we start with tackling the simplified problem where the motion of the system is mainly an arbitrary rotation and the translation of the camera between two views with respect to its distance to the imaged scene is negligible. We start by studying the relationship between images of lines on unitary sphere followed by proposing a simple algorithm for matching lines assuming the rotation of the system is known a priori or it can be estimated from some correspondences in two views. Two methods are also discussed for retrieving R in the case it is not known a priori.

4.1 INTRODUCTION

Our method for line matching consists of 3 main steps. First, lines of interest have to be detected in both images. Second, the line segments are projected from 2D to 3D by lifting to unitary sphere. Locating corresponding lines using the relation derived in the following section is the final stage of the line matching algorithm. The geometric relation between two images required a priori here is the rotation of imaging system. This can directly be recorded during image acquisition or later by different available methods such as methods based on matching corresponding vanishing points [14, 15]. In this work, we are interested in the last step of the matching algorithm. In [85, 83], we presented a pipeline for automatic line matching with focus on paracatadioptric systems under the short baseline motion of the system by employing line intersection correspondences as input to RANSAC in order to compute the rotation of the imaging system. In this work, however, we employ two different methods for estimating rotation; one is an already developed and robust method of retrieving the rotation using vanishing points direction and second one is a simple alternative method which will shortly be explained. We aim to formulate a generic method of matching lines for all central imaging systems under the short baseline motion including perspective cameras. While in the perspective case line matching is rather efficiently solved, to the best of our knowledge, this is the first work dealing with this problem in catadioptric images.

4.2 PROPOSED METHOD

In this section, we derive the relation between normal vector of the great circle of any 3D line represented in the first unitary sphere coordinate system and its corresponding vector expressed in the second system which in return gives us an adequate tool to match lines. Then, we give a brief description of the two algorithms for recovering the rotation of the imaging system.

4.2.1 *The relation between images of 3D lines on unitary sphere*

Even if the motion of the camera is mainly a rotation, the images of lines on the image plane are arbitrarily located and there is no definitive geometric constrain for matching them. This problem is even more serious for omnidirectional images due to huge deformation of the image which makes any photometric-based line matching tools useless.

Having the intrinsic parameters of the imaging system, the key idea is to project the image on the unitary sphere, turning the conic curves (images of the lines on the image plane) into their corresponding great circles on the unitary sphere. Knowing that a great circle is fully defined by the normal vector of its plane, the problem of matching conics is then reduced to matching these vectors. In this section we show that under short range motion, two corresponding great circles are mainly related by the rotation part of the imaging system motion. Consider a line in 3D scene with two separate 3D points X_1 and X_2 on it. Suppose n is the non-normalized vector of the plane which passes through these two points and the origin of the first unitary sphere and n' is the corresponding vector expressed in the second model (Figure 13). Then:

$$\begin{aligned}
&n' = (RX_1 + t) \times (RX_2 + t) = \\
&det\,(R)\,R^{-T}\,(X_1 \times X_2) + [t]_\times\,R\,(X_1 - X_2) = \\
&det\,(R)\,R^{-T}n + [t]_\times\,R\,(X_1 - X_2) = Rn + [t]_\times\,R\,(X_1 - X_2)
\end{aligned}$$

Where the metric transformation of the imaging system (represented by two unitary spheres in Figure 13) is defined by the rotation matrix R and translation vector t and R^{-T} is the inverse transpose of R. Note that for a rotation matrix, $det(R) = 1$ and the transpose inverse is the same as R. The above relation coincides with the relation obtained in [92, 8] in which the equivalent Euclidean Plücker representation of the line is used to derive a similar formula:

$$n' = Rn + [t]_\times\,Rl,\; l = \frac{(X_1 - X_2)}{\|X_1 - X_2\|}$$

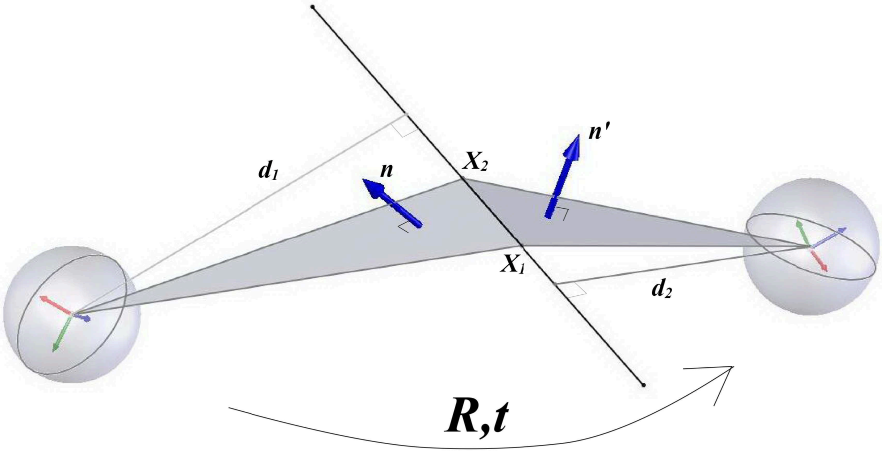


Figure 13: A 3D Line in the scene and its projections on a unitary sphere at two different positions. n and n' are the normal vectors of related great circles.

Where the 3D line segment is represented by its infinite supporting line represented by two vectors l and n. l is a unit vector parallel to the line, and n is a non-normalized vector to the plane defined by the line and the origin of the coordinate system and its norm is equal to the distance of the line to the origin, e.g.$\|n\| = d$, see Figure 13. Therefore if the transformation between two positions of the imaging system is a pure rotation ($t = 0$) or the movement of the system in comparison to its distance to the scene is very small (short baseline, for example aerial imaging), we can neglect the second term in the above equations and conclude that under the pure rotation or short base line motion, n and n' are related by the rotation matrix:

$$n' = Rn \tag{4.1}$$

This equation can also be visually verified as shown in Figure 14. If the motion of the imaging system is a short range motion, the images of a world point on the unitary sphere at two different positions are approximately related by the rotation of the system. If the motion is a pure rotation, it is well-known that there is no parallax and the corresponding points on the unitary spheres are absolutely related by the rotation of the system. Note that there is a considerable arbitrary rotation between two unitary sphere coordinate systems. One immediate result is that for the case of short baseline, after estimating the rotation matrix, for each line in the first image, all which is needed to find its corresponding line in the second image is to multiply R at normal vector of great circle of the line. The calculated vector is pointing at the same direction as the normal vector of great circle of corresponding line is pointing (inside a reasonable angular distance error). In other words, the match $n \rightleftharpoons n'$ is considered a correct one, if the angular difference between two vectors Rn and n' (called $\triangle$ hereafter) is less than a preset tolerance, tol (in radian):

$$\Delta = \arccos\left(|n' \centerdot \frac{Rn}{||Rn||}|\right) < tol \tag{4.2}$$

where ($\centerdot$) stands for dot product and $|....|$ and $||....||$ stand for absolute and norm functions respectively.

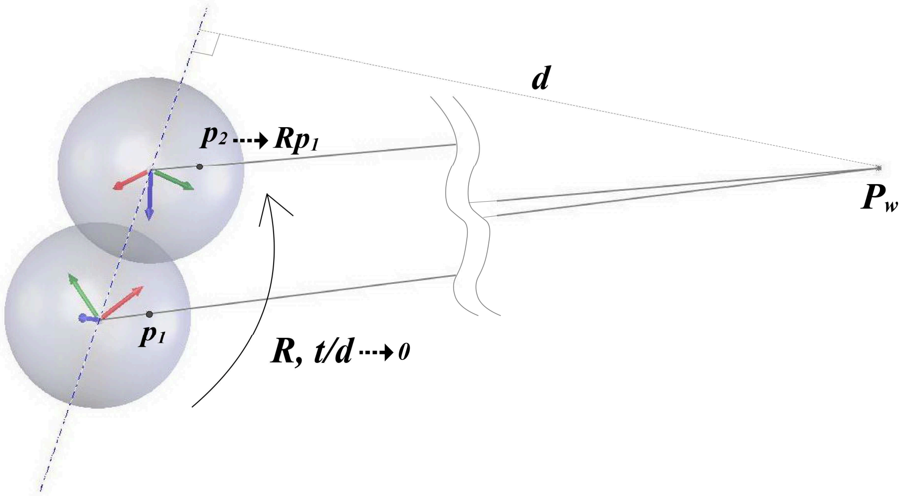


Figure 14: The relation between the corresponding points on the unitary spheres under short baseline motion. The translation w.r.t. scene depth is negligible but there is a considerable arbitrary rotation between two unitary sphere coordinate systems.

4.2.2 *Recovering R*

For our method to work, we first need to recover the rotation between two views. There are several methods for estimating R applicable to all types of central imaging systems (cf. Bazin et al. [14] for a review on these methods and their pros and cons). Regarding the simplicity and robustness, we have experimented with two automatic methods, one from Bazin et al. [14] which works in urban scene with at least two groups of 3D parallel lines and the other one is our proposed method which is suitable for short baseline motion as follows.

4.2.2.1 *Recovering R using vanishing points correspondences*

Having extracted and matched some vanishing point correspondences (Appendix A.3), the relative rotation between two views, can be computed using the simple linear method of Bazin et al. [16]. Assume V_1 and V_2 (and their corresponding V_1' and V_2') are unit vectors corresponding to two vanishing directions. R can be decomposed into a rotation axis N and an angle θ which can be recovered as follows:

- if $V_i' = V_i\,,\, i \in [1,2] \Rightarrow R = I$.

- if $\begin{cases} V'_i = V_i & i,j \in [1,2] \\ V'_j \neq V_j & i \neq j \end{cases} \Rightarrow \begin{cases} N = V_i \\ cos\theta = \frac{V'_j.V_j-(V_j.N)^2}{1-(V_j.N)^2} \\ sin\theta = \frac{V'_j.(N\times V_j)}{||N\times V_j||^2} \end{cases}$

- if $V'_i \neq V_i\,,\, i \in [1,2] \Rightarrow \begin{cases} N = \frac{(V_i-V'_i)\times(V_j-V'_j)}{||(V_i-V'_i)\times(V_j-V'_j)||} \\ cos\theta = \frac{V'_j.V_j-(V_j.N)^2}{1-(V_j.N)^2} \\ sin\theta = \frac{V'_j.(N\times V_j)}{||N\times V_j||^2} \end{cases}$

where I is the 3×3 identity matrix. Finally, using Rodrigues' formula, rotation matrix R can be computed as:

$$R = I + sin\theta[N]_\times + (1 - cos\theta)[N]^2_\times$$

where $[N]_\times$ is the skew-symmetric matrix corresponding to vector N.

4.2.2.2 *Recovering R using point correspondences and RANSAC*

The idea behind this approach is already depicted in Figure 14. In the case of short baseline motion, the image of any point from the scene on the unitary sphere goes under the same rotation as the imaging system similar to vanishing points. Exploiting this fact, we suggest the following simple method for recovering R:

Algorithm 4.1 Recovering R using RANSAC.

```
1: Given two images taken by a central imaging system under short range
   motion;
2: By means of automatic feature matching algorithms such as SIFT, extract
   enough point correspondences between two views;
3: Lift these correspondences to the unitary sphere;
4: Using RANSAC [33] or similar fitting algorithms, find the best rotation matrix
   which relates these corresponding 3D points;
```

Note that this method is only feasible when the imaging system goes under a short range motion. Note also that theoretically, having the images of two salient 3D points (which are not collinear with the center of unitary sphere) and their correspondences is sufficient to estimate the rotation matrix. However we employ RANSAC to be able to automatically extract some interest points and their correspondences without being concerned about the errors in detection of positions of these points in the image planes and also any possible mismatches. Note also that the simple linear method of previous section also can be applied on any two accurate correspondences to estimate the rotation.

4.3 IMPLEMENTATION DETAILS

Since up to now, there is not any generic feature matching method applicable to omnidirectional images (and therefore capable of handling the deformation of such images), and especially since our aim is to develop some generic tools for constructed scenes (where vanishing directions are usual to be found), we adapt the method based on vanishing points to recover the rotation. Therefore in our experiments, we used the first approach since we had enough vanishing points available. The first approach is also more efficient since it uses already extracted features of our interest, lines, to estimate R in comparing to the second approach which includes extra steps of extracting salient point correspondences and fitting a rotation matrix to them.

The proposed method is composed of the following main steps:

Algorithm 4.2 Short baseline line matching algorithm

```
Given two images taken by a central imaging system under short range motion and the preset tolerance $tol$;
Extract their lines by computing normal vectors of their great circles after projection onto the unitary sphere (cf. Appendix A.1);
Extract and match two dominant vanishing directions among the extracted lines and use them to estimate $R$ (cf. section 4.2.2.1 and Appendix A.3);
Match lines using the relation 4.2;
```

Figure 15 along with Figure 16 demonstrates the steps of our algorithm on a pair of synthetic images. We have applied the R on whole first sphere for the sake of demonstration. In practice and during the implementation only the normal vector and two end points of each segment (necessary to find the segment bounding box for the case there are ambiguities) are affected.

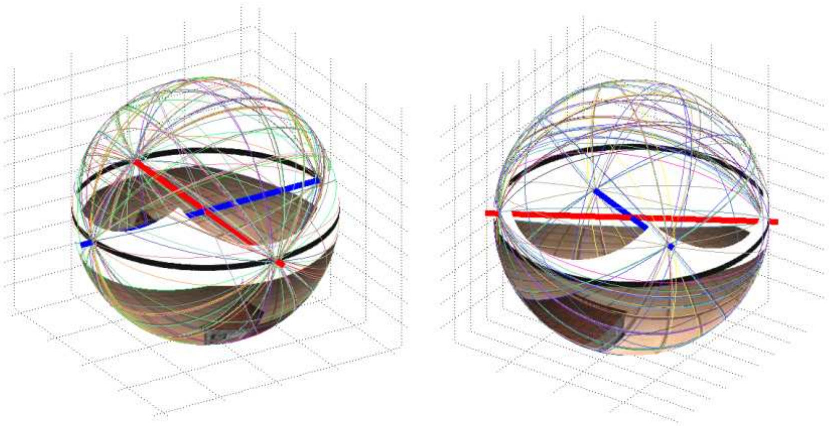


Figure 15: Projection of two paracatadioptric images on the unitary sphere, their extracted great circles and two dominant vanishing directions. For a better demonstration, half of the great circles are hidden.

In the last step, for some segments, matching great circles using the relation 4.2 is not enough and one ambiguity may occur due to fragmented segments when more than one segment are lying on corresponding infinite line because these segments are all located on the same 3D scene plane (as it is shown in Figure 18). To resolve this ambiguity we also find the corresponding bounding box of the segment in the first image and we choose the candidate segment which is inside the bounding box or is intersecting it.

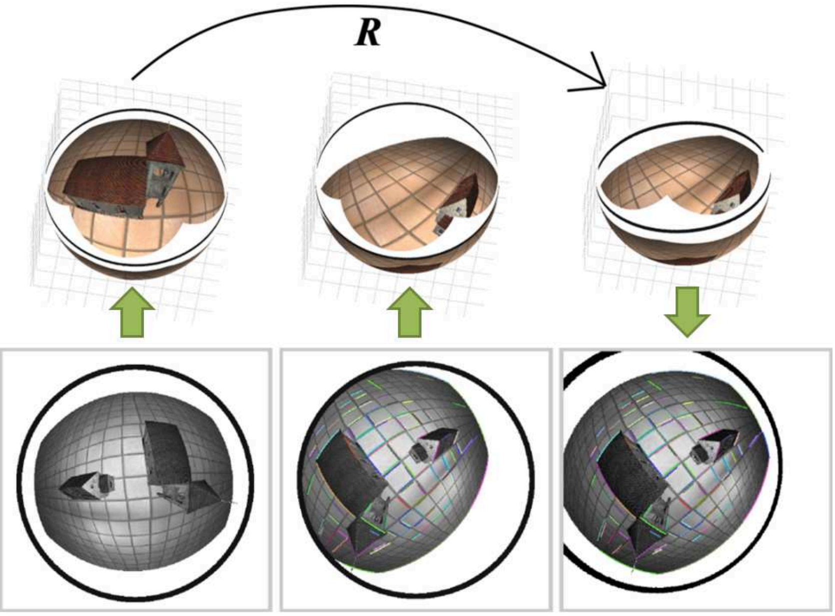


Figure 16: Steps of the proposed algorithm: Lifting images onto unitary sphere, recovering R and rotating the first image according to R. Segments on the back projected image now coincide with their corresponding in the second image.

Before extending the above algorithm to a more efficient one, it is necessary to explain a simple calibration method for perspective cameras as follows:

4.3.1 *Auto-calibration of the perspective camera*

The perspective camera which we use in our experiments is a Canon G9. During several calibrations, we found that the camera pixels are square, so called a natural camera. In fact, for a modern CCD camera, it can be assumed that the pixel is a square and the principal point is also close to the middle of image plane [103]. It has been shown that for such a camera, the image plane is a metric plane and two orthogonal vanishing points are sufficient to estimate the focal length of the camera which is the only unknown from the camera matrix to be estimated [69]. We benefited from this property to remove the hassle of calibrating the camera before each image acquisition while allowing for the focal length to change (zooming) during the acquisition and instead to calibrate the camera using simply vanishing points which are needed to be extracted for estimating the rotation anyway.

Assume two image points $v_1 = (v_{1x}, v_{1y})$ and $v_2 = (v_{2x}, v_{2y})$ are such orthogonal vanishing points, and K, the camera matrix for a natural camera with principal point in the center of the image is:

$$K = \begin{bmatrix} f & & h/2 \\ & f & w/2 \\ & & 1 \end{bmatrix}$$

where (h, w) is the size of the image. Two image points v_1 and v_2 back-project to two rays with directions $V_1 = K^{-1}\tilde{v}_1$ and $V_2 = K^{-1}\tilde{v}_2$ in the camera coordinate system. The angle between the two rays is then given by the familiar cosine formula:

$$cos\theta = \frac{V_1^T V_2}{\sqrt{V_1^T V_1}\sqrt{V_2^T V_2}} = \frac{\tilde{v}_1 \omega \tilde{v}_2}{\sqrt{\tilde{v}_1 \omega \tilde{v}_1}\sqrt{\tilde{v}_2 \omega \tilde{v}_2}} \tag{4.3}$$

where $\omega = (KK^T)^{-1}$ is the image of absolute conic (IAC). Since two vanishing points are orthogonal, the above equation is simplified to:

$$\tilde{v}_1 (KK^T)^{-1} \tilde{v}_2 = 0 \tag{4.4}$$

which is a quadratic equation in term of f with the solution:

$$f = \frac{\sqrt{-4v_{2x}v_{1x} + 2v_{2x}h - 4v_{2y}v_{1y} + 2v_{2y}w + 2v_{1x}h + 2v_{1y}w - w^2 - h^2}}{2} \tag{4.5}$$

If more than two mutually orthogonal vanishing points are available (*e.g.* three Manhattan directions), any two vanishing points among three possible configurations can be randomly selected to estimate the focal length. The three calculated focal length will not necessarily be identical due to noise and lack of accuracy in detecting vanishing points. An average of these values can be considered as the best estimation for the focal length. If different, the camera matrix for the second camera, K', can be found similarly.

As a final remark, note that we do not need to match vanishing points between views and calibration is done merely using vanishing point from each image separately.

4.3.2 *An extension*

Vanishing point matching algorithm suggested by Bazin et al. [14] (refer to Appendix A.3 for details) assumes that vanishing points provide strong photometric information inside the spherical regions defined by them in the equivalent sphere. However, during our experiments with real images taken from constructed scenes, we found that these histograms are not enough discriminative for images of poorly textured scenes, like indoor environments and constructed scenes where edges and plain homogeneous surfaces such as walls are dominant. As a result we developed a more compact algorithm, relaxing the assumption that vanishing points are already matched. The new algorithm tries to match vanishing points and lines simultaneously.

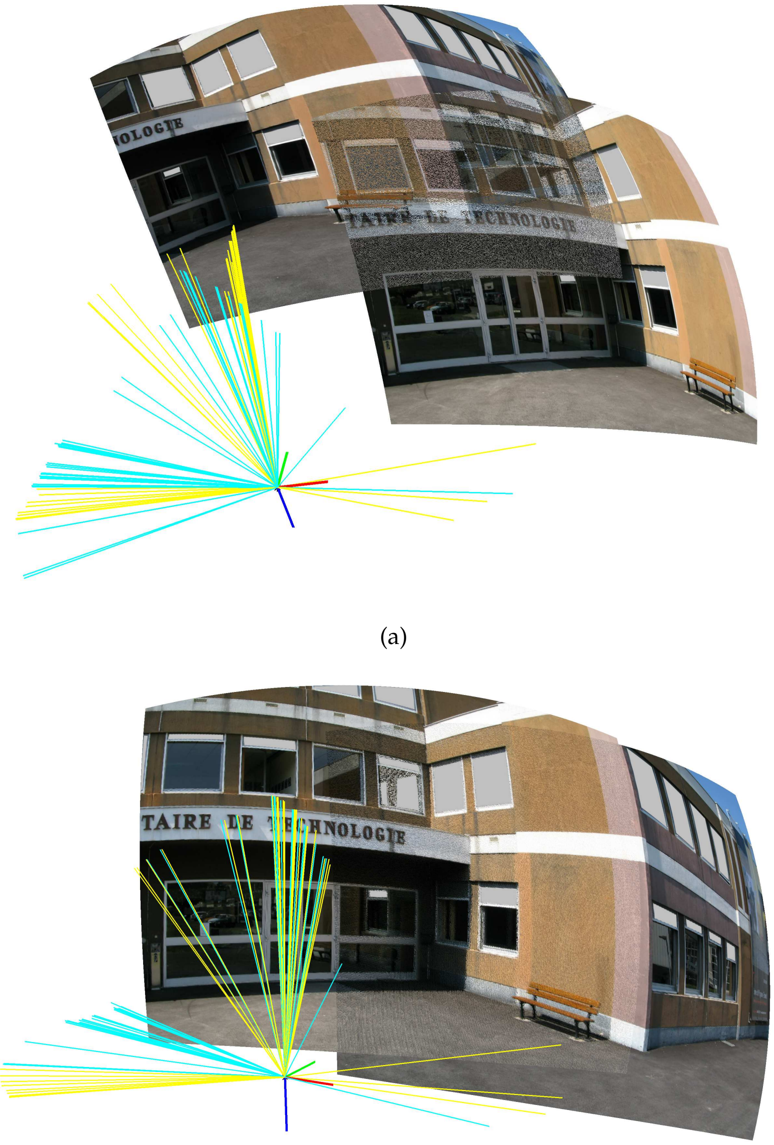
After extracting vanishing points, the algorithm considers all possible matching solutions between vanishing points and picks up the one that results in maximum number of line correspondences returned using the relation 4.2. If lines are enough randomly distributed in the images, the highest number of matches will occur under correct correspondences between vanishing points. Figure 17 better explains the idea. Two images are projected on the unitary sphere after computing and applying the rotation on the second image. The normal vectors corresponding to lines in each image are drawn at the origin of the sphere with a different color (cyan for the first image and yellow for the second image). If the computed R is correct (which means if the assumed correspondences between vanishing points is correct) then one expects more correspondences to be found compared to the number of correspondences for a wrong estimation of R. To better visualize the figure 17, notice that since all normals are drawn with identical norm, two lines are matched if their normals appear with the same length in the figure and they are closely located beside each other.

The number of possible matching solutions is 8 (if only two vanishing points are detected) or 16 (if three vanishing points are extracted). To justify these numbers, let (v_1, v_2, v_3) be three unit vectors on the unitary sphere corresponding to the 3 main directions extracted from the first image and (v'_1, v'_2, v'_3) be three unit vectors from the second image. Note that given a possible matching solutions $v_i \Longleftrightarrow v'_j$, there exist 4 possible combinations of the two other vanishing points and therefore for 3 vanishing points, there are 14 possible combinations. Since each vanishing direction corresponds to 2 antipodal points on the unitary sphere, the possible combinations will be 24 among which only 16 combinations are identical and 8 are repeated. Algorithm 4.3 summarize our extended method for simultaneous matching of vanishing points and lines as a pseudo-code.

The combinatorial functions $index(max(S))$ return the index corresponding to the biggest value in the vector S.

4.4 EXPERIMENTAL RESULTS

During our experiments, for non-perspective imaging systems (Paracatadioptric and fish-eye), the algorithm suggested by Bazin et al. [13] was used to extract vanishing points and the calibration method of Mei and Rives [81] was employed to calibrate them. For our perspective imaging system (Canon G9), the algorithm proposed by Tardif [127] was employed to extract two vanishing points from the images and the simple calibration method presented in section 4.3 was employed



(a)

(b)

Figure 17: The effect of estimated R on the normal vectors corresponding to lines in images on sphere. (a) R is wrong and two sets of normals do not exhibit any overlap. (b) R is correct and two sets of normals overlap

Algorithm 4.3 The extended algorithm

```
Given two images taken by a central imaging system under short range motion and the preset tolerance $tol$;
if the images are perspective then
    Extract their line segments and dominant vanishing points;
    Calibrate the camera (section 4.3.1);
    Project lines and vanishing points on the unitary sphere;
else
    Calibrate the system (Mei and Rives [81]);
    Extract their lines by computing normal vectors of their great circles after projection onto the unitary sphere;
    Extract all dominant vanishing directions among the extracted lines;
end if
$Sol \leftarrow$ All 8 (or 16) possible matching solutions between vanishing points;
for each $Sol(i)$ do
    Estimate $R$ (section 4.2.2.1);
    $Matches = []$;
    for each line $n$ from the first image do
        for each line $n'$ from the second image do
            if $\Delta < tol$ (Equation 4.2) then
                $Matches \Leftarrow Matches + [\, n \rightleftharpoons n' \,]$;
            end if
        end for
    end for
    $M(i) \leftarrow Matches$;
    $S(i) \leftarrow$ size($Matches$);
end for
$j \leftarrow$ index(max($S$));
return $M(j)$ and $Sol(j)$;
```

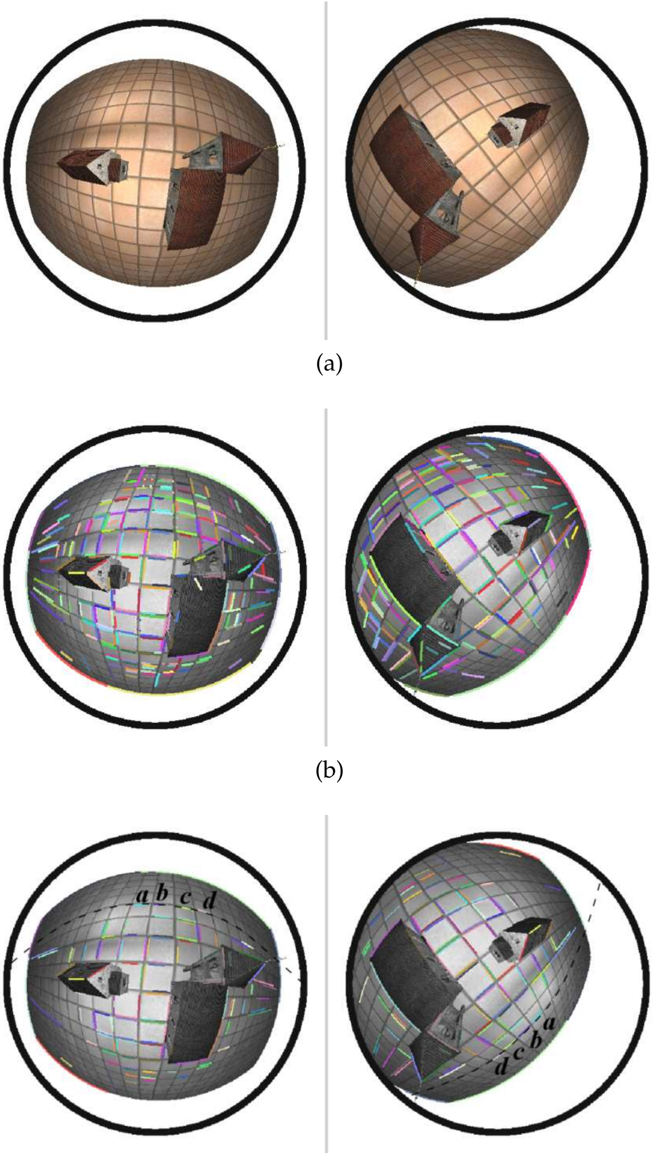
for calibration. Unless mentioned differently, the tolerance *tol* was set to one degrees for all the experiments.

4.4.1 *Synthetic images*

The developed algorithm was first applied on the synthetic paracatadioptric aerial images of Figure 18. There is a 110 degrees rotation around the optical axis of the system and 30 and 20 degrees around two other axes all measured w.r.t. a fixed coordinate system. The two extracted dominant vanishing directions on the unitary sphere are presented in Figure 15. The translation of the system is negligible. Only lines of length 15 pixels or more are considered. The angular threshold for matching great circles is set to one degrees. The total of 261 and 226 segments are obtained for the left and right images, respectively. The algorithm outputs the matches displayed at the bottom row of the figure. All of the 121 matches obtained are correct.

4.4.2 *Real perspective images*

Figure 19 shows the result of applying our algorithm on two real perspective images. Note that this method is far simpler than an approach such in [115] in which photometric properties of the segments neighborhoods along with epipolar geometry are combined to do the same job. For this experiment, rotating imaging system around its focal point was not easy since this point is somewhere inside the camera and we needed a special flexible fixture to carry out the job. However, rotating (while trying to avoid translating) of imaging system can be considered as a short baseline motion. Images of Figure 19(a) are taken by a random rotation of the imaging system in this way. Even though matching 2 vanishing points is enough to recover R, we use 3 vanishing points to reduce the overall error (Figure 19(b)). The recovered R is composed of an approximately 31 degrees rotation around the optical axis of the system and 4 degrees and 4.5 degrees around two other axes, all measured w.r.t. a fixed coordinate system. For this example, the numbers of segments extracted are 284 and 245 for the left and right images respectively. Obtained matches are shown in Figure 19(c). 120 out of 129 matches are correct. The performance of the proposed method has decreased not only because the motion of the system is not a real short baseline motion but mainly because perspective imaging systems suffers from a limited field of view. The wider field of view of a perspective camera results in the better extraction of lines and the longer line segments. Note that the larger error in computing the position of the lines causes larger error in the estimation of the vanishing points and therefore a less accurate recovery of R and eventually more mismatches.

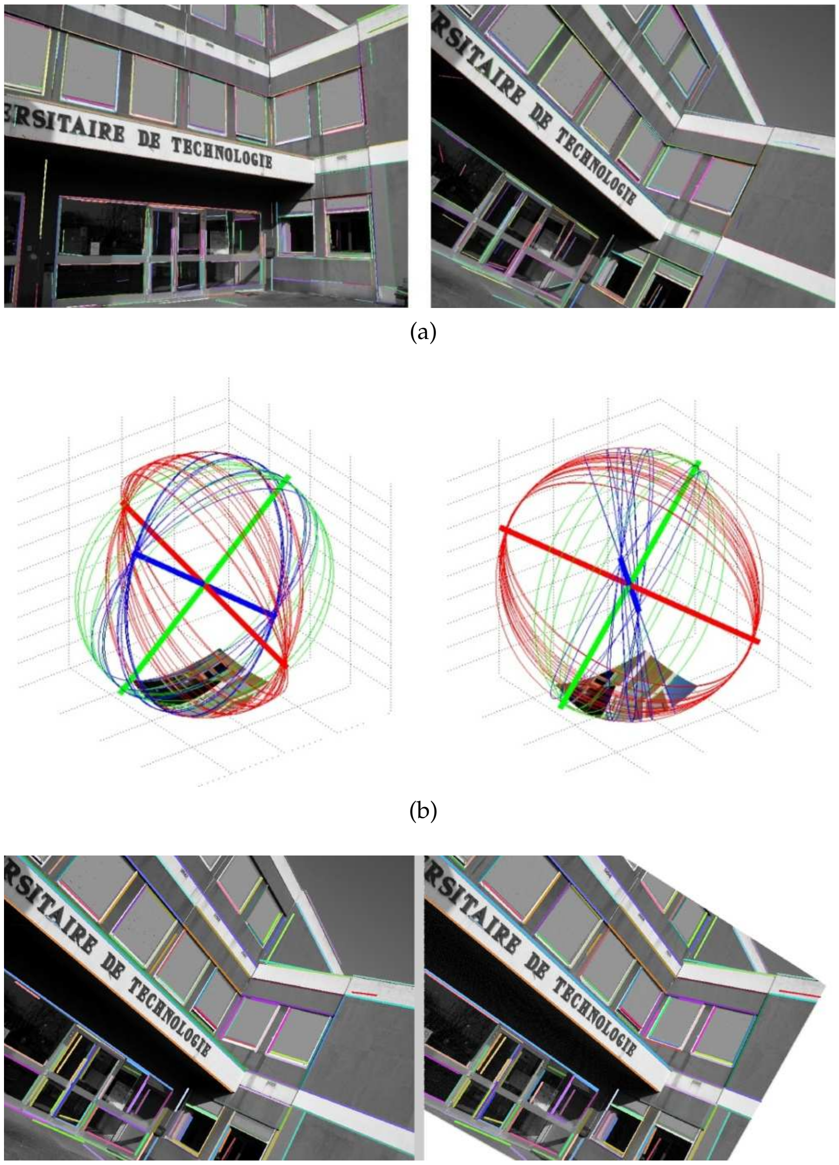


(a)

(b)

(c)

Figure 18: (Top) Two paracatadioptric images and (Middle) their extracted segments. Bottom: matched lines (each color represents one correspondence. All of the 121 matches shown are correct. Note that segments a, b, c and d share the same great circle (dashed line). The end points of each segment are used to find the correct correspondence.



(a)

(b)

(c)

Figure 19: (a): Two perspective images and their extracted segments. (b): Related great circles onto unitary sphere and three dominant vanishing directions (75 % of lines are hidden). (c): Matched lines (each color represents one correspondence). Note that the image on the left is the second view and the image on the right is the first view after applying the R on it. 93% of 129 matches shown are correct.

4.4.3 *Real omnidirectional images*

Figure 20 shows the result of applying our algorithm on two real fish-eye images taken while trying to avoid translating the camera. Figure 20(b) shows 2 vanishing directions and their corresponding great circles. For this example, the numbers of extracted segments are 376 and 415 for the left and right images respectively. The preset value for the tolerance is two degrees. Obtained matches are shown at the bottom row of the Figure. 107 out of 119 matches are correct. 12 mismatches are found where 2 or more line segments are located very close to each other.

4.4.4 *Fusing different central images*

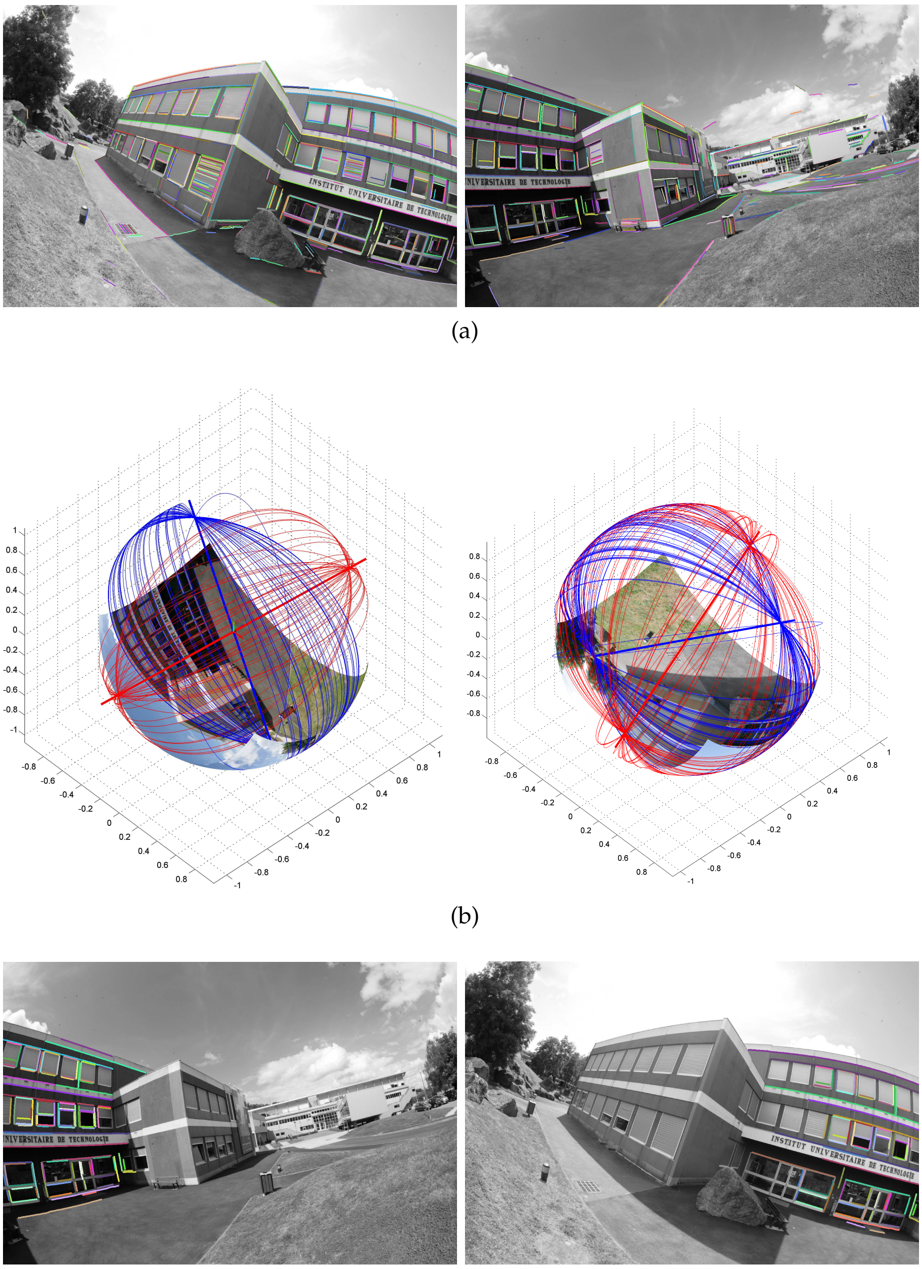
As it was mentioned before, the algorithm 4.3 is based on unit sphere which means it is independent of the type of images and it works as long as the images are calibrated. For demonstration, we have applied our algorithm to establish correspondences between a real fish-eye image and a real perspective image taken from the same point of view (Figure 21). For this example, the number of extracted segments is 440 and 415 for the perspective and fish-eye images respectively. 112 out of 149 matches are one-to-one correct matches and the rest are one-to-many or many-to-many matches due to close line segments.

4.5 other applications

Due to weak parallax, two consecutive short baseline images are of no use for 3D reconstruction. However having a sequence of such images one can reconstruct the common part of the scene between the first and the last image. Other than matching lines between two central images, the algorithm presented in this chapter can be used to track line segments between consecutive images and eventually establish some useful segment correspondences between the first and last images having enough parallax to be used for motion estimation and reconstruction methods such as [153] (an optimized version of this algorithm will be presented in chapter 6).

Another application is creating high resolution panoramic images from high resolution perspective images taken by rotating a perspective camera around its center of projection and shooting photos, such as the images in Figure 22. Note that even though omnidirectional imaging systems such as catadioptric cameras can be used to construct such images, the resulting panoramic image suffers from non-uniform low resolution. Also note that here we only use the algorithm to recover the rotation between images in order to align them and we do not need really to establish any correspondences. In fact here we use line matching as a measure of aligning images by correctly matching vanishing points.

Figure 23 shows a schematic of the method with real panoramic reconstruction. The images were taken with some overlap between consecutive ones but no effort

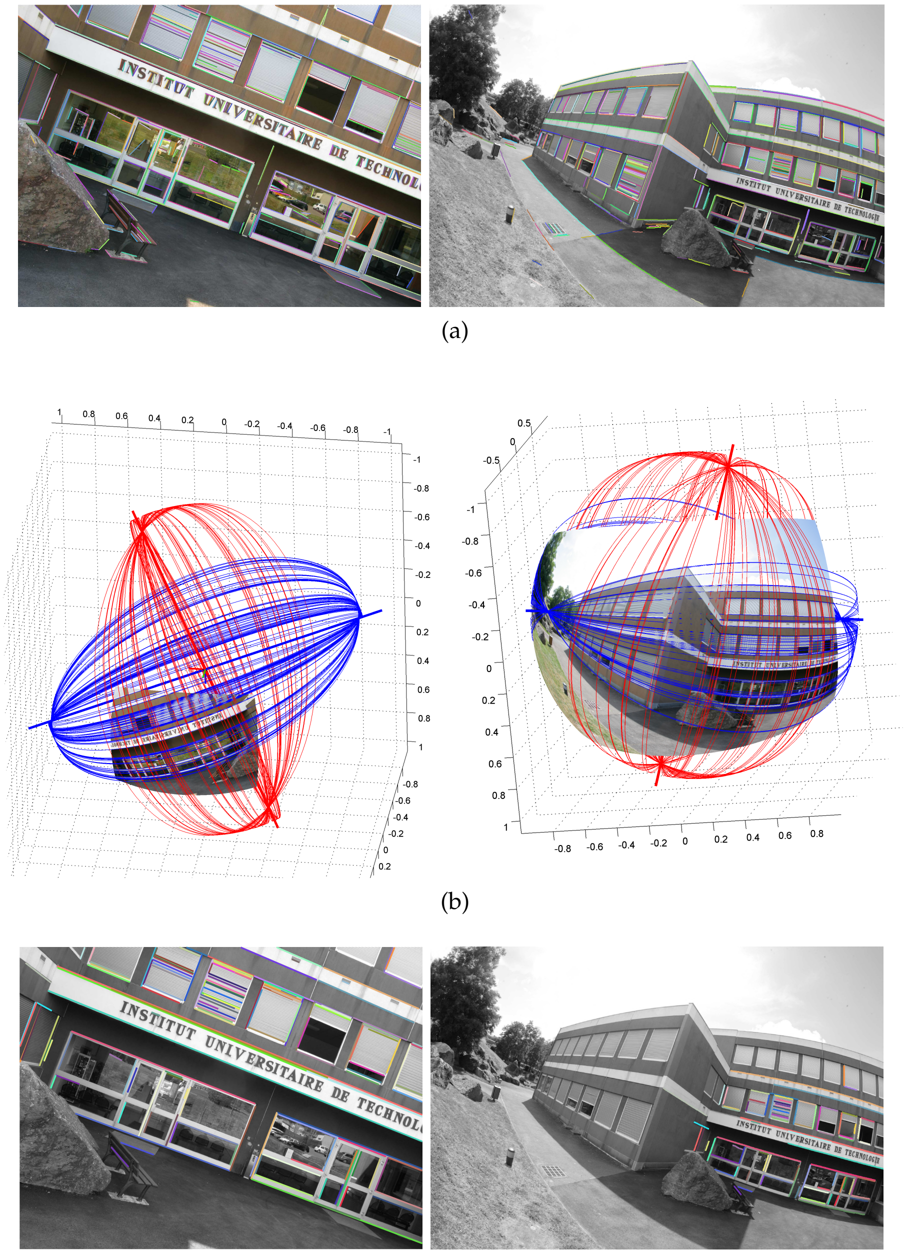


(a)

(b)

(c)

Figure 20: (a): Two fish-eye images and their extracted segments. (b): Related great circles onto unitary sphere and two dominant vanishing directions (50 % of lines are hidden). (c): Matched lines. Each color represents one correspondence.



(a)

(b)

(c)

Figure 21: (a): A fish-eye and a perspective image and their extracted segments. (b): Related great circles onto unitary sphere and two dominant vanishing directions (for the fish-eye image, more than 66 % of lines are hidden). (c): Matched lines between two images. Each color represents one correspondence.

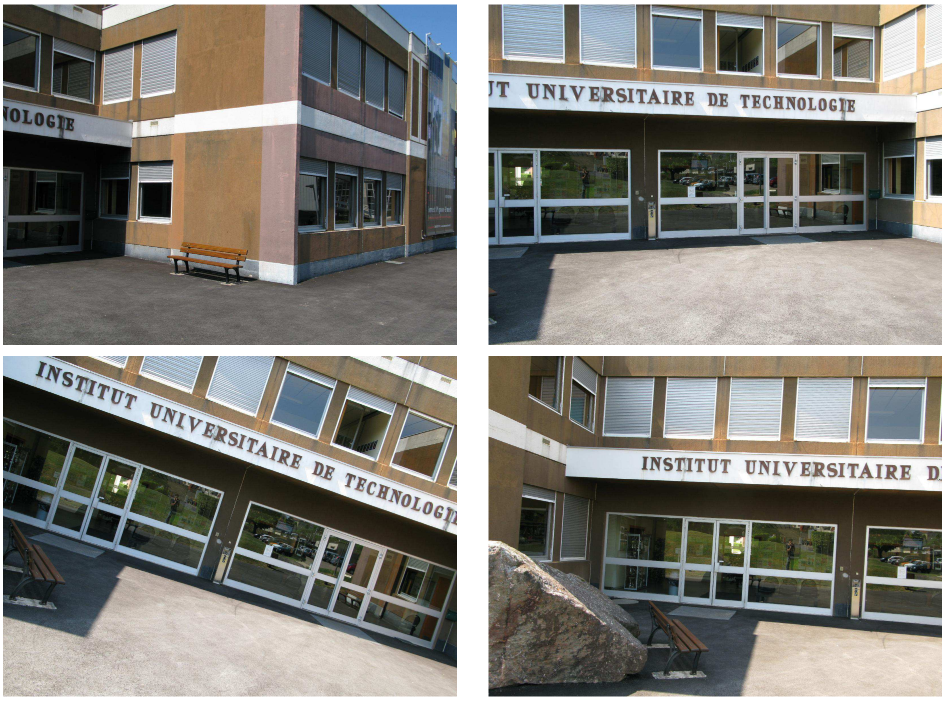

OLOGIE
UT UNIVERSITAIRE DE TECHNOLOGIE
INSTITUT UNIVERSITAIRE DE TECHNOLOGI
INSTITUT UNIVERSITAIRE D

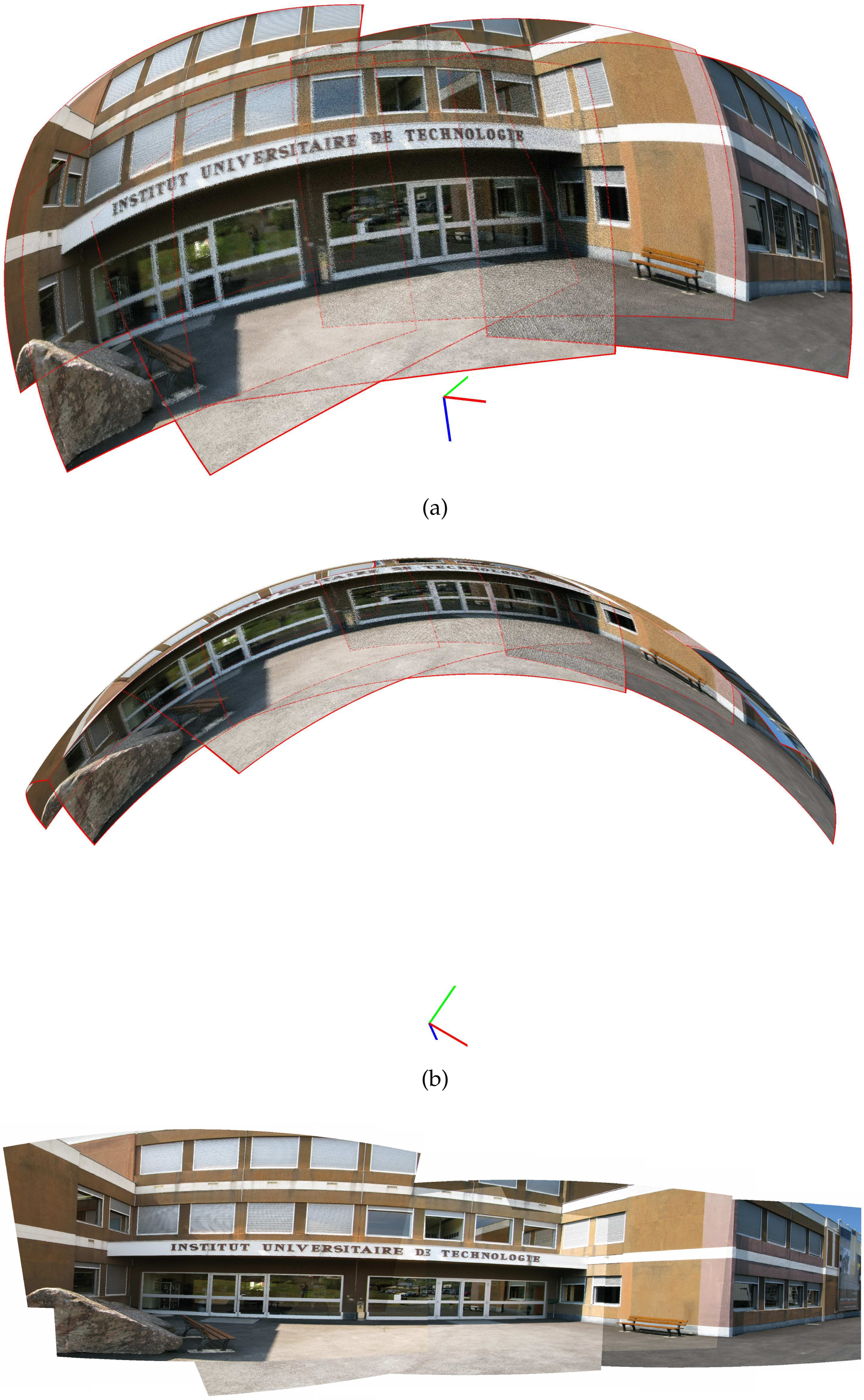
Figure 22: Four among six high resolution perspective images taken by rotating a perspective camera around its center of projection.

was made to keep the axis of rotation fixed (which is inevitable in a relaxed photography).

4.6 CONCLUSION AND OUTLOOK

This chapter dealt with the problem of matching lines for all types of central imaging system under a short baseline motion by presenting a generic and simple line matching approach. The method is composed of two main steps of extracting line segments and estimating vanishing directions followed by simultaneously recovering the rotation R and matching lines. Also, two methods for retrieving R, one based on matching vanishing points and the other based on matching any two feature points were proposed. Finally, various experimental results on both synthetic and real images taken by different central cameras as well as an application of the algorithm for creating high resolution panoramic images from high resolution perspective images were also presented.

The state of the art line matching methods use demanding techniques (for example using epipolar geometry [115]) to match lines between images with short baseline and due to deformation of omnidirectional images, they do not even work on these kind of images at all. On the other hand, we developed a very simple and intuitive method which is generic and it works for both perspective and omnidirectional images. It is based on the fact that the motion of the system for a short base-line movement is mainly a rotation and in constructed scenes,



(a)

(b)

(c)

Figure 23: (a,b) Stitching all images on the Unitary Sphere, (c) an unwrapped cylindrical image.

the rotation can be estimated by matching vanishing points which are easily available in such scenes.

5 FAST LINE MATCHING BETWEEN DISPARATE VIEWS OF PLANAR SURFACES

In the previous chapter, we developed some tools based on lines extracted from two images taken with short baseline. Now, we will relax the assumption of short baseline and aim to develop a generic method of line matching between two views of a constructed scene, no matter what the motion of the system is. As usual, we first start by a simplified problem where the images are perspective and the scene is planar. We address the problem of matching randomly oriented lines parallel to a scene plane. We do so by finding the vanishing points of each line by intersecting the line on the image plane with the line at infinity of the plane. The matching algorithm is then based on looking for the lines in both images which follow the infinite homography. We also use vanishing points to calibrate the camera, assuming a natural perspective camera. We then move on to a more generic scenario where the scene is constructed of more than one plane. Here we face a more complex problem. Hoping to solve some difficulties of formulating the problem on the image plane and following our aim to expand the method to all central images, we then formulate the problem using unitary sphere and we show that the new algorithm can perform fairly well for planar scenes such as aerial images.

5.1 RELATED WORK

Most approaches for solving the correspondence problem are based on metric information, such as topological arrangements of points, line orientation conservation, etc. which are not preserved under perspective projection and therefore they only work for images that have been taken under short baseline motion [57, 76, 151]. For long baseline motion different methods exist which are either based on the prior knowledge of some geometric constraints in the scene such as known projections of four corresponding coplanar points [39] or known epipolar geometry [115] or based on some perspective invariants (quantities that are invariant under perspective viewing). It is well established that if a scene consists of an arbitrary set of points or line segments in 3D and it goes under a general motion then there are no invariants of its image under projection [21]. Therefore, some assumptions regarding the structure of the viewed scene have to be made to gain suitable projective invariants. The most common assumption made in the literature is that the features to be matched lie on one or more planes in the scene [90, 109, 71]. Our method also benefits from a quantity which is invariant to the motion of the camera and it is applicable as long as there is a plane (real or imaginary) in the scene which is parallel to all lines. In this work, we present a novel and fast approach for finding the correspondence of two sets of line segments which are images of a set of 3D lines parallel to a 3D plane in the scene. The assumption we make is that there are two sets of parallel lines among these lines.

Until now, only a few methods for automatic line segment matching for wide baseline stereo exist [115, 130, 45, 12, 139, 138] (see table 3 , chapter 3). Although not really in the context of matching but homography estimation through matching points and lines together, the method developed by Lourakis et al. [71] is of interest to us since it also particularly applies to planar scenes. However, the first important difference between the method presented here and their method is that for our method to work it is not necessary that the lines be located on the scene plane and it suffices if they are parallel to it whereas their methods only works for completely planar scenes. Secondly, for their geometric constraint (so called two-line two-point ($2\mathcal{L}2\mathcal{P}$) projective invariant) to work, they need both points as well as lines to match two planar views while our method is purely a line matching approach.

5.2 PROPOSED METHOD

Our approach of matching lines starts with computing the rotation between two views using two vanishing point correspondences as was already explained in the previous chapter, section 4.2.2.1. We then compute the infinite homography, H_∞, between two views by means of the rotation and the camera intrinsic parameters. Knowing the infinite homography, we are then able to match lines based on the symmetric homographic transfer error of their vanishing points between two images as will be shortly explained in following sections. For the complete description of the vanishing point, the line at the infinity and the absolute conic refer to [52]. Here, some basics will be recalled.

A scene line intersects the plane at infinity π_∞ in a point and the image of this point on the image plane is the vanishing point of the line. Similarly, parallel planes in 3-space intersect π_∞ in a common line, and the image of this line is the vanishing line of the plane. Since the lines parallel to a plane intersect the plane on π_∞, it is easily seen that the vanishing point of a line parallel to a plane lies on the vanishing line of the plane. Therefore, given a line in 3D space parallel to a plane, the intersection of the image of the line on the image plane (a segment) with the vanishing line of the plane results in the vanishing point of the line as it is shown in Fig. 24. Two sets of parallel lines A and B which are also parallel to the plane π in 3-scene intersect the plane at infinity at V_A and V_B. The vanishing line of the plane on each image plane is the line connecting the images of these two points. The vanishing point of any other 3D line L parallel to the plane can then be found by intersecting the vanishing line and the image of the 3D line. This is the clue which we exploit to find the image of the intersection of each line with the plane at infinity (the vanishing point of the line on each image plane) followed by establishing correspondences by verifying weather each pair of putative vanishing point correspondences (e.g. line correspondences) are following the infinite homography between two views.

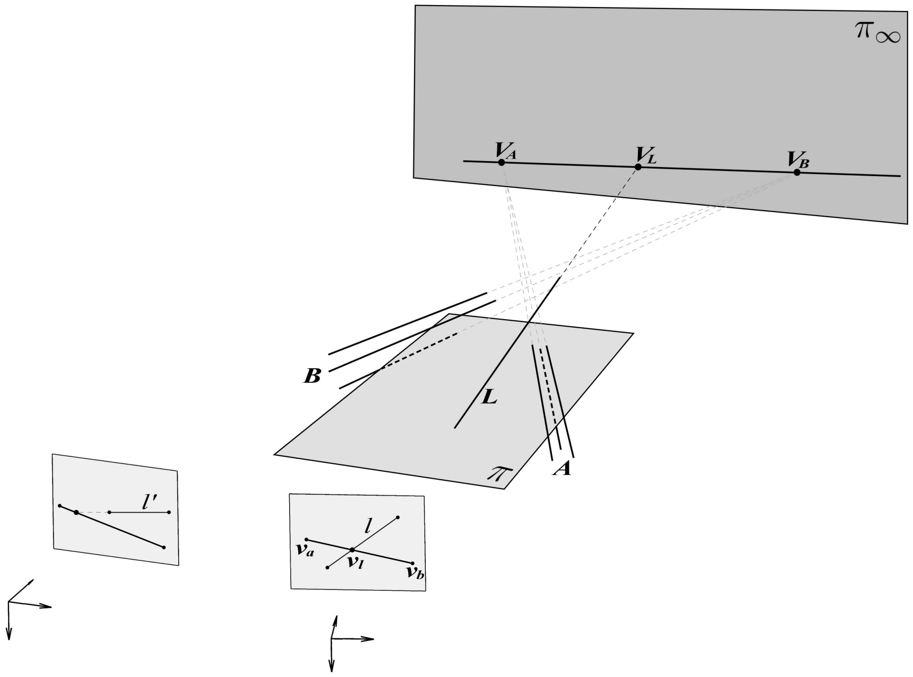


Figure 24: Intersections of the plane π and lines parallel to it with the plane at infinity and their images on the image planes.

5.2.1 *Estimating the infinite homography:*

The infinite Homography H_∞ encloses the cameras' parameters and the rotation part of the motion between two positions of the cameras:

$$H_\infty = K'RK^{-1} \tag{5.1}$$

Having estimated the cameras' intrinsic parameters (K and K') and having extracted and matched some vanishing point correspondences (Appendix A.3), the relative rotation between two views can be computed using the simple linear method of [16] which was also already described in previous chapter, section 4.2.2.1.

5.2.2 *One scene plane*

We match lines based on the symmetric homographic transfer error of their vanishing points between two images. Let us consider the example illustrated in Fig. 25. The line v_1v_2 (respectively $v'_1v'_2$ in the second image), is the line at infinity of a scene plane and $H_\infty^T l'$ and $H_\infty^{-T} l$ are homographic transformations of line segments l and l' between two images by the estimated homography H_∞. If l and l' are images of the same 3D line which is parallel to the scene plane, then i, the vanishing point of the line, can be found by intersecting the segment and the line at infinity:

$$i = \tilde{i}_{(1:2)}/\tilde{i}_{(3)}\ , \quad \tilde{i} = (\tilde{v}_1 \times \tilde{v}_2) \times l \tag{5.2}$$

where $\tilde{i}_3$ is the third element of $\tilde{i}$. i_h, $i'$and i'_h are found in the similar way. The symmetric homographic transfer error can now be computed as:

$$\triangle_{sht} = \sum d(i, i_h)^2 + d(i', i'_h)^2 \tag{5.3}$$

where $d(.,.)$ is the Euclidean distance between two points in the image. Note that if the line segment l is a true match for l' and their pre-image (a line in 3D space) is parallel to the plane, i and i_h (and $i'$and i'_h on the second image) should coincident and the symmetric transfer error should be zero. However due to the noise and the error in the detection of the lines and estimating the infinite homography, the transfer error can be up to several pixels. If two segments are not corresponding or their pre-image is not parallel to the plane, one expects a large transfer error. Depending on the expected noise, a suitable threshold, *tol* (in pixels), is selected and a match is accepted if its transfer error is less than the threshold. For the simulation with images of size 800x600 and zero mean noise with 2 pixel std, this threshold was set to 4 pixels.

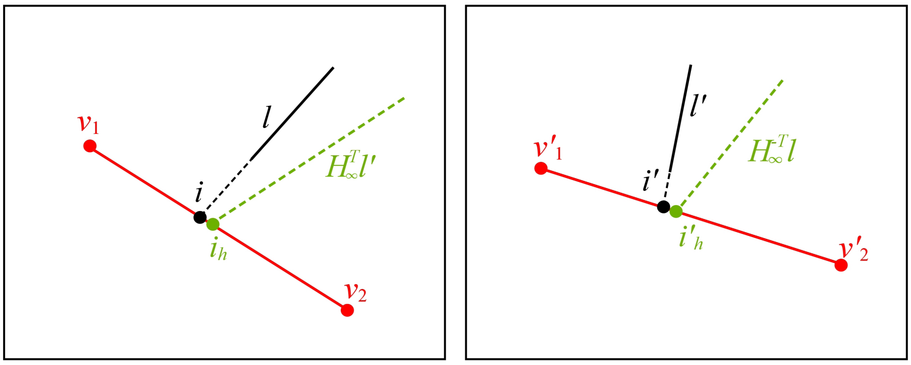


Figure 25: The definition of the symmetric homographic transfer error. Refer to the text for the explanation.

5.2.3 *More than one scene plane*

If more than one scene plane is detected (i.e more than two vanishing points are available e.g. in Manhattan scenes), the algorithm works as before except now the transfer error should be calculated for each plane. Consider Fig. 26. Three vanishing points are corresponding to three main Manhattan directions. Each two main directions define a plane in the scene. Note that if the line segment

l is a true match for l' and their pre-image is parallel to one of these planes, the symmetric transfer error should be small and less than the threshold. On the other hand, if two segments are not corresponding or their pre-image is not parallel to the plane, one expects a large transfer error. If a line does not have a match in the other image or its pre-image is not parallel to any of the scene plane, its symmetric transfer error with any of the lines in the second image will be large and it should be categorized as unmatched.

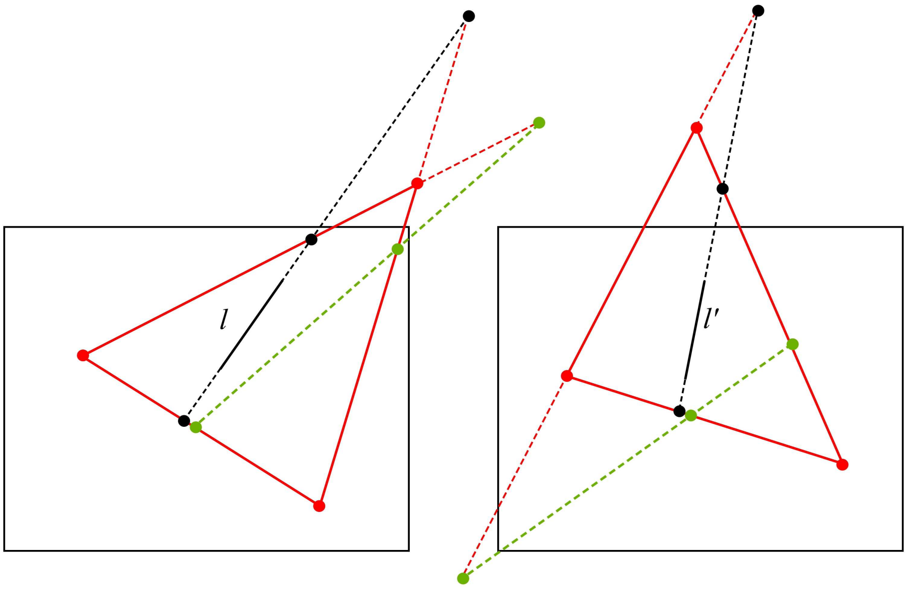


Figure 26: The distance between green and black points of intersection with each line at infinity is directly related to the symmetric transfer error. Refer to the text for the explanation.

5.2.4 *The first algorithm*

The proposed method is composed of the main steps depicted in algorithm 5.1. The algorithm starts by extracting vanishing points that are assumed to exist among extracted lines on the image plane (we used the technique presented in [127]). As the set of putative matches, it then considers each line in one image as a potential match for each and every line in the other image. The correct matches are then identified and kept when their symmetric homographic transfer error is in the order of a few pixels.

If a line is involved in more than one match, the match with the minimum transfer error is kept and the rest of the matches are removed. However, note that when matching line segments, it occurs that one segment matches with more than one segment in the other image due to existence of fragmented segments (line segments split into several smaller, more or less collinear line fragments) and the segments which are the images of parallel lines in the scene (e.g they form an identical intersection with the line at infinity because they are parallel so

Algorithm 5.1 The first proposed algorithm

```
Given two perspective images and the preset tolerance $tol$;
Extract their line segments and extract all dominant vanishing points among the extracted lines;
Calibrate the camera using these vanishing points (subsection 4.3.1) ;
$Sol \leftarrow$ All possible matching solutions between vanishing points;
for each $Sol(i)$ do
    Estimate $R$ (section 4.2.2.1 ) and then compute $H_\infty$(Equ. 5.1);
    $Matches = []$;
    for each selection of two vanishing point $v_1$ and $v_2$ (respectively $v'_1$ and $v'_2$ in the second image) do
        for each line $l$ from the first image do
            for each line $l'$ from the second image do
                if $\triangle_{sht} < tol$ (Equation 5.3) then
                    $Matches \Leftarrow Matches + [\, l \rightleftharpoons l' \,]$;
                end if
            end for
        end for
    end for
    $M(i) \leftarrow Matches$;
    $S(i) \leftarrow$ size($Matches$);
end for
$j \leftarrow$ index(max($S$));
return  $M(j)$ and $Sol(j)$;
```

they will intersect in the same point on the line). In both cases, the intersection of the segment with the vanishing line of the plane lies on (very close) to that of another segment (in other words, they have the same (very close) transfer errors). Therefore also in this case, a line is involved in more than one match. To take care of this particular case, if a line is involved in more than one match, the match with the minimum transfer error and all the matches which have a difference in transfer error within a certain tolerance region are kept and the rest are removed.

Fragmented segments have to be collinear in both images, therefore we can furthermore separate fragmented line segments from disparate parallel lines by checking co-linearity of the segments in both images. Note that it is important that two line segments be collinear in *both* images since these segments may just be collinear accidentally in one image while belonging to two different lines. This is, however, unlikely to happen in both images.

For the parallel lines, however, computing intra-set correspondences in two set of matched parallel lines with the constraint presented in this thesis is not possible and we need to apply other constraints on these lines. We use sidedness constraint as described in [30]) and later expanded in [12] which states that, for a triplet of feature matches, the center of the first feature should lie on the same side of the directed line going from the center of the second feature to the center of the third feature. For a line feature, the center of the feature is defined as its midpoint. If the set contains a pair of line matches, the midpoint of the first segment must be on the same side of the second line segment in both views.

Also, when matching line segments, there is another group of lines which do not have any correspondences (e.g. they are not detected in the other image) or they are not parallel to the scene plane. Note that these segments interfere with the matching process by increasing the number of one-to-many matches. The effect of these lines on the performance of a simplified version of the algorithm will be analyzed more through simulation in section 5.3.6.

5.2.5 *Discussion*

By performing some simulations, we found that if the algorithm outlined above is used to match lines between two images of a scene constructed from planar surfaces, then the results are perfect as long as the data is noise-free. However, in practice, this is never the case, and the matching results are not good in the presence of noise. To trace the roots of the problem, we investigated the effect of noise on the transfer error for a given match and we found two main reasons. The first cause is related to the number of lines to be matched. The effect of noise is to perturb lines to lines that their directions lie close to the correct lines, increasing the number of one to many or many to many matches. This effect will be more serious when the number of lines is high. The second reason is better depicted in Figure 27.

If the segment on the image plane forms a small angle with the line at infinity, or the distance of the segment to the line at infinity is long, even a minor error in

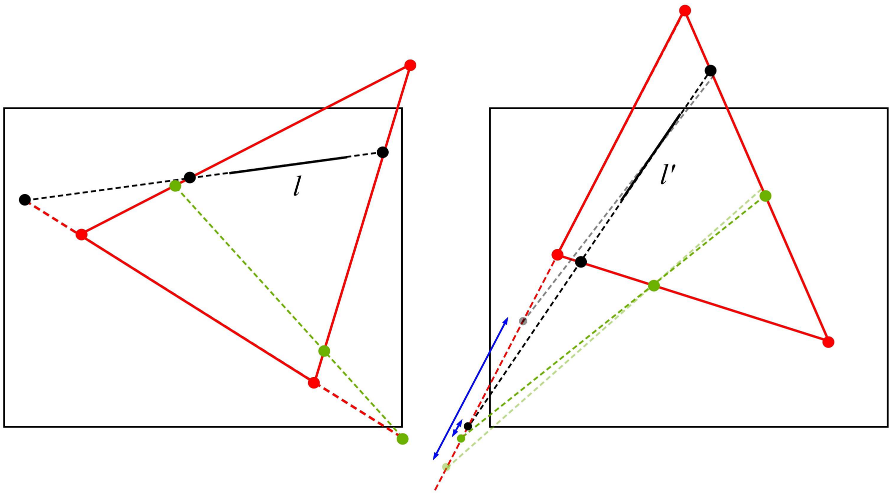
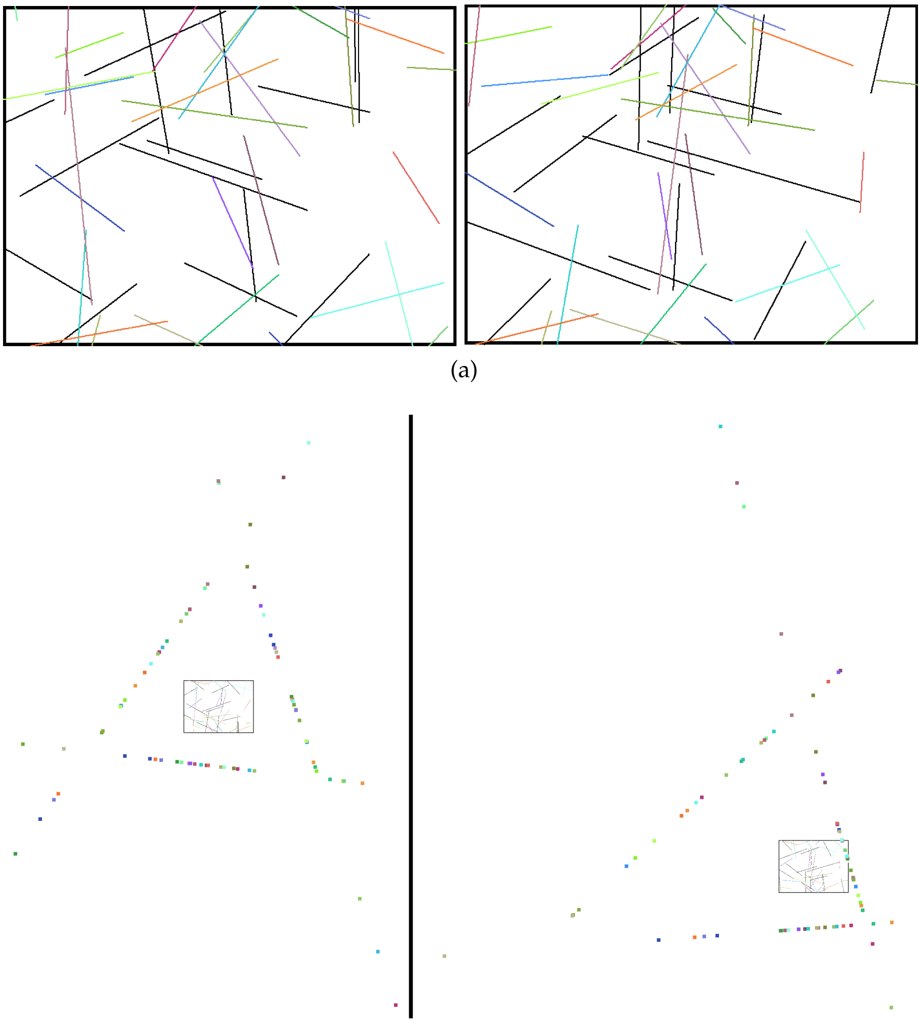


Figure 27: Small angle between a segment (or its homographic transformation from the other image) and the line at infinity can dramatically magnify the effect of error in the extraction of lines on the homographic transfer error.

the orientation of extracted segment can largely move its vanishing point along the line at infinity. The same is true for the homographic transformation of the segment from the other image. As a result, the total symmetric homographic transfer error can be very large and the putative match is rejected even if it is a correct match. This situation can happen very frequently which means the computed transfer error for a large number of real matches is not resembling the true similarity. See Figure 28 for an example of how vanishing points are vastly scattered around the image. The position and the orientation of the each line at infinity depends on the orientation of its plane in 3-scene. If the angle between the 3D plane and the image plane is small, the line at infinity can be located very far outside the image boundary since this line is in fact the intersection between the image plane and a plane parallel to the 3D plane and passing through the focal point. One way to overcome this problem is to consider only the angle between two lines connecting the center of the segment to its vanishing point and its transformation from the other image as error distance. Another way is to normalize the transfer error by the distance of the center of the segment to the line at infinity along the line. We tried these solutions and some other approaches but none of them improved the performance of the algorithm. Working on the unitary sphere, however, helped us to develop a better algorithm (at least for the case of one scene plane) as will be presented in the rest of this chapter. Working on the unitary sphere has many advantages: no need for normalization, a generic domain for all central imaging systems and easier extraction of vanishing points are among them.



(a)

(b)

Figure 28: (a) Projection of synthetic lines on two image planes, (b) Their vanishing points along lines at infinity.

5.3 LINE MATCHING ON THE UNITARY SPHERE

Besides the problems mentioned above, the proposed algorithm is only applicable on perspective images. Aiming to solve these difficulties and following our goal to expand the method to all central images, we then formulated the problem using the unitary sphere. In this domain, lines (curves) on the perspective (omnidirectional) image plane change to great circles on the unitary sphere. Vanishing points change to unit vectors and instead of H_∞, they are mapped by R from the first image to the second one.

In the rest of this section, by line n_l we mean the normal vector of the great circle plane corresponding to the line segment l projected on the unitary sphere, and a point is meant to be the unit vector corresponding to projection of a pixel point from image plane onto the unitary sphere.

5.3.1 *The second algorithm*

We match lines parallel to each plane based on their angles with the direction corresponding to one of the vanishing points of the plane. The algorithm starts by first extracting vanishing points. Assuming line n_l, the line that we like to find its corresponding in the other image, is parallel to a plane defined by two vanishing points V_A and V_B (we will soon relax this assumption), V_l, the vanishing direction of n_l can be computed by:

$$V_l = (V_A \times V_B) \times n_l \tag{5.4}$$

One of the vanishing points, namely V_A is then chosen as the reference and using the cosine formula, the angle between the line and the reference direction is found:

$$\theta = \arccos\left(\frac{V_A^T V_l}{\sqrt{V_A^T V_A}\sqrt{V_l^T V_l}}\right) \tag{5.5}$$

These steps are repeated for all line segments in each image in order to compute the angle between each line and the reference direction in 3D.

Having the angle between each line and the reference direction, the matching is very easy now. Given a line segment and its angle with respect to the reference direction in the first image, the corresponding line segment in the second image should hence have the same angle (inside a reasonable tolerance to compensate for the noise) with the reference direction in the second image. In other words, if the difference between two angles θ and θ' corresponding to two line segments l and l' from first and second image respectively (called Δ_θ hereafter) is less than a preset tolerance, tol_1 (in radian):

$$\Delta_\theta = |\theta - \theta'| < tol_1 \tag{5.6}$$

where |....| stands for absolute function, then the match is considered a correct one. Figure 29 visually demonstrates the principale of the method. Two red and blue directions are corresponding to two main vanishing directions of the scene plane and the black great circle is the line at infinity of the plane which also passes through these two vanishing points on the unit sphere. Consider any other line projected on the unit sphere. Here two such lines are represented by their great circles and by lawn green and light steel blue colors. It can be shown that the angle between the intersection of the line with the black great circle (marked by a cross here) and one of the vanishing direction (for example the red one) is the same and independent of the motion between two unit spheres (two cameras).

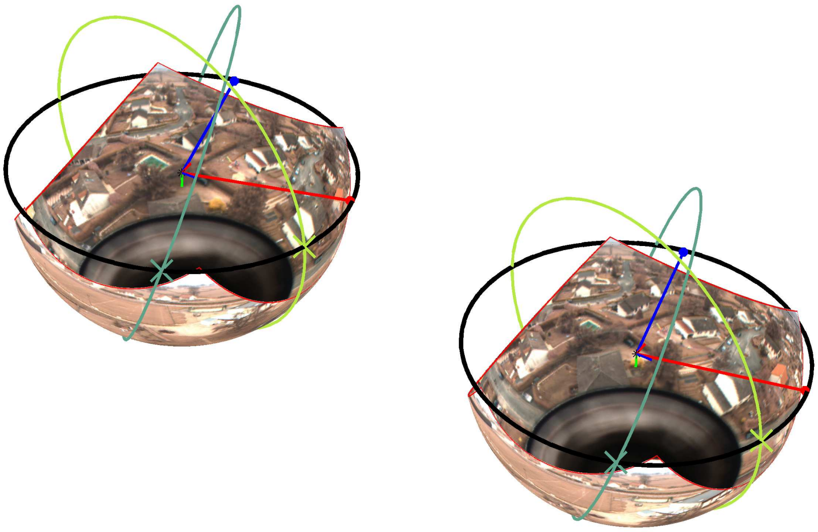


Figure 29: Two aerial catadioptric images projected on the unit sphere. Refer to the text for the explanation.

The assumption that all lines are parallel to a plane defined by two vanishing points V_A and V_B (while actually many of them are parallel to other planes defined by other pairs of vanishing points), will cause many mismatches. The correct matches are identified and kept by checking whether they are really parallel to the plane by computing their angles with the plane. For each given match $n_l \rightleftharpoons n_{l'}$, the direction of their pre-image in 3D, L can be computed by:

$$L = n_l \times (R^{-1} n_{l'}) \tag{5.7}$$

since L lies on the intersection of two planes defined by n_l and $R^{-1}n_{l'}$ passing through their corresponding camera centers and this intersection is independent of the location of the planes (cameras) in 3D space and it depends only on the relative orientation of two planes (two camera frames). In other words, L is independent of the translation of the second camera w.r.t to first camera and it only depends on the rotation between two views. Now if l and l' are images of the same 3D line which is parallel to the scene plane, then L and the normal vector of the plane should be orthogonal:

$$(V_A \times V_B) \boldsymbol{.} L = (V_A \times V_B) \boldsymbol{.} (n_l \times (R^{-1} n_{l'})) = 0 \tag{5.8}$$

where $(\boldsymbol{.})$ stands for dot product. Due to the noise, this equality will never be satisfied and we need to define an angular tolerance:

$$\Delta_{orth} = \frac{\pi}{2} - \arccos\left(|(V_A \times V_B) \boldsymbol{.} (n_l \times (R^{-1} n_{l'}))|\right) < tol_2 \tag{5.9}$$

The proposed method is composed of the following main steps:

Algorithm 5.2 The second proposed algorithm

```
Given two images taken by a central imaging system and the preset tolerances
  $tol_1$ and $tol_2$;
Extract their line segments and extract all dominant vanishing points among
  the extracted lines;
$Sol \leftarrow$ All possible matching solutions between vanishing points;
for each $Sol(i)$ do
    Estimate $R$ (section 4.2.2.1);
    $Matches = []$;
    for each selection of two vanishing point do
        for each line $l$ from the first image do
            for each line $l'$ from the second image do
                if $\Delta_\theta < tol_1$ and $\Delta_{orth} < tol_2$ (Equations 5.6 and 5.9) then
                    $Matches \Leftarrow Matches + [\, l \rightleftharpoons l' \,]$;
                end if
            end for
        end for
    end for
    $M(i) \leftarrow Matches$;
    $S(i) \leftarrow$ size($Matches$);
end for
$j \leftarrow$ index(max($S$));
return $M(j)$ and $Sol(j)$;
```

5.3.2 *Special cases*

If a line in 3D and two camera centers are coplanar, then any 3D line on their plane can be the pre-image of projections of the line on two images, including the line at infinity of the plane. This line at infinity, if in general position, will intersect three line at infinity of the 3 scene planes in 3 points and therefore these two segments can be correctly matched by each of three planes. In other words, the 3D line seems to be parallel to all 3 orthogonal planes! This special case is not a problem for the algorithm and the result is even a more robust match. As a remark note that the reconstruction of the line is not possible since there exist no parallax.

5.3.3 *Discussion*

Even though by projection on the sphere, the shortcomings of the previous method was handled, the new algorithm still suffers from some inevitable limitations which hold true for the previous algorithm working on the perspective image plane as well as the new generic one. These limitations can be better understood through some simulations in order to evaluate the performance of the method with respect to noise level.

The simulations were carried out on configurations of random line segments as shown in Fig 30. For a realistic simulation, throughout all experiments, we assume calibrated virtual cameras with the calibration results of a real camera with image size of 704x528 pixels, effective focal length of 710.34 pixels and the principal point at (340.74,259.42). Each pixel is a square with the size of 0.0014 millimeter. The camera has a random motion (rotation and translation) and we only admit motion so that the scene cube can be seen from both camera poses.

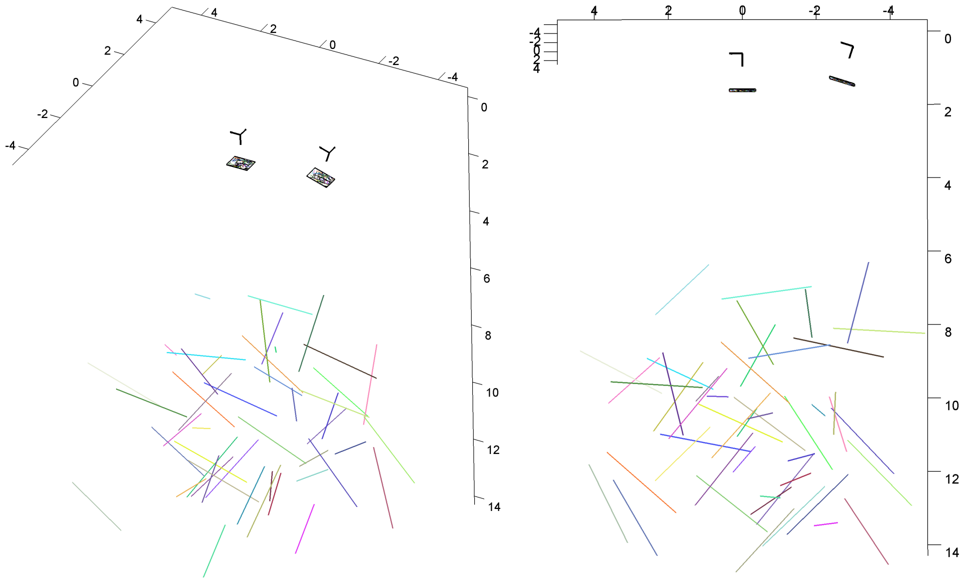


Figure 30: The simulation configuration. The position and orientation of the lines in 3D are random but parallel to 3 planes with also random orientations. Cameras and image planes are drawn 100 times bigger for a better visualization.

All lines are randomly oriented parallel to 3 imaginary planes and among the lines parallel to each plane, there are two groups of lines which are also parallel to each other which are used for estimating two vanishing directions for each plane. The number of lines in each group is fixed and equal to 10 for all experiments. Finally, two segments are considered matched if the difference in their angle with the reference direction in 3-scene is less than 2 degrees and also their pre-image in 3D is parallel to the plane inside a tolerance region of 2 degrees.

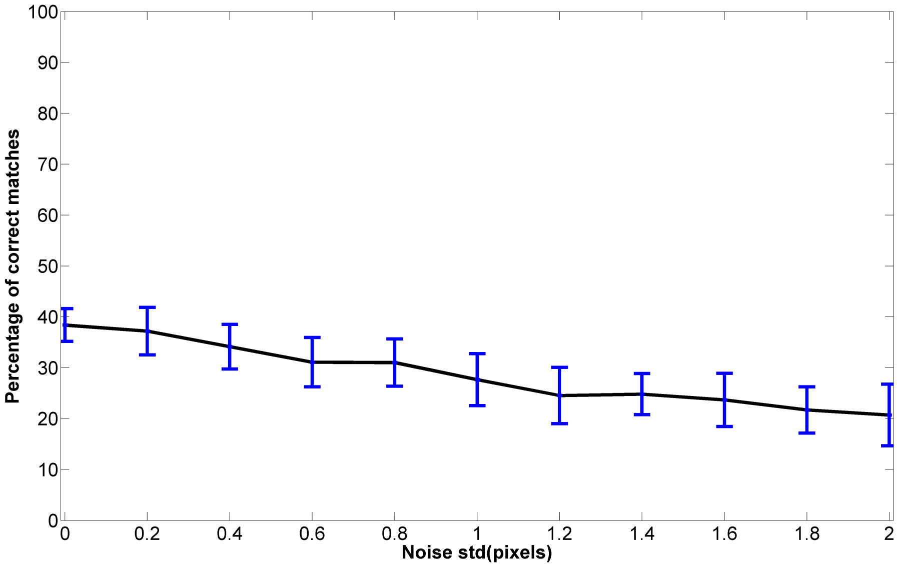


Figure 32: Dependence on noise level. Noise levels are reported in terms of the standard deviation of a zero mean Gaussian.

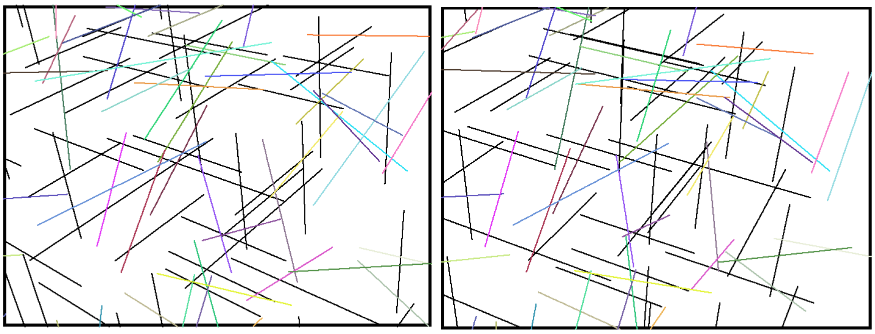


Figure 31: Projection of synthetic lines on two image planes. The black lines belong to the sets of parallel lines while the colored lines are randomly oriented.

The endpoints of the line segments are perturbed by a zero mean Gaussian noise with different values of standard deviation, σ . We do so independently in the x and y directions and we also only admit noise between -3σ and 3σ and then quantize the points to the nearest pixel. We vary noise from 0.0 to 2.0 pixels, simulating a quite noisy image condition. For each noise level, we generate 100 trials and in each trial, we generate 45 lines in 3 sets of 15 lines parallel to each plane (excluding 30 lines used for estimating three vanishing directions). We also generate 5 lines which are not parallel to any planes.

The result of the simulation is shown in Fig. 32. Surprisingly, even with noise-free data, not more than 40 percent of matches are correct. The constraint for checking parallelism is not enough discriminative due to the high possibility that the pre-image of many random non-real matches can be also parallel to one of the planes and this ambiguity is unresolvable.

Table 4 shows part of matching result of the algorithm where identical line numbers are true matches. Note that matches $2 \rightleftharpoons 4$, $3 \rightleftharpoons 7$, $4 \rightleftharpoons 2$, $7 \rightleftharpoons 10$ and $14 \rightleftharpoons 2$ are wrong matches though they satisfy both tolerance regions of 2 degrees. Tightening the tolerances can reduce the number of mismatches when the noise present is weak but it can also decrease the number of matches when the noise is considerable (In fact, if the data is noise free, 100% correct matches are guaranteed by setting both tolerances close to zero)

Line no in first image	Line no in second image	Δ_θ	Δ_{orth}
1	1	0	0
2	2	0	0
2	4	0.6	2.2
3	3	0	0
3	7	1.6	2.7
4	2	0.6	0.9
4	4	0	0
5	5	0	0
7	10	2.3	2.5
14	2	1.9	2.2

Table 4: Part of matching result of the proposed algorithm. Refer to the text for the explanation.

The tolerance values for the above algorithm are functions of the expected noise and finding an analytical relation for obtaining these values with respect to the expected noise is not an easy task. Even if these values are set so that the percentage of correct matches is in its highest, still the number of mismatches would be high due to the complexity of scene (i.g. more than one dominant plane). For example, applying the algorithm on a simple scene such as the scene shown in figure 33 (composed of three planes) results in 237 matches (though there are on total 96 lines to be matched) where less than 10 percent are correct. This means that the number of one to many and many to many matches is high.

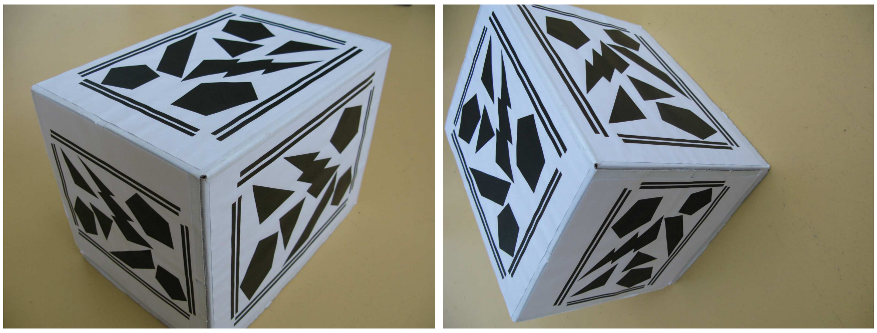


Figure 33: Example of a complex scene consisting of more than one plane. Applying the algorithm on these images results in 237 matches where only less than 10 percent are correct.

Therefore, we decided to evaluate and apply the algorithm assuming the scene is planar such as aerial images. The algorithm is essentially the same as above algorithm except that if more than two vanishing points are detected, any two of them can be used to estimate R (since this vanishing points are located on the same line at infinity and therefore they should be collinear).

We continue with performing some simulations for determining the performance of the method with regard to the noise level, the number of lines and the ratio of non-parallel lines to the total number of lines. By non-parallel lines we mean the lines which are not parallel to any of the scene planes.

5.3.4 *Simulation 1 (Dependence on noise level)*

This simulation experiment was designed to determine how the accuracy of the matching would vary as the amount of error in the extraction of the line segments is increased. The same procedure as explained in previous section is used to generate random line segments and add noise on their end points, except now it is assumed that there is only one dominant scene plane and a line is either parallel to this plane or intersecting it. For each noise level, we generate one set of 50 lines parallel to the plane. We also generate 5 lines which are not parallel to the plane. The preset value for both tolerance regions is 2 degrees.

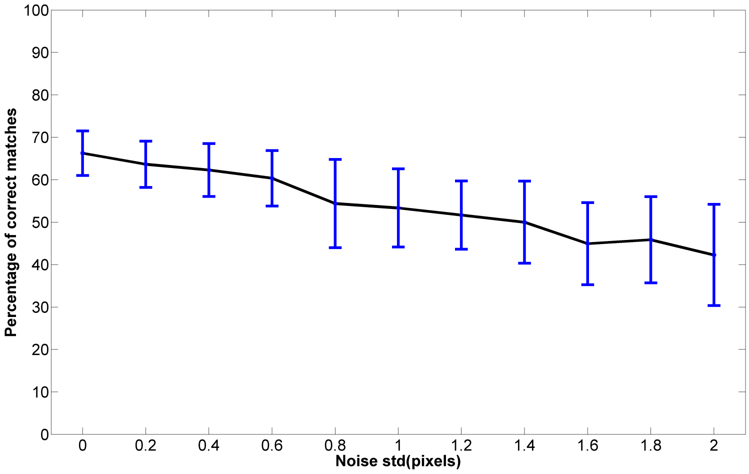


Figure 34: Dependence on noise level. Noise levels are reported in terms of the standard deviation of a zero mean Gaussian.

In Fig. 34, we observe that as expected the number of correct matches decreases as the noise were increases. Note, however, that even with the presence of a Gaussian noise with std of 2 pixels on the endpoints of the extracted segments, more than 45 percent of matches are correct contrary to the complex scene composed of 3 planes where even with noise-free data, less than 40 percent of matches are correct (See Figure 32).

5.3.5 *Simulation 2 (Dependence on the number of lines)*

The simulation parameters are exactly as previous one except the total number of lines to be matched is varied from 15 to 100. The noise is fixed on 0.5x0.5 pixel. Note, in Figure 35, that the algorithm performs better for the lower number of lines. This result is not surprising since in the presence of noise, increasing the number of lines increases the number of one-to-many correspondence and many-to-many correspondence. As the number of extracted segments goes behind 65, on average less than 50 percent of matches are correct. Loosing the second tolerance region to 3 degrees, improves the performance of the algorithm. This proves the importance of the preset tolerances.

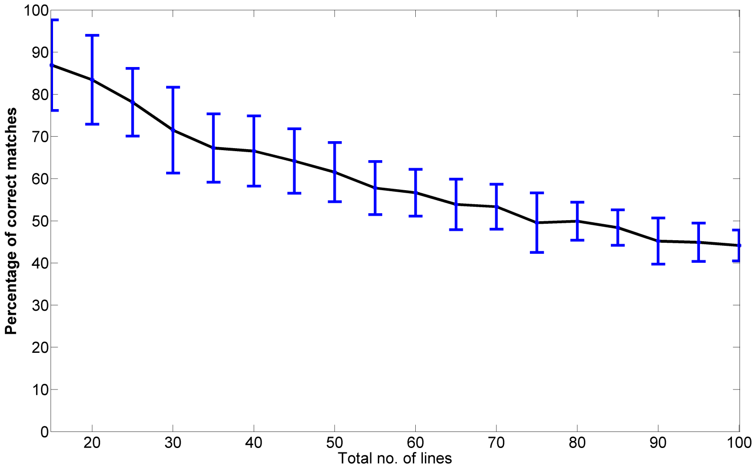


Figure 35: Dependence on the number of lines (0.5x0.5 pixel noise).

5.3.6 *Simulation 3 (Dependence on the percentage of random lines).*

Once again, the simulation parameters are exactly as simulation *1* except the ratio of non-parallel lines to the total number of lines is varied from *0* to *70* percent. The noise is fixed on *0.5x0.5* pixel. Note, in Figure 36, that the algorithm performance decreases as the percentage increases. Once again, this result is not surprising since in the presence of noise, randomly oriented lines may intersect the line at infinity of the plane at the same location as a real match and therefore increasing multiple assignments. However as can be seen from the graph, in the presence of *0.5x0.5* pixel noise and when the ratio is less than *40*%, on average, more than *50*% of matches turn to be correct.

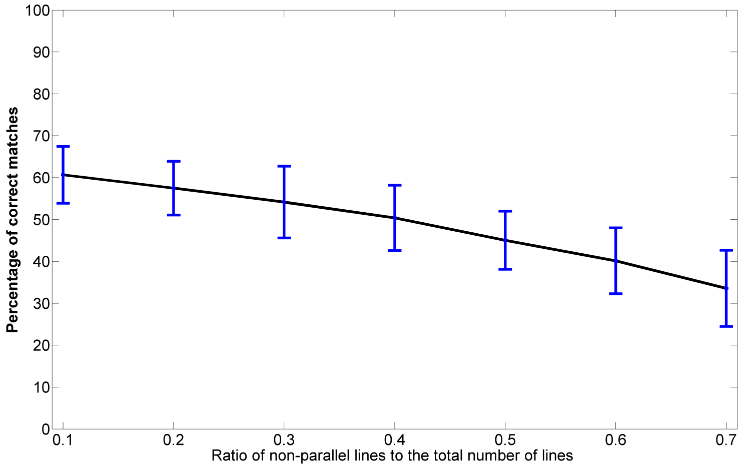


Figure 36: Dependence on the percentage of random lines.(1.0x1.0 pixel noise).

CLARIFICATION: In an early paper published on this work [110], the result of simulation are better than the result presented here. This is because, the algorithm presented in [110] assumes all the lines are parallel to the plane and it does not check for parallelism while in the algorithm presented here, many correct matches are removed during this verification (though some mismatches are also filterered out but for the most of the simulations, the number of filtered-out correct matches is higher). Also, the noise is fixed on *1x1* pixel while for a calibrated camera, setting the noise on *0.5x0.5* pixel should be more realistic. Note that *0.5x0.5* pixel noise means that the endpoints of the segments can be perturbed up to *1.5* pixels in both x and y directions.

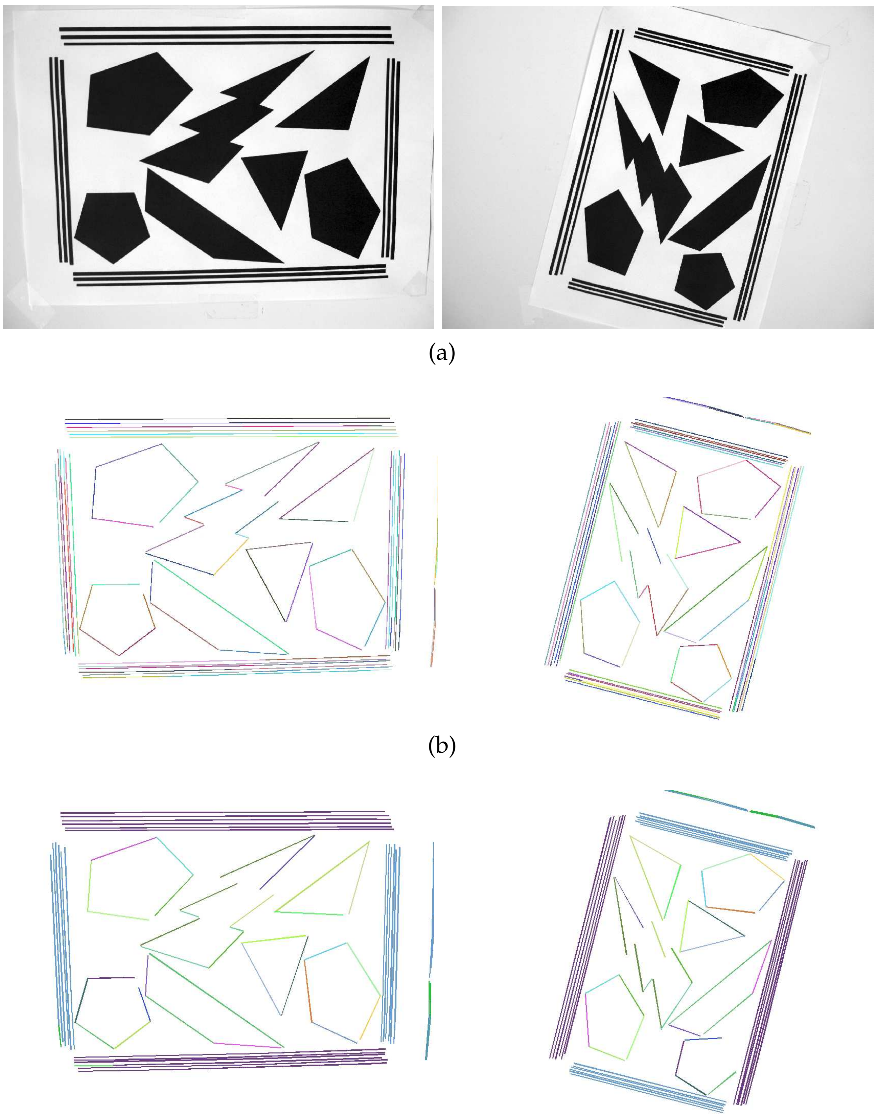
These results give us an insight on the reason for the bad performance of the original algorithm for three planes. All the lines which are not parallel to a scene plane will act as non-parallel lines to the plane under consideration. Assuming each plane has equally number of lines parallel to it, more than *66* percent of all lines (*2**100/3* and also considering the lines which are parallel to none of the planes) are not parallel to each scene plane which means only *35* percent of matches (See the graph in figure 36 where the performance of the algorithm for *66*% of non-parallel lines is around *35*%) based on each plane are correct. This result matches with the result of the graph in Figure 32 for the original algorithm, where for the same amount of noise (*0.5x0.5*) the percentage of correct matches is also around *35*%.

5.3.7 *Experimental results on real central images*

5.3.7.1 *Perspective images*

The first experiment refers to the image pair shown in Figures 37(a) which are the images of a drawing on a wall, imaged from two considerably different viewpoints. The most prominent lines were extracted automatically using Matlab embedded functions for extracting lines using Hough transform and the algorithm proposed in [127] is employed to extract two vanishing points from the images. Total numbers of line segments extracted from left and right images were *92* and *77* respectively. The results of applying the proposed method on these images are shown in 37(c), in which corresponding lines are distinguished with identical colors. Excluding all sets of parallel lines, the algorithm outputs *24* one-to-one correspondences which all are correct. Applying the sidedness constraint on the segments inside each group of parallel lines increases the number of matches to *34* among which *29* are correct. After extracting line segments and two vanishing points, the execution time for the matching stage can be neglected.

The second experiment refers to a pair of aerial images shown in figures 38(a) taken from two very different viewpoints. Note that in aerial images, the majority of the extracted line segments are from the horizontal lines in the scene since vertical lines are either occluded in one or both images or their projections on the image planes are very short segments which are filtered out during the extraction of the segments. The segments extracted from these two images are



(a)

(b)

(c)

Figure 37: (a) Two views of a drawing composed of shapes with straight edges, (b) the extracted line segments, and (c) the computed correspondences. Each color present one match and parallel lines have identical color.

shown in Figures 38(b). Only the most prominent line segments were extracted by filtering out short segments. The numbers of extracted line segments are *144* and *121* line segments from the left and the right images respectively. The output of the proposed method is shown in 38(c), in which corresponding lines are distinguished with an identical color. Majority of the segments belong to five groups of parallel lines and the algorithm outputs *38* one-to-one correspondences among which *32* are correct. Applying the sidedness constraint on the segments inside each group of parallel lines increases the number of matches to *64* among which *51* are correct. The running time for the matching stage is negligible.

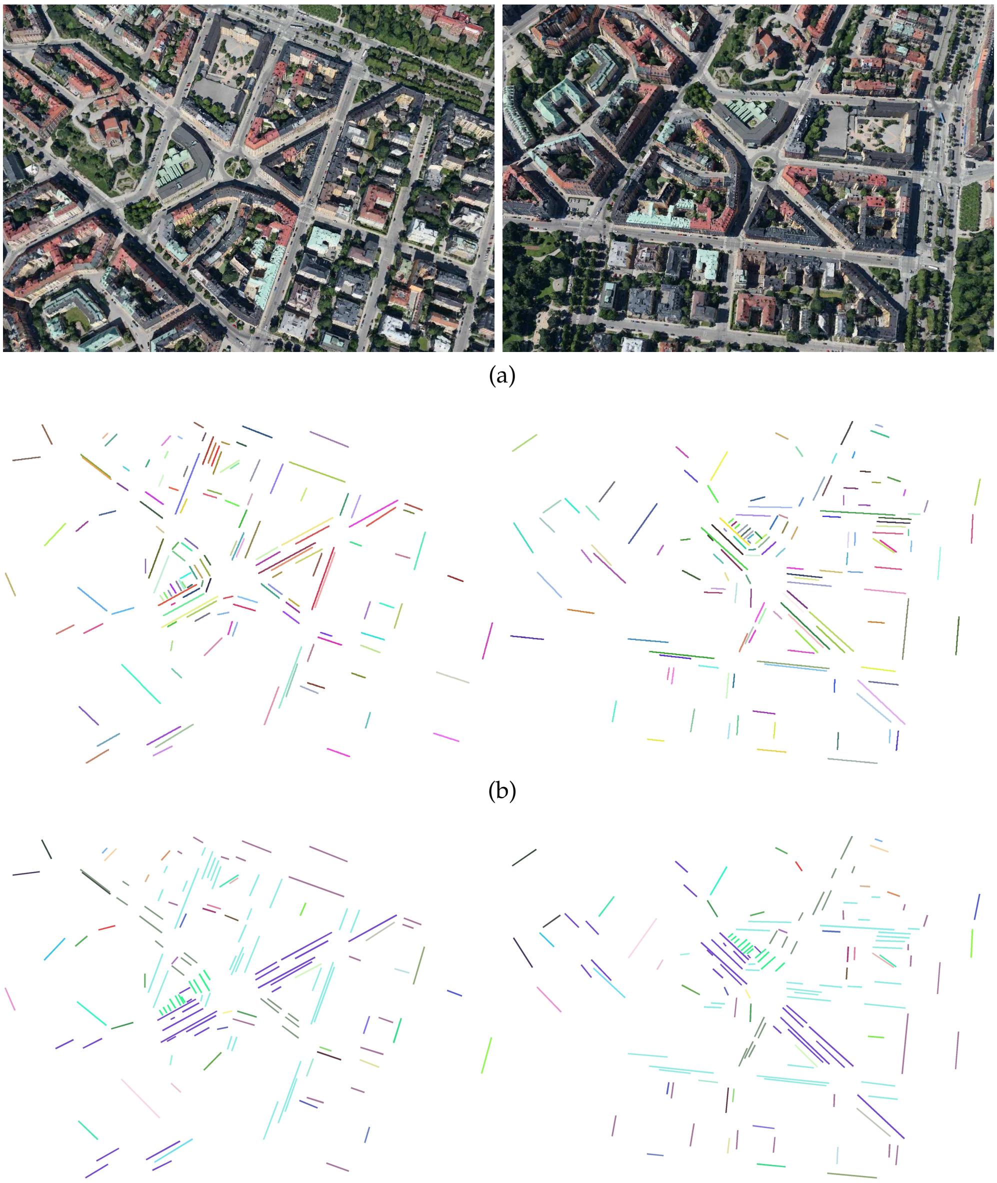
5.3.7.2 *Omnidirectional images*

The only aerial catadioptric image sequence available to us with known camera intrinsic parameters was a set of images captured during a field experiment from a hot-air balloon using the folded catadioptric camera (Figure 12(c)). Two images taken from these set and their extracted conic segments are shown in figure 39. The total numbers of line segments extracted from left and right images were *63* and *68* respectively. The results of applying the proposed method on these images are shown in 40(b), in which corresponding lines are distinguished with identical colors. The algorithm outputs *12* one-to-one correspondences (which are all correct) and *14* many-to-many correspondences. Applying the sidedness constraint on the segments inside each group of parallel lines results in *5* more one to one correct matches.

5.4 CONCLUSION AND OUTLOOK

The problem which we tried to tackle in this chapter was to match two sets of randomly oriented but parallel to a scene plane lines using the location of their intersection with the line at infinity of the plane. Eventually, we proposed a method which has several advantages. It exploits a geometric constraint (the angle between two lines) based on the structure of the scene, without need for the motion of the camera to be known. It is, therefore, also capable of handling disparate views since it employ a constraint which is independent of the motion of the camera. Finally it is computationally very fast and can be run in real-time. Despite all these advantages, the stand-alone algorithm presented here for matching lines between two views of a planar surface is sensitive to the noise and the output of the algorithm is altered as the preset tolerance regions are loosen or tighten.

The geometric constraints presented in this chapter narrow down the whole set of possible matches (*i.e.* each line in one image is a potential match for each and every line in the other image) to a much smaller set including correct matches. Therefore, the subject is still open and one proper direction to improve the result of current algorithm could be to further separate the correct matches from the

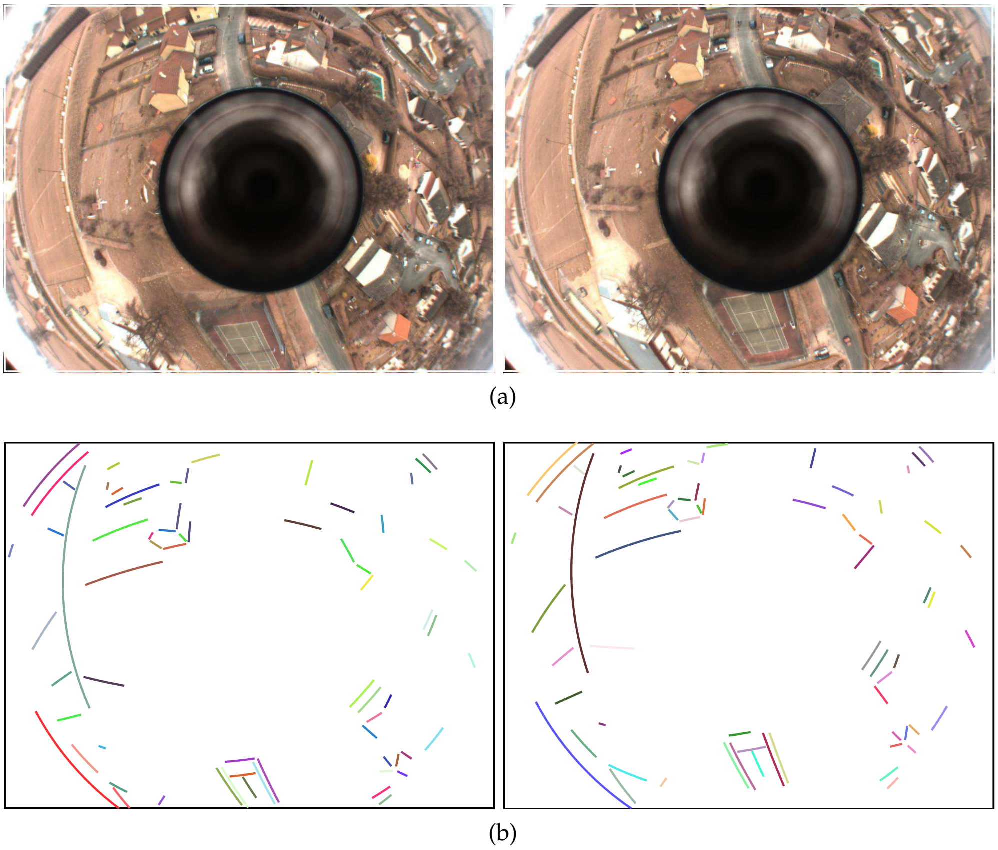


(a)

(b)

(c)

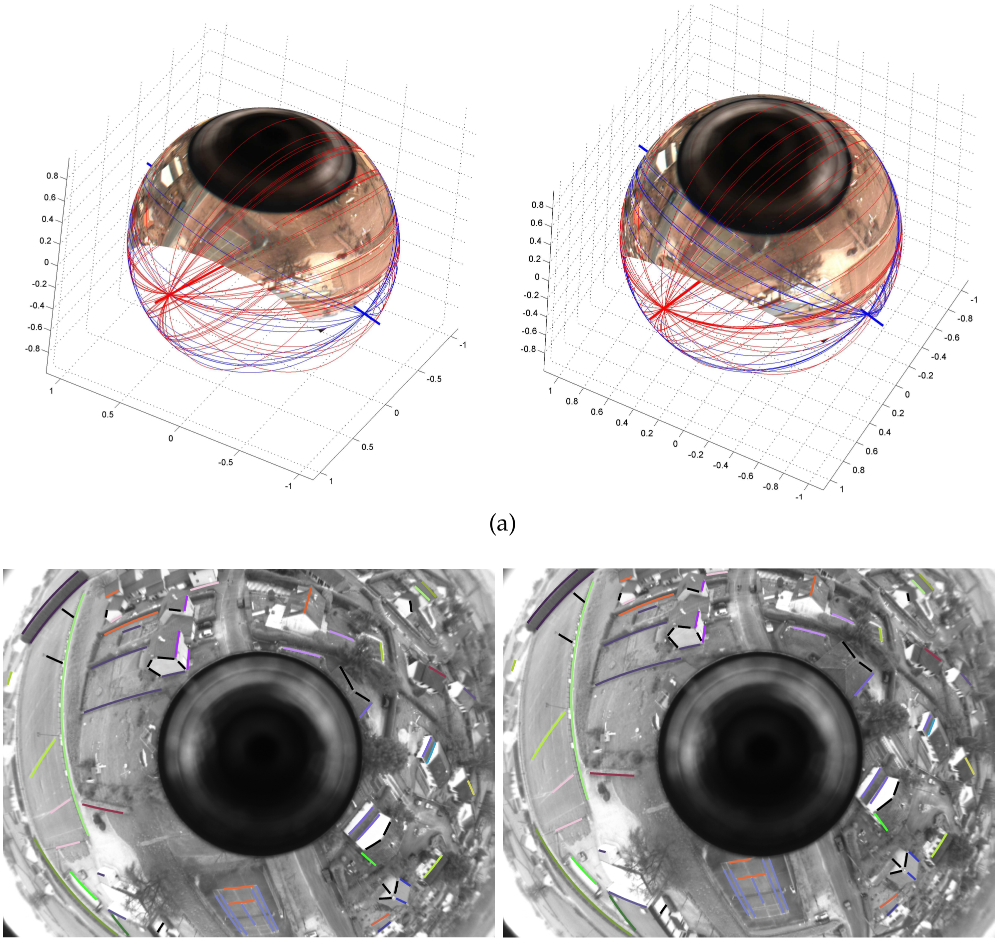
Figure 38: (a) Two aerial images (courtesy of C3 Technologies), (b) the extracted line segments, and (c) the computed correspondences. Each color present one match and parallel lines have identical color.



(a)

(b)

Figure 39: (a) Two aerial catadioptric images and (b) their extracted conic segments.



(a)

(b)

Figure 40: (a) The extracted vanishing directions, and (b) the computed correspondences. Each color present one match and parallel lines have identical color.

rest by considering the spatial and topological configuration of group of lines on unitary spheres or the image planes.

As a future work, an interesting application of the constraint presented in this chapter for perspective images can be to use it as a filter/booster constraint for leveraging the result of any chosen line matching technique currently available rather than using it as a stand-alone algorithm. To do so, consider the output of the selected technique as a set of putative matches. During the filtering stage, whenever vanishing points are available, the ambiguous or false matches can be detected and filtered out by verifying whether their symmetric homographic transfer error is bigger than a chosen threshold. Similarly, during the boosting stage, for any unmatched segment in one image, its possible correspondence can be found by looking for the segment in the other image with transfer error less than a chosen threshold.

Part III

MOTION ESTIMATION AND RECONSTRUCTION FOR CONSTRUCTED SCENES USING TWO VIEWS

In this part, motion estimation and planar surface reconstruction for constructed scenes using two views are investigated and some efficient algorithms are proposed.

Chapter 6: We introduce a unique and efficient way of computing the overlap between two segments which considerably decreases the overall computational time of a segment-based motion estimation and reconstruction algorithm already existing in literature.

Chapter 7: We present an algorithm for reconstruction of piece-wise planar scenes from only two views and based on minimum line correspondences.

6 AN OPTIMIZED LINE SEGMENT BASED STRUCTURE AND MOTION ALGORITHM FOR TWO VIEWS

Generally speaking, motion estimation is the second main stage of SfM process after the feature matching and before the final reconstruction. Wide baseline motion estimation from point correspondences between two views has been the subject of much investigation and even though this is still a very active field of research, many fast, simple and efficient methods have been proposed in such studies. On the contrary, motion estimation from line correspondences between two wide baseline views has not received much attention. The reasons for this are manifold. First, line segments are more difficult to detect and match as was sensed from previous chapter and also due to the reasons mentioned in the section 1.2. Secondly, classical methods such as [129, 9] which use supporting lines (geometric abstraction of straight line segments) need many line correspondences across at least three images. For only two views, during the reviews of works related to the line-based structure from motion, we found several works in which the impossibility of motion determination from the line correspondences between only two views (*i.e.* correspondences between image planes of lines *i.e.* normal vector of the plane passing through the 3D line and the origin of the central imaging system) is geometrically (but not algebraically) shown [141, 9, 27]. As it was shown in the previous section 5.3.1, assuming the rotation part of the motion, R, can be estimated (using for example vanishing points), for each given match $n_l \rightleftharpoons n_{l'}$, the direction of their pre-image in 3D, L can be computed by:

$$L = n_l \times (R^{-1} n_{l'})$$

Having these directions, we were wondering if this knowledge can provide necessary constraints to also estimate the translation part of the motion using just two views. Unfortunately it can easily be shown that knowing the direction of the matched lines do not provide any extra constraints since these line directions are independent of the translation T. In other words, it can be shown that:

$$\frac{n_l \times n_{l'}}{-(n_{l'}.T)} = u$$

Where u is a non-normalized vector in the same direction as L. Note that in this part of the thesis, we will use t and capital T to distinguish between the normalized and un-normalized translation vectors respectively. It can be clearly seen from this equation that T only contributes to the magnitude of the u but not its direction. The simulation shown in Figure 41 also supports this result. The position of the second camera is arbitrarily perturbed. However, the reconstructed 3D line, L' , is still pointing in the same direction as the original 3D line L.

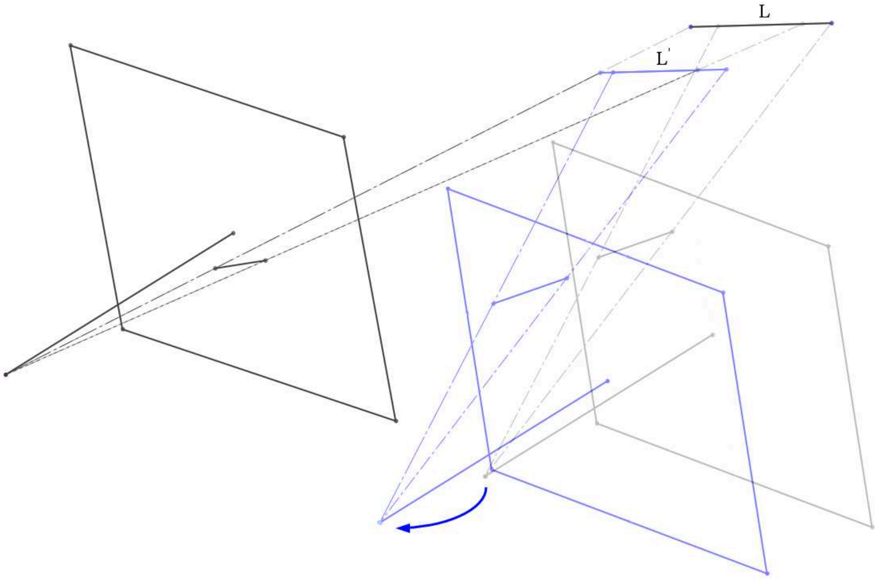


Figure 41: The direction of pre-image of a given match is independent of the relative positions of two views. Gray color represents the true position of the second camera system and blue color represents the perturbed position. There is no rotation between two views.

On conclusion, any arbitrarily perturbed position of the second camera system always yields in a set of reconstructed lines which are always parallel to their corresponding 3D lines, hence, knowing the direction of the lines in 3D does not provide us with extra constraints to solve the translation problem.

Therefore, some assumptions about the nature of the scene and line segments are necessary. To our knowledge, the algorithm introduced by Zhang [153] is, so far, the only work on motion estimation based on only two views of only line segments. The algorithm tries to recover the motion using the epipolar geometry by maximizing the total overlap of line segments in correspondence through benefiting from three main assumptions: considerable overlap between each two matched line segments, relatively large set of line correspondences and finally the variation of the line segments being random. Because a closed-form solution is not available, a five-dimensional motion space (three for rotation and two for the translation) has to be sampled.

In this chapter, after a brief summary of the original algorithm, we try to improve the efficiency of this algorithm by introducing an efficient measure of overlap between two co-linear segments as well as estimating the rotation using vanishing points instead of sampling the rotation space which considerably decreases the overall computational time of the algorithm. For all our data sets in hand, it was also found that the sampling strategy of the original method often is not dense enough to obtain a good initial guess and one needs to sample the motion space with very small steps to obtain an acceptable solution. This observation also motivated us to work on decreasing the time for calculating the objective function in order to be able to search for a good solution over a densely sampled motion space in a shorter time.

6.1 motion estimation by maximizing overlaps

In this section, we present a brief summary of the Zhang's algorithm for solving the motion problem by maximizing the overlap of line segments. The problem to be solved is that given the cameras intrinsic parameters and two sets of line segments, which are in correspondence, estimate the camera extrinsic parameters (motion R and t).

Consider the pair of line segments (l, l') in correspondence as shown in Fig. 42.

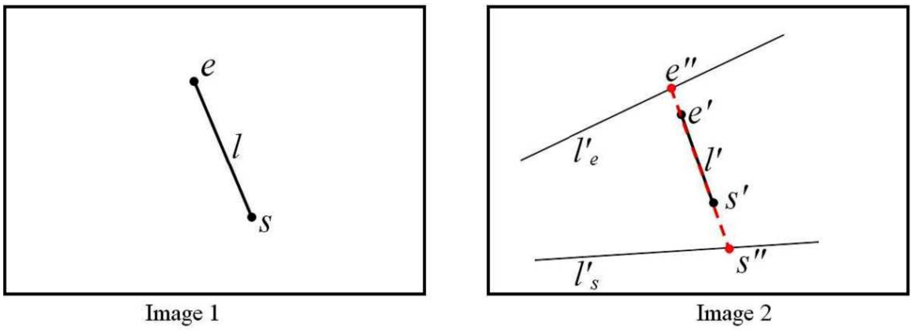


Figure 42: Overlap of two line segments in correspondence.

The line l'_s in the second image is the epipolar line of end point s from the first image, *i.e.* $l'_s = E\tilde{s}$, where $E = [t]_\times R$ is the essential matrix [52]. $\tilde{s}$ is the ray which passes through the end point s and the center of first camera and it can be easily computed since the camera intrinsic parameters are assumed to be known. Similarly the line l'_e is the epipolar line of the other end point e. Taking cross product of each of these two epipolar lines with the segment $e's'$ results in their intersection, s" and e", with the segment. Provided that the epipolar geometry (i.e. matrix E, or the motion $(R\ ,\ t)$) between two images is correct, then s and s" correspond to a single point in space; so do e and e". Thus, the statement that two line segments l and l' share a common part of a 3D line segment is equivalent to saying that line segment s"e" and line segment $s'e'$ (*i.e.*l') overlap. The overlap length, $\mathcal{L}'$, for two line segments in correspondence can then be computed from:

$$\mathcal{L}' = \begin{cases} min(||e' - s'||, ||e" - s'||, ||e' - s"||, ||e" - s"||) \\ \qquad if \left\{ \begin{array}{r} (s" - s')\,\mathbf{.}\,(e' - s") > 0 \\ or\ (s" - s')\,\mathbf{.}\,(e' - s") > 0 \\ or\ (s" - s')\,\mathbf{.}\,(e' - s") > 0 \\ or\ (s" - s')\,\mathbf{.}\,(e' - s") > 0 \end{array} \right\} \\ -min(||e' - s"||, ||e" - s'||) \\ \qquad\qquad\qquad otherwise \end{cases} \tag{6.1}$$

where **.** stands for dot product of two vectors. The overlap length is positive if two line segments overlap, otherwise it is negative. We assume that the orientation information of a line segment is not available (*i.e.* the correspondence between end points of the segments is not known). The overlap length in the first image, denoted by $\mathcal{L}$, can be computed exactly in the same way. Since a small overlap length for a short line segment is as important as a large overlap length for a long line segment, we should use the relative overlap lengths, $\mathcal{L}'/||l'||$ and $\mathcal{L}/||l||$, to measure the overlap of the pair of line segments. The relative overlap length takes a value between *0* and *1* when two segments overlap; otherwise it will be negative. We define relative non-overlap length between two corresponding segments l_i and l'_i in the second image as:

$$\mathcal{H}_i = (1 - \frac{\mathcal{L}'_i}{||l'_i||}) \tag{6.2}$$

which is *0* when two segments completely overlap, between *0* and *1* when they partially overlap and bigger than one when there is a gap between two segments. We can now formulate the motion problem as estimating the camera motion parameters (R ; t) by minimizing the following non-linear objective function:

$$\mathcal{F} = \sum_{i=1}^{n} (\mathcal{H}_i + \mathcal{H}'_i) \tag{6.3}$$

where n is the number of lines. The algorithm can be summarized as the following pseudo-code:

Algorithm 6.1 Zhang's algorithm.

```
Given two perspective images, extract and match their line segments.
Sample the rotation and the translation space with sufficient steps.
for each sample $R(i)$ in the rotation space do
    for each $t(j)$ in the translation space do
        For hypothesized motion $E = [t(j)]_\times R(i)$ calculate objective function $\mathcal{F}_0(i,j) = \sum_{k=1}^{n} (\mathcal{H}_k + \mathcal{H}'_k)$
    end for
end for
for each of 10 best solutions in matrix $\mathcal{F}_0$ do
    Using downhill simplex method, minimize $\mathcal{F}$ (Equ.6.3) starting with the best solution as initial guess.
end for
```

If sampling of motion space is with adequate small steps, at least one of the ten minimization efforts in the last loop converges to a good solution.

6.2 THE NEW MEASURE OF OVERLAP

In the above algorithm, $\mathcal{H}'$ (or $\mathcal{H}$), the function for computing relative non-overlap length for each line correspondence is the most frequently called function and reducing computational time of this function can largely decrease the overall computational time of the algorithm. Consider the two possible configurations of two collinear line segments as shown in Fig. 43. The coordinates of two endpoints of the overlap part, $(Xmin, Ymin)$ and $(Xmax, Ymax)$ can be found by :

$$\begin{cases} X_{min} = max(min(s'_x, e'_x),\ min(s"_x, e"_x)), \\ X_{max} = min(max(s'_x, e'_x),\ max(s"_x, e"_x)), \\ Y_{min} = max(min(s'_y, e'_y),\ min(s"_y, e"_y)), \\ Y_{max} = min(max(s'_y, e'_y),\ max(s"_y, e"_y)), \end{cases}$$

and the overlap length can be expressed by its Cartesian length:

$$\mathcal{L}'_i = \mathcal{L}'_{ix} + \mathcal{L}'_{iy}$$

where

$$\mathcal{L}'_{ix} = (X_{max} - X_{min}), \quad \mathcal{L}'_{iy} = (Y_{max} - Y_{min})$$

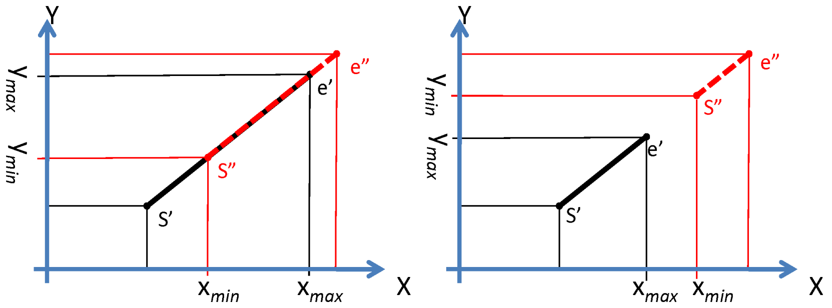


Figure 43: Two possible configurations of two collinear line segments.

Note that the output of this new measure of overlap length is exactly equal to that of Equ. 7.1 but without need for a $if - then$ construct with four OR conditions. While computing the relative non-overlap length, computational time can further be reduced by half by considering only one of the Cartesian components of the overlap part and the segment in the second image (we chose x component) based on the relation shown in Fig. 44:

$$\mathcal{H}_i = (1 - \frac{\mathcal{L}'_{ix}}{||l'_{ix}||}) \tag{6.4}$$

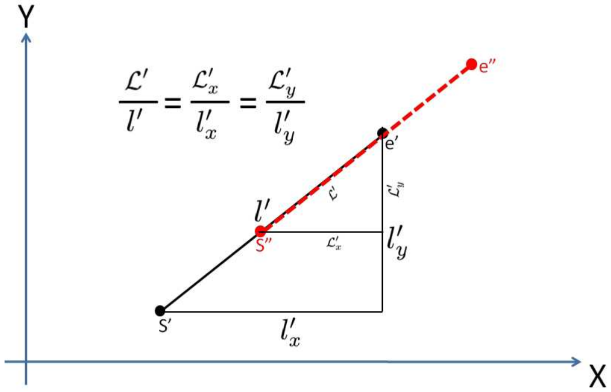

Y
$\frac{\mathcal{L}'}{l'} = \frac{\mathcal{L}'_x}{l'_x} = \frac{\mathcal{L}'_y}{l'_y}$
e''
e'
l'
$\mathcal{L}'$
$\mathcal{L}'_y$
S''
$\mathcal{L}'_x$
l'_y
S'
l'_x
X

Figure 44: The relation between Cartesian components of the overlap part and the segment in the second image. The ratios of corresponding sides of two right triangles are constant.

However, care should be taken when the segment is vertical where y components should be used to avoid undefined division $0/0$. In order to have a very accurate comparison between two measures, we carefully counted the number of CPU cycles needed to run the assembly instructions for the new non-overlap length measure as defined by Equ. 6.4 versus the measure defined by Equ. 7.2 compiled using a C compiler on an Intel Pentium machine. Our new non-overlap length measure needs 302 clock-ticks (including the $conditional\ if$ for proper treating of vertical segments). The original measure of non-overlap needs a minimum of 720 clock-ticks (if the first inequality condition among four inequalities in Equ. 6.1 is satisfied) and a maximum of 858 clock-ticks (if none of four inequalities are satisfied). We can not use clock-ticks average since for the majority of samples in motion space and except for some random lines, the rest of lines do not exhibit an overlap therefore the computation of the objective function for these samples requires maximum number of clock-ticks. This means our new measure can be computed on average slightly less than 858/302 = 2.841 times faster than the measure introduced by Zhang, assuming that the variation of the line segments is random. Refer to the result section for a comparison using real data.

6.3 DENSE SAMPLING OF TWO DIMENSIONAL TRANSLATION SPACE

Sparse sampling of the 5 dimensional motion space (three for the rotation and two for the translation) followed by refinement of the best samples as suggested by Zhang is problem-in-hand dependent and depending on how far the best initial guesses are to the global minimum, the optimization stage can use considerable number of iterations to converge to a good solution or it may not be able to converge at all. Thanks to the possibility of estimating the rotation part of the motion through matching two vanishing points in constructed scenes and also thanks to our faster method for calculating the objective function, we are able to obtain an initial guess for the rotation in advance and we only need to sample the translation space more densely, resulting in a better initial guess closer to the global minimum with less time required by the optimization algorithm to converge to a good solution. The results in the next section demonstrate how this new approach can help to recover the motion for the examples where the sparse sampling followed by a refinement of the best samples cannot converge to a good solution.

The new fast algorithm can be summarized as the following pseudo-code:

Algorithm 6.2 The fast proposed algorithm using vanishing points

```
Given two perspective images, extract and match their line segments.
Match two dominant vanishing directions among the extracted lines and use
them to estimate $R$ (cf. section 4.2.2.1 and Appendix A.3)
Sample translation space with sufficient steps
for each $t(j)$ in the translation space do
    For hypothesized motion $E = [t(j)]_\times R$ calculate objective function
    $\mathcal{F}_0(j) = \sum_{k=1}^{n} (\mathcal{H}_k + \mathcal{H}'_k)$
end for
for each of 10 best solution in matrix $\mathcal{F}_0$ do
    Using downhill simplex method, minimize $\mathcal{F}$ (Equ.6.4 and 6.3) starting
    with the best solution as initial guess.
end for
```

Note that this fast algorithm is applicable only in constructed scenes where at least two vanishing points can be extracted and matched. If this is not the case, then for more accurate results, the original algorithm with the new objective function and a denser sampling of the motion space should be used.

6.4 RESULTS

We have already shown the efficiency of the new objective function in the terms of execution cycles. In this section, however, we give the results on two real data sets where for the last set the original algorithm fails to recover the motion due to sparse sampling of the motion space.

The first set of real data is an image pair of a bakery (Fig. 45). The position and rotation of the second camera with respect to the first one was obtained through a very careful setup and use of a gyroscope:

$R = [-0.0073, -0.3049, -0.0036]$,

$t = [0.9318, -0.0123, 0.3629]$,

where the translation t is normalized and the rotation R is represented by a 3D Rodrigues' vector (whose direction is that of the rotation axis and whose norm is equal to the rotation angle). The segments which are aligned with the epipolar lines are neglected during computing total overlap and later for the scene reconstruction since in this case computed intersections, s and s'' are unstable and the calculated overlap can be irrationally big, resulting in eliminating a good solution.

For speed comparison, we applied the algorithm with both original and new objective functions on this data. We extracted and matched 85 lines between two views manually. Through searching for the initial motion estimation by the sampling strategy as described in the original algorithm (*i.e.* sampling the range $[-\frac{\pi}{4}, \frac{\pi}{4}]$ with steps equal to $\frac{\pi}{8}$ for the rotation and 40 uniform sampling of a Gauss hemisphere based on the icosahedron for the translation), only 1 of 10 best samples converged to the good solution. The result of the best solution is shown in Fig. 45. The whole process (excluding the time for line extraction and matching) took 3159 seconds composed of 1479 seconds for evaluating the objective function over the motion space and 1680 seconds for optimization of 10 best initial guesses (both algorithm were implemented in Matlab and were executed on the same computer). Though replacing the function for computing relative non-overlap with our function does not alter the output of the new algorithm, however it reduces the overall computational time to 1215 seconds (around 2.6 times faster, including all overhead computations). The error in the translation direction is 1.0847°. The error in the rotation angle is 0.3087° and the error in the rotation axis is 1.9147°.

The results are even much better by using the fast algorithm 6.2 where we estimate the rotation between two views using vanishing points and use it as an initial guess. To do so, we used the approach suggested in [14] . Fig. 46 shows projection of two images on unitary sphere. Dominant directions (vertical and

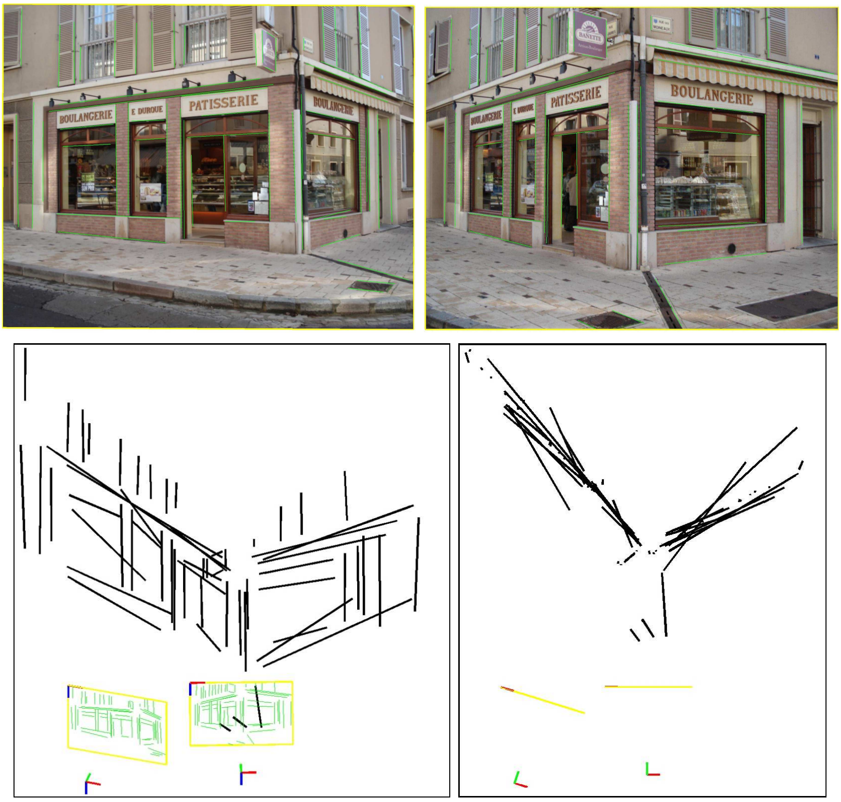


Figure 45: Top row: Two images of a bakery with 85 matched line segments superimposed on the images (in green) . Bottom row: 3D reconstruction of the bakery by Zhang's technique. The right image corresponds to the top view.

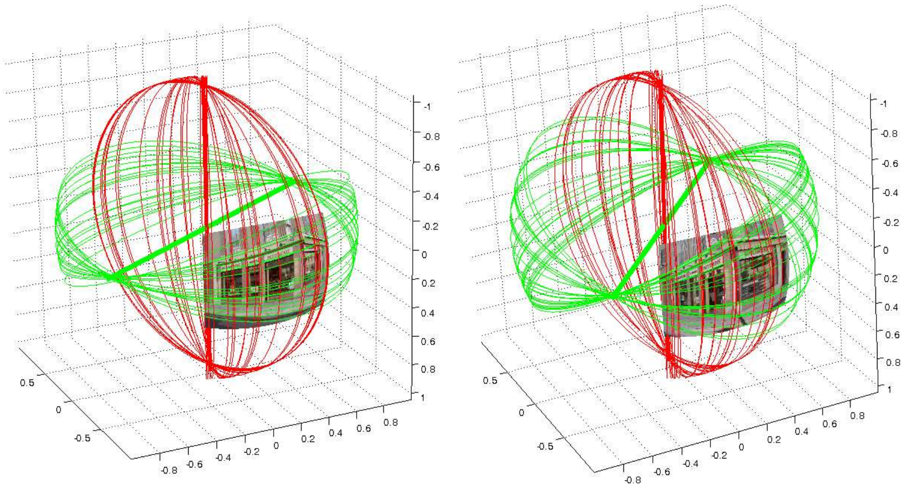


Figure 46: Unitary spheres after projecting the images onto them and extracting vertical and horizontal vanishing points. For a better visualization only a small percentage of lines are shown.

one horizontal) of lines in the scene were used to estimate the rotation of the camera sufficiently accurate. The error in the rotation angle is 1.103° and the error in the rotation axis is 2.074°. For this example, we used 80 uniform sampling of translation space. It turns out that the good solution obtained by the original algorithm is already the global minima and therefore doubling the sampling of the translation space does not alter the results, however the computational time of the whole process took 430 seconds (around 7.3 times faster than the original algorithm).

Fig. 47 shows the second pair of images taken from the real stereo image data set available at INRIA [63]. The transformation from the first camera to the second camera is:

$R = [-0.0004, 0.3133, 0.0717]$,

$t = [-0.9859, -0.0441, 0.1617]$,

We extracted and matched 104 lines between two views manually. Through searching for the initial motion estimation by original sampling, none of the best samples converged to a good solution. The best solution reconstruction corresponding to the initial guess with the smallest value of objective function is shown in Fig. 48 which is apparently a wrong solution.

As a matter of fact, this is an example of a scene where it can be shown that the global minimum is closely located to many local minima and only a fine sampling of the motion space can result in a good solution. Unfortunately, the scene does not consist of at least two accurate vanishing directions. Therefore we applied our dense sampling strategy by 90 sampling of translation space and 1330 sampling of rotation space. The evaluation of the objective function for all these samples takes around 3120 seconds. The best solution's reconstruction is shown in Fig. 49. The error in the translation direction is 1.9°. The errors in

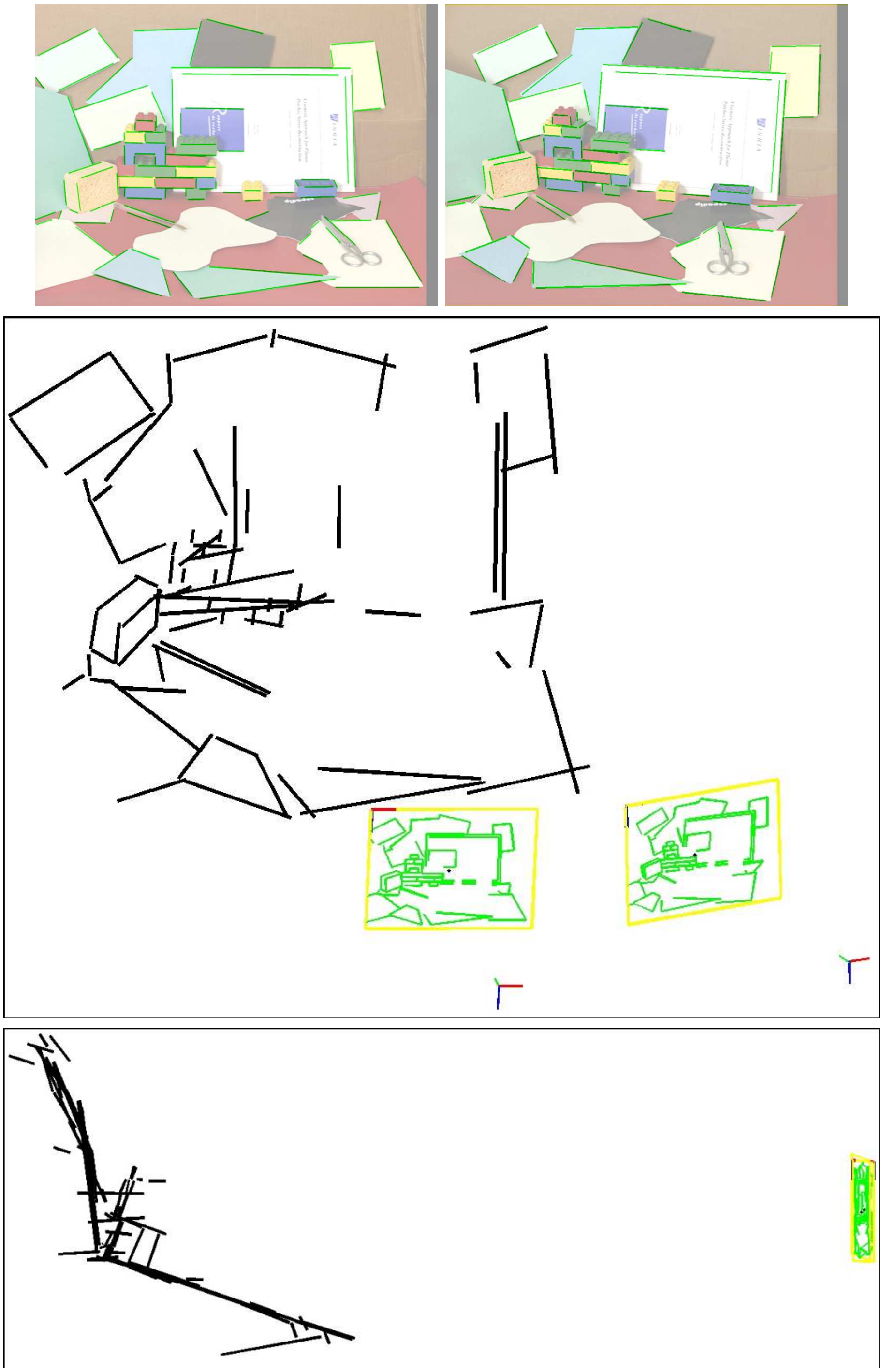


Figure 47: Top row: A stereo pair with 104 matched line segments superimposed on the images (in green) . Middle row: A perspective view of the 3D reconstruction by classical stereo including two camera image planes. Bottom row: a view from the side.

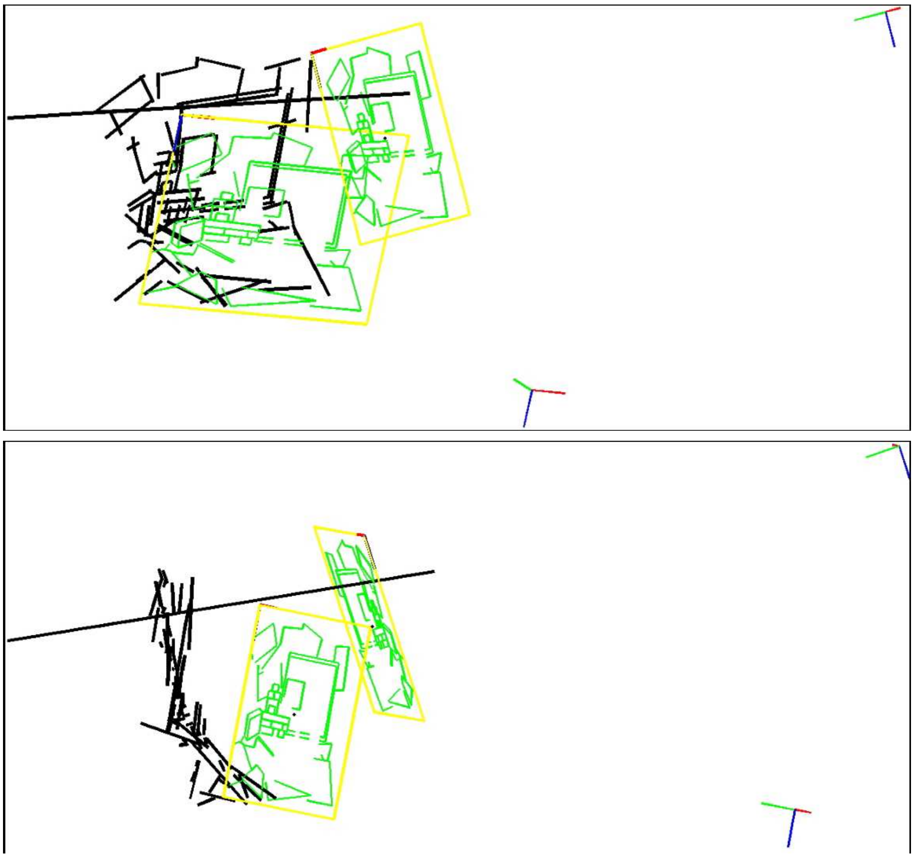


Figure 48: 3D reconstruction of the scene by the best solution of the structure from motion technique described by Zhang. The bottom image corresponds to a side view. Apparently this is not a good solution.

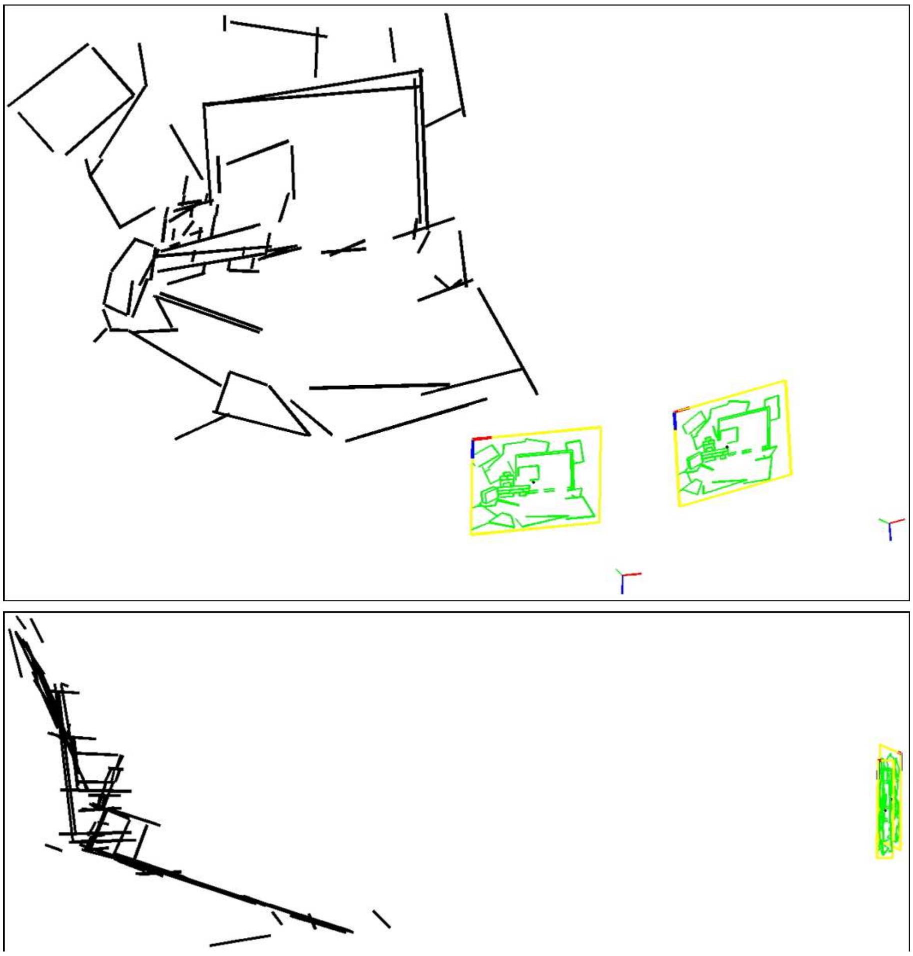


Figure 49: 3D reconstruction of the scene corresponding to the sample with minimum value of objective function from densely partitioned motion space.

the rotation angle and rotation axis are 0.94° and 1.986° respectively. This result is already very good and further optimization is not necessary. One can notice that even though we are benefiting from a faster objective function, however the evaluation of all samples in the dense motion space is quite time consuming and except inevitable evaluation of all these samples, there is not any other deterministic alternative approach for such particular examples.

6.5 conclusion and outlook

We introduced a new measure of overlap which increases the speed of calculating the overlap between two line segments in correspondence. It also allows a denser sampling of the motion space for finding initial guesses for optimization of the non-linear objective function for recovering motion based on line segment correspondences and therefore facilitating the search for a good solution where due to the nature of the scene, sparse sampling followed by optimization does not converge to a good solution. We also suggested, whenever possible, to estimate the rotation between two views using vanishing points and use it as an initial guess in order to reduce the sampling space to two dimensions. We demonstrated this situation by giving the results on two real data sets including the scene where the original algorithm fails to recover the motion.

7 MOTION AND STRUCTURE FROM TWO CENTRAL VIEWS OF PIECEWISE PLANAR SCENES

The optimized method presented in the previous chapter for motion estimation from line correspondences between two wide baseline views has this main drawback that it needs relatively large set of correspondences of line segments randomly distributed and oriented in the scene. Though the number of extracted line segments from an image of a constructed scene can be relatively high, matching them is a very difficult task especially if the motion has a long baseline. Moreover, in constructed scenes, the assumption of randomly oriented lines may not hold since the majority of the lines are pointing in the same direction as one of three main Manhattan directions.

While searching for new methods of line matching in constructed scenes, we came to realize a simple method of simultaneous motion estimation and reconstruction of planar surface using minimum requirement of matched lines which is especially powerful in dense reconstruction of planar surfaces. Architectural interiors are often very poorly textured and as a result, the number of extracted interest points is too small for dense 3D reconstruction. The problem is even more challenging if the images are of low resolution or bear the omnidirectional deformation. Therefore, in order to create a dense 3D model of a poorly textured scene, we suggest an approach which uses as much as possible photometric information available and some prior knowledge about the planar nature of the scene. Despite the fact that two views of lines are not enough to estimate motion, we present a novel method that still uses two views of just one line and all pixel information around it and simultaneously estimates the translation and reconstructs the surface.

7.1 THE METHODOLOGY

Using both geometric and photometric constraints of the line and its image neighborhood and reconstructing the 3D surface around the line is the key idea of our method. For each view, a set of planar facets passing through the 3D line in space are hypothesized and the plane hypothesis which shows higher similarity between two views is chosen to be the best reconstruction of the surface (see figure 50). The camera translation is simultaneously recovered during the construction of the surface.

For the rest of this text, we assume that the first camera coordinate system is aligned with the world coordinate system with no loss of generality. The

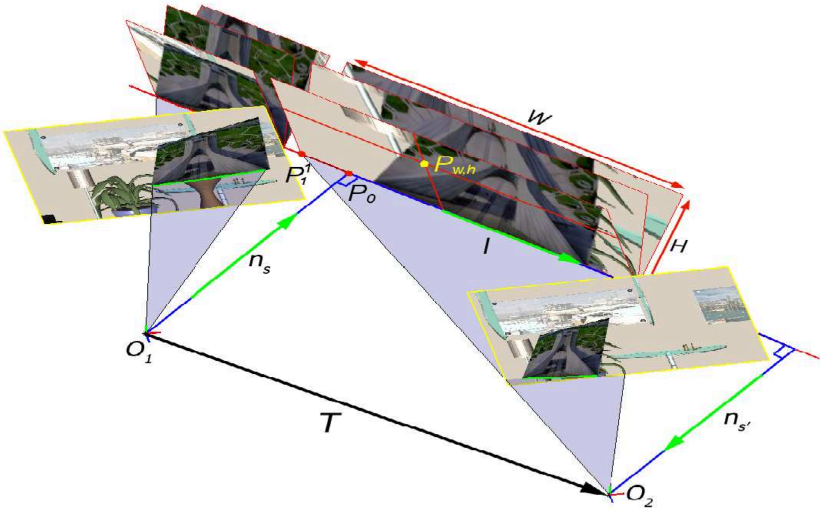


Figure 50: Geometry of proposed method. On each hypothesized plane (three planes are shown here), one rectangular mesh is built on the same side of the 3D line from each camera image.

intersection of two planes passing through each segment and the origin of the related camera results in l, the direction of the 3D line:

$$l = \frac{n \times R^{-1} n'}{\|n \times R^{-1} n'\|} \tag{7.1}$$

where n and n' are known images of the 3D line on the image planes (*i.e.* normal vector of the plane passing through the 3D line and the origin of the camera) and the camera rotation, R, is known *a priori*.

The full reconstruction of the line, however, is not possible since at least the ratio of the distances of the line to the origins of the cameras should be known. T, the translation from first camera to the second camera (recall that we are using t and capital T to distinguish between the normalized and unnormalized translation vectors respectively) can be decoupled in three vectors and be expressed by (see Fig. 50):

$$T = n_s s + l s_l - n_{s'} s' \, , n_s = \frac{l \times n}{\|l \times n\|} \, , n_{s'} = \frac{l \times R^{-1} n'}{\|l \times R^{-1} n'\|} \tag{7.2}$$

where s and s' are the unknown distances between 3D line and the origins of the cameras and s_l is an unknown scalar value. Therefore in order to determine the translation of the system we need to find these three values. Since the final solution will always be up to a positive scale, we can set one of these values to a fixed value (we chose s). Now the problem is simplified to estimating the other two scalars. To estimate these two values we need to use the photometric information around the line. However any comparison between the areas around the images of the line in two views is meaningless due to distortion caused by the motion between two views. To overcome this difficulty we reconstruct the surface and therefore the texture on it from each image by sweeping a hypothesized plane through space which pass through 3D line and find the plane which best matches two reconstructed surfaces. In the next section we derive the necessary formulas for the parameterized reconstruction of the 3D surface and its texture seen by each camera, referred to as the "mesh image" hereinafter.

7.1.1 *Building 3D rectangular meshes and mesh images*

For each image, the steps of building a 3D rectangular mesh with the orientation n_m on one side of the 3D line are as follows. Note that, for the sake of clarity in the schematic figures, we draw a perspective image plane instead of more generic image sphere.

Let p_i^j denote the back projection ray (a point on the image sphere) of i^{th} end point of the line segment of the j^{th}camera and P_i^j denote its corresponding Cartesian coordinates on the 3D line with respect to the world coordinate system (see Fig.50). P_i^j can be found by intersecting p_i^j with the plane which passes through the 3D line and has the orientation n_m. After some calculations, this intersection can be expressed by (here T, as a superscript, means transpose):

$$P_i^j = O_j + \frac{(n_m \cdot (P_0 - O_j))}{\left(n_m \cdot \left(R_j p_i^j - O_j\right)\right)} \left(R_j p_i^j - O_j\right) \ , \begin{array}{c} i = 1,2 \\ j = 1,2 \end{array} \tag{7.3}$$

where O_j and R_j are the focal points and camera rotations with respect to the world coordinate system, respectively. Since the first camera coordinate system is aligned with the world coordinate system, we have $O_1 = [0\,0\,0]^T$, $R_1 = I$ (*Identity matrix*) , $R_2 = R$ and $O_2 = T$ (since O_2 is now the translation vector between two camera positions). The 3D point P_0 is a point on the 3D line, and it can be chosen to be the closest point of the line to the origin of the first camera: $P_0 = n_s s$. For each image, $P_{w,h}^j$, the 3D coordinates of the mesh at location (w, h) is:

$$P_{w,h}^j = P_1^j + wl + h\frac{l \times n_m}{\|l \times n_m\|} \ , \ w = 0 : W^j \ , \ h = 0 : H^j \tag{7.4}$$

The mesh resolution should be carefully selected since it depends on the scale of whole reconstruction which in return is determined by the value chosen for s. W^j should be chosen proportional to the distance between P_1^j and P_2^j in order to simplify the registering of two meshes at later steps of the method:

$$W^j = round(k \left\| P_1^j - P_2^j \right\|) \tag{7.5}$$

where k is a positive value. The higher the k is, the higher the resolution of the mesh is and therefore the reconstruction of the surface is more accurate at the expense of higher computational time.

H^j, the height of the mesh must be high enough to include neighboring texture around the segment, otherwise during the registration, the similarity measuring function can fail to match mesh images.

After some calculations, $p^j_{w,h}$, the corresponding image pixel for each mesh vertex $P^j_{w,h}$ can be expressed by:

$$p^j_{w,h} = \left[\Delta_x + u_j \; \Delta_y + v_j\right]^T, \begin{matrix} w = 0 : W^j \\ h = 0 : H^j \end{matrix}, j = 1, 2 \tag{7.6}$$

$$\Delta = R^{-1} \left(\frac{f_j}{r_3^T \left(P^j_{w,h} - O_j \right)} \left(P^j_{w,h} - O_j \right) \right) \tag{7.7}$$

where r_3 is the third column vector of rotation matrix R. Fig.51 shows examples of what mesh images look like as the parameters change. Only for ground truth parameters, the surface textures in both mesh images are identical. Note that one of the strips is extended up to the image border, therefore, the algorithm works as long as a part of the surface attached to the line is seen by both cameras and it is not necessary that the segments overlap.

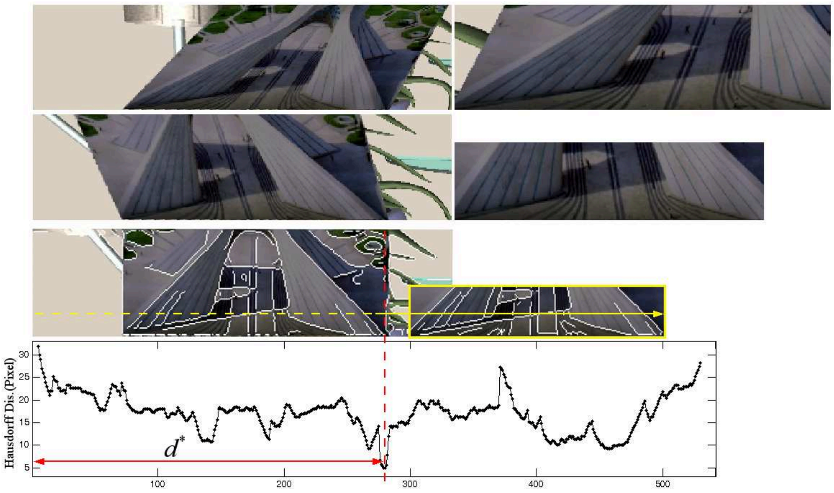


Figure 51: Two top rows: 2D mesh images where orientation of the surface and depth ratio are different from the ground truth. Two bottom rows: As the second mesh image sweeps the first mesh image, the Hausdorff distance between points of overlapping parts is computed (point sets are shown by white pixels). The best estimation of the surface orientation and s' occurs to have the lowest Hausdorff distance. The location where this smallest value occurs, d^*, is used to register two meshes in 3D and therefore recovering s_l.

7.1.2 *Estimation of surface orientation, depth ratio and T*

After setting s to an arbitrary value and choosing the mesh resolution accordingly, the best values of scalars s' and s_l and also the orientation of the surface which minimizes the Hausdorff Distance between two point sets (extracted from mesh images using edge detectors such as Canny) must then be estimated in order to recover T (up to a scale factor) from the equation 7.2. Refer to the section 7.3 for a justification on using Hausdorff Distance as the similarity measuring function. This needs a brute force algorithm for searching among all possible values for these variables which is computationally expensive. Fortunately, the problem can be formulated in a simpler way and computational burden can be greatly reduced by observing the following facts from the geometry of the proposed method:

- s_l can be set to zero since its value has no effect on retrieving the mesh images (*i.e.* image pairs of Fig. 51 do not change as value s_l changes). However its real value is necessary for computing T and it will be recovered through registering two meshes in 3D (Eq. 7.8).

- Changing depth values s and s' has a zooming effect on their mesh images (*i.e.* doubling the distance between camera origin and the 3D line results in a twice larger mesh image). This suggests that in order to reduce computational time, instead of reconstructing from scratch, the second mesh image can be built just once by setting s' to its maximum expected value and then the effect of reducing s' can be simulated by resizing the second mesh image using interpolation. We limit the possible depth ratios $\frac{s}{s'}$ to the range $[\frac{1}{3}\ 3]$ in order to restrict the search and chose 20 equi-spaced points on this interval. In practice, this ratio is not likely to exceed this range.

- The searching space for the surface ground truth orientation, n_m, can also be reduced by taking into account that not all orientations of the hypothesized plane can generate proper image on the same side of the line segment. For details see Fig.52.

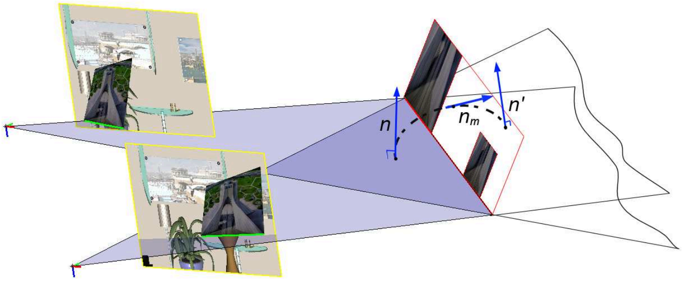


Figure 52: The searching space for the surface ground truth orientation. Instead of searching all potential surface orientations, a smaller set of likely surface normals varying between n and n' should be considered.

The pseudo-code algorithm 7.1 summarizes our method for simultaneous estimation of surface orientation and depth ratio. The combinatorial functions $index(min(M))$ return the index corresponding to the smallest value in the matrix M.

Algorithm 7.1 The proposed motion and structure algorithm.

```
Given two images taken by a central imaging system, extract their line
  segments and match a line segment between two views belong to the surface
  under consideration for reconstruction.
Match two dominant vanishing directions among the extracted lines and use
  them to estimate $R$ (cf. section 4.2.2.1 and Appendix A.3)
$s \leftarrow$ An arbitrary positive value
for each $n_m(i)$ between $n$ and $n'$ do
    for each $\frac{s}{s'}(j)$ between $\frac{1}{3}$ and 3 do
        - Construct two mesh images and extract edge points.
        - $M(i,j) = $ MIN(Hausd. Dis. between two point sets).
    end for
end for
$(i^*, j^*) \leftarrow$ index(min($M$));
return $n_m(i^*)$ and $\frac{s}{s'}(j^*)$;
```

The surface orientation n_m and the depth ratios $\frac{s}{s'}$ corresponding to the smallest distance in the matrix M inside nested loop are the best estimations of these two parameters. Note that the nested loop is very fast since not only the template and image to be searched are small, but also the whole image is not searched. Since it is not clear that which side of the line is planar, we run the algorithm for both sides and chose the side which gives a better reconstruction (*i.e.* lowest Hausdorff distance). If the Hausdorff distances for both sides are small enough, then there is a high chance that either the line is the intersection of two planar surfaces or it is located inside the surface instead of its boundaries (in the later case two computed orientations match).

After estimating n_m and s', computing s_l is straightforward. Let denote $d^\star$ the distance at which the Hausdorff distance between two mesh images corresponding to the best estimation of s' and the surface orientation occurs to be minimum as shown in Fig. 51. After some simple geometric considerations, s_l can be expressed by:

$$s_l = P_0 - P_1^1 + d^\star l + \left(P_1^2 - P_0 - W^2 l\right)\left(s/s'\right) \tag{7.8}$$

Knowing all the three scalars, T can be computed by equation 7.2.

7.2 MORE THAN ONE LINE CORRESPONDENCES

Theoretically, one line correspondence on a textured surface is enough to estimate the translation. While it is noteworthy that the approach works from only a single correspondence, in practice one can usually determine multiple good correspondences and combining the estimates from all available correspondences can help to verify and refine the accuracy of the estimated translation as well as reconstructing more planar surfaces of the scene. Assume T_i and T_j are two such estimations and $\frac{s_i}{s'_i}$ and $\frac{s_j}{s'_j}$ are corresponding depth ratios. It is clear that:

$$\frac{T_i}{\|T_i\|} = \frac{T_j}{\|T_j\|}, \; \frac{\|T_i\|}{\|T_j\|} = \frac{s_i}{s_j} = \frac{s'_i}{s'_j} \tag{7.9}$$

The first constraint implies that the direction of the translation is identical for each line correspondence. Even though translation vectors computed by each line correspondence have different magnitude (due to different scales), they should have the same direction. This constraint can provide a method for verifying the other estimated translation directions. One can also improve the accuracy of the final translation vector by computing the vector which has the minimum deviation from all the estimations. The second constraint relates the overall scale between two line (surface) reconstructions and can be used to reconstruct all planar surfaces of the scene in one uniform framework (refer to the Fig.55 for an example of application).

7.3 DISCUSSION

The bottleneck of our proposed method is finding the location of one of the mesh images in the other one. This is simply well-known template matching problem for which there are numerous number of methods available in literature with different cons and pros. These methods are usually different in the level of invariability to various deformations such as translation, rotation, scale, affine or perspective. Due to the nature of our problem in this stage, we are looking for the most simplest similarity measuring function which should not be invariant to any deformation except translation (to escape any false positive matches since we want the function to show a high similarity only when the surface reconstruction is corresponding to the ground truth). One choice is ZNCC which is very simple but as it is shown in [84], it increases the sensitivity of the algorithm to the error in the estimation of rotation since this function is very sensitive to the pixel displacement which is inevitable during forming mesh images. Therefore we employed a generalized Hausdorff Distance which has been shown to work well in comparing images and it is computationally efficient when the template undergoes a simple translation [62]. It also can deal with individual pixel displacement errors while remaining sensitive to overall mesh images deformation.

The hausdorff Distance simply provides a means of determining the resemblance of one point set to another, by examining the fraction of points in one set that lie near points in the other set. For more details on this similarity measuring function, refer to Appendix B.

It is worth mentioning that, in order to compare the surrounding of the line segments in two views, we also approached the problem in a similar way to what plane-sweeping methods do for feature matching and depth recovery using homography induced by each plane hypothesis [36] and we ended up estimating 3 unknowns (instead of 2 for our proposed method) plus a higher computational cost. One also may notice that our work is very different from plane-sweeping approaches which use the already known motions between several views of a scene for 3D reconstruction while our method uses lines first to recover the motion from only two views and at the same time to reconstruct the surfaces of the scene.

We consider our approach as a simple and efficient method for dense reconstruction of a planar object or scene from two images taken with considerably long baseline motion and with minimum need for the user interaction compared to other methods. Using methods such as Zhang's method [153] or multiple-view methods, we are only able to reconstruct 3D line segments and not any surfaces, therefore our method has the advantage of dense reconstruction. Also note that since we reconstruct a line and part of the surface attached to it, any other line coplanar with the surface is also reconstructed and we do not need to match and reconstruct these lines anymore (for example each wall of the bakery and the pavement attached to it (Fig. 55) are reconstructed using only one line from their intersection and we do not need to deal with the rest of the lines on these two surfaces). On the contrary, these coplanar lines and also parallel lines can decrease the performance of the methods which rely on only geometric information.

7.3.1 *Sensitivity to the error in estimating R*

While reconstructing two meshes, the estimated rotation between two views plays a critical role. Generally speaking, the error in the estimation of the rotation can have a considerable effect on the estimation of the translation and this is always the case when estimating the displacement by a decoupling approach. The line extraction stage can also introduce some error in the direction and location of the segments on the image plane. In order to find the extend of the sensitivity of the recovered translation to these errors, we have carried out 10 sets of *100* experiments with simulated image pair of Fig.50 and instead of identity matrix, we set R to a rotation matrix with an arbitrary rotation axis and a rotation angle around this axis which increases between each set by *0.2* degrees (*i.e.* in the 10th set of experiments, there is a 2 degrees rotation between two views). The results of the experiments are shown in Fig.53.

In general, the introduced error increases as the angle of rotation increases but this is not true for all experiments. This reflect the fact that the angles close to *90* degree between axis of rotation and the 3D line may cause greater error in the estimation of the translation than the increase in the angle of rotation. This is true because in this case, the highest deformation on the mesh images occurs which can easily affect the performance of the similarity measuring function in correctly registering the two mesh images. Therefore a better similarity measuring function is advised if the error in the estimation of the rotation is expected to be high.

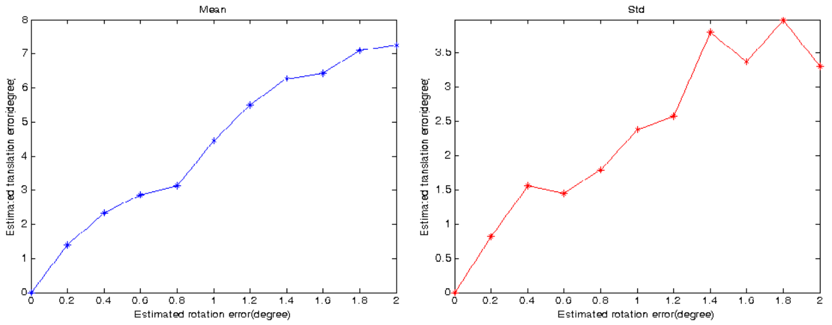


Figure 53: The mean/std error for the estimation of the translation direction.

7.4 an interactive 3d reconstruction interface

Two important steps of algorithm 7.1 are estimation of rotation and matching line segments on the surface/es under reconstruction. While the former step can be automatic using vanishing point extraction and matching methods, the latter step is still a challenging task especially for omnidirectional images. Therefore we suggest an interactive interface where, after automatic estimation of vanishing points and recovering the rotation, as the user selects two corresponding line segments between two views, the algorithm reconstructs the surface and estimate the related translation. The interaction ends when all planar surfaces of the scene are reconstructed. For more details on this interactive algorithm refer to Appendix B.

7.5 experimental results

Here we present some result and comparison with the ground-truth (whenever available) which were compiled during the development of the method and the algorithm. Note that since, for the real perspective examples, we did not benefit from above interactive reconstruction interface to locate the corresponding lines, the results are less accurate due to the introduced localization and orientation errors during automatic detection of the lines.

7.5.1 *Perspective*

7.5.1.1 *Synthetic images*

To prove the validity of the method, we tried our algorithm on simulated images such as images of Fig.50 in which there is no rotation between two views. Note, however, that there are still a few sources of error such as the error in the position and direction of extracted line segments. Despite these errors, the translation was estimated almost without error (the angle between estimated direction of the translation and ground truth was 0.08 degrees).

7.5.1.2 *Real images*

The first pair of real images are the same set presented in the previous chapter in Figure 45 taken from a bakery which is mainly a piecewise planar structure. Using Zhang's algorithm, the motion is well estimated but the reconstruction is not very accurate. This may reflect the fact that the computed overlap between segments is not correct when there are segments that are almost parallel to their epipolar lines. This is the case for many horizontal segments of this example.

For comparison, we applied also our algorithm on this data. We use all lines which can automatically be extracted from each image (without need for matching them) to recover the rotation and only one line correspondence to recover the translation and two more line correspondences to verify and refine the results.

For estimating the translation, we chose 30 equi-spaced points on the searching interval for the surface orientation. The output of our algorithm for reconstruction of 3 scene surfaces and 3 estimated translation vectors associated to each surfaces is shown in Fig. 54.

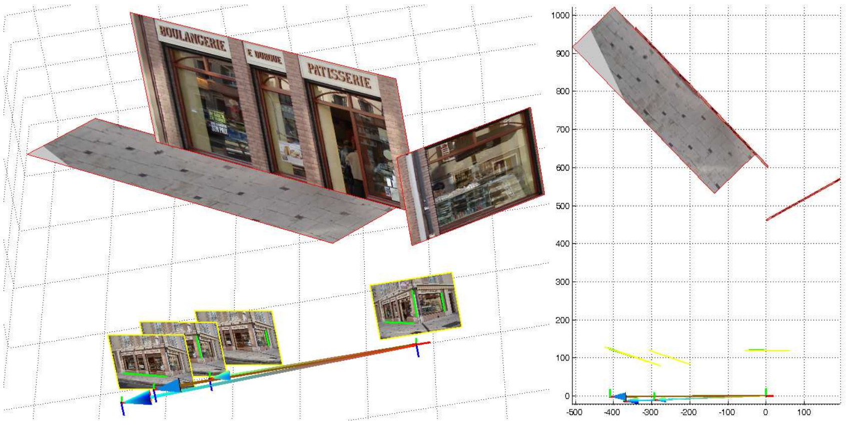


Figure 54: Reconstruction of the pavement and two walls of the bakery and 3 estimated translation vectors related to each surfaces. The scale of each surface reconstruction is different from others. The right image corresponds to the top view.

All three surfaces plotted in a unique frame are shown in Fig. 55. The worst estimated translation error among three is 4.7°. The estimated motion is comparable with the Zhang's algorithm and as it can be seen from Fig.54, the immediate result of the algorithm is a more useful reconstruction of the scene. In order to test the accuracy of the reconstruction, we took a few concrete distance and angle measurements (shown with blue arrows) and compared them with the result of the reconstruction. The worst estimated angle error between planes is less than 3° and the difference in the distance between landmarks is not exceeding 2.4%.

Fig.56 shows the second pair of real images taken from a street sidewalk. The green segments are the edges attached to three chosen planar surfaces. The final

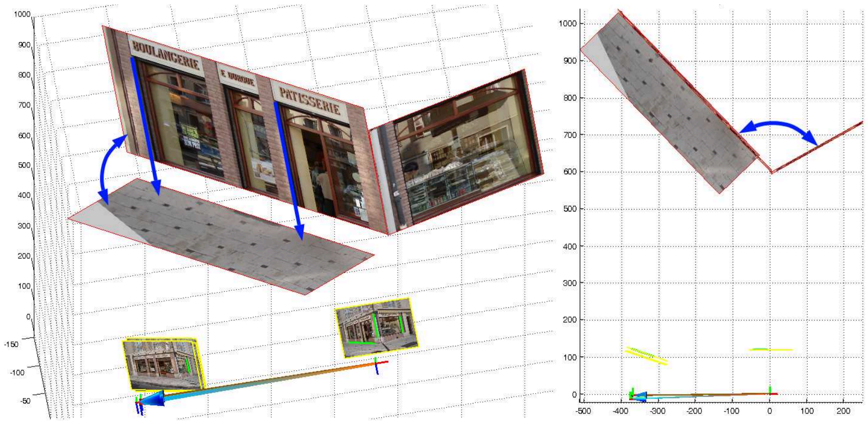


Figure 55: Reconstruction of the bakery and 3 estimated translation vectors in a unique frame.

output of the algorithm is shown in Fig.57. Though we did not record the real motion of the camera, the result of the reconstruction is acceptable and therefore the estimated motion is accurate enough for such method.

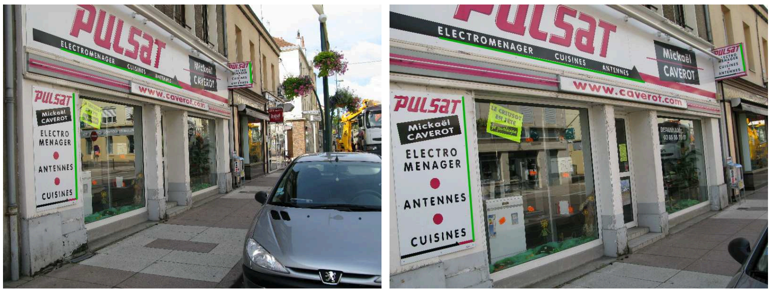


Figure 56: Image pair of a street sidewalk scene.

Careful readers may notice that the accuracy of motion estimation and reconstruction of the proposed algorithm should not be compared with multiple-view-based methods such as [129, 9, 3] which are generally more accurate but need extraction and matching of many line (point) correspondences between more than two views, a very difficult task especially if the motion has a long baseline. Besides, they are not applicable on omnidirectional images. For piece-wise planar scenes, our algorithm outperforms such generic methods in the sense of speed and accuracy of results are also comparable.

7.5.2 *Omnidirectional*

Figure 58 shows two images from an interior scene which is also mainly a piecewise planar structure taken by our classic paracatadioptric system (Figure 12(d)). For this example, we also use all lines which can automatically be extracted from each image to recover the rotation (58(b)) and only one line correspondence to recover the translation and 3 more line correspondences to verify and refine

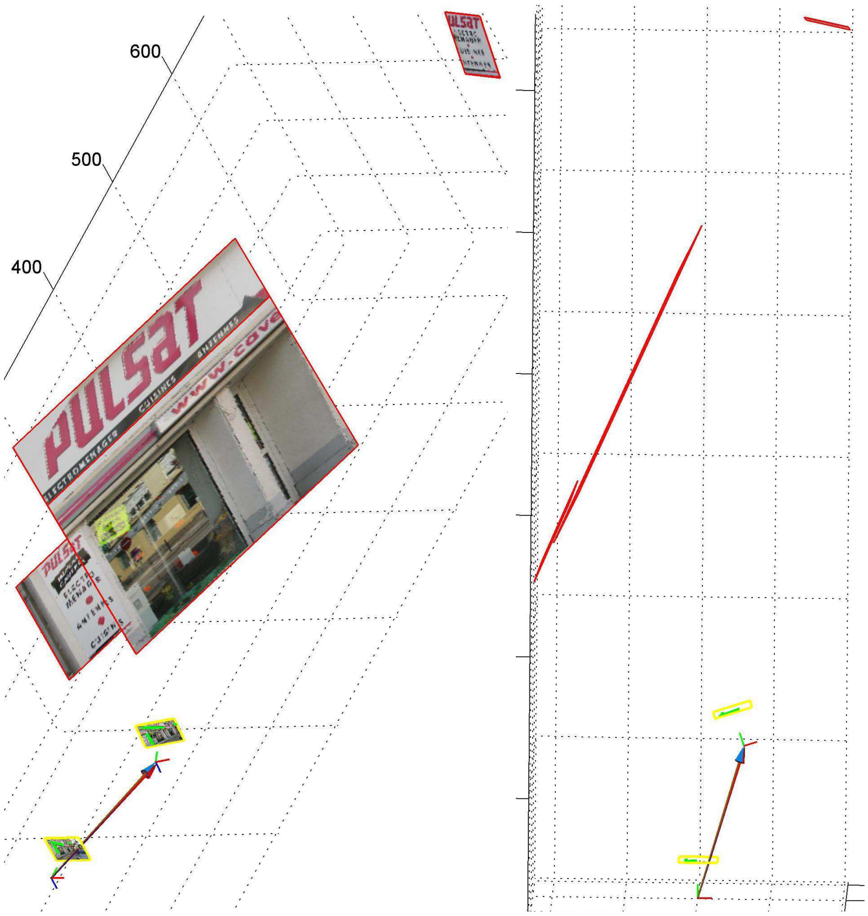
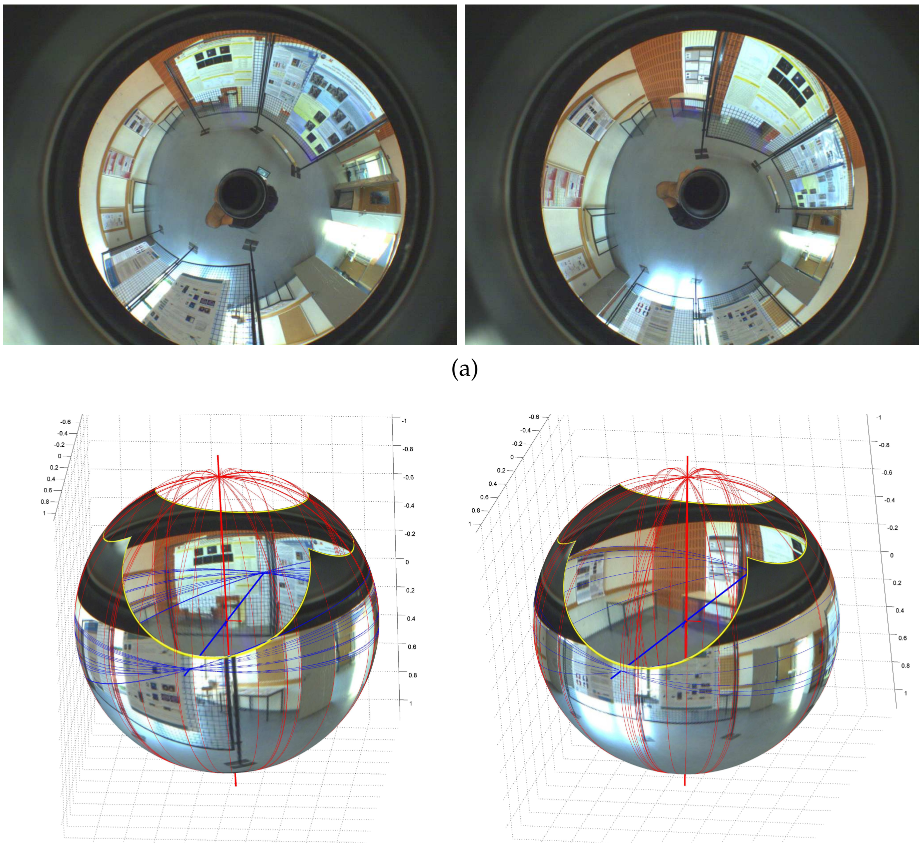


Figure 57: Reconstruction of 3 surfaces in the street view and estimated translation vectors in a unique frame.



(a)

(b)

Figure 58: (a) Two interior catadioptric images and (b) their extracted vanishing directions.

the results. The searching interval for the surface orientation is 30 equi-spaced points. The output of the algorithm for reconstruction of 4 main scene surfaces and 4 estimated translation vectors associated to each surfaces is shown in Fig. 59. The worst estimated translation error among four is 3.6°.

All four surfaces plotted in one uniform framework are shown in Fig. 60. We did not have neither the true motion of the camera nor any metric information from the scene to evaluate the accuracy of the estimated motion and overall reconstruction but the fact that the errors between estimated translations is very small is an indication of the accuracy of the results.

With other images, similar results were obtained. For more results refer to Appendix B.

7.6 CONCLUSION AND OUTLOOK

We built a novel and efficient interactive interface especially suitable for piecewise planar scenes, proving that architecture modeling can be made very simple

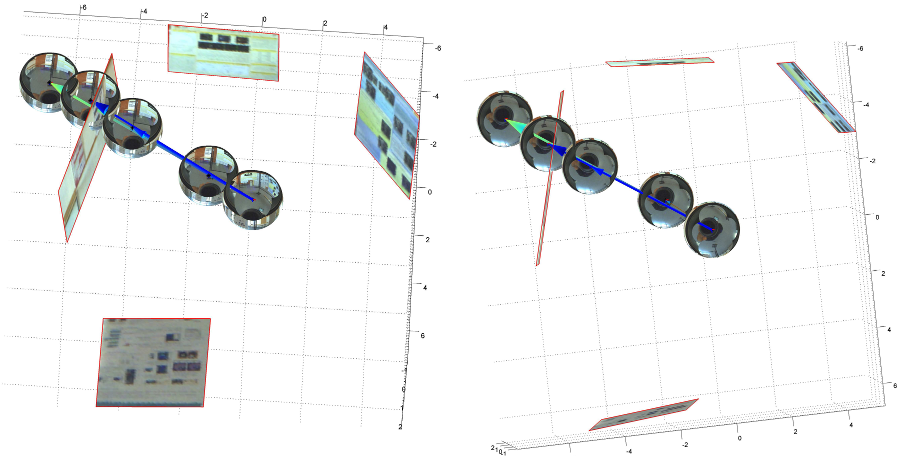


Figure 59: Reconstruction of four main planar surfaces of the scene and 4 estimated translation vectors related to each surfaces from two catadioptric images. Scale of each surface reconstruction is different from others. The right image corresponds to the top view.

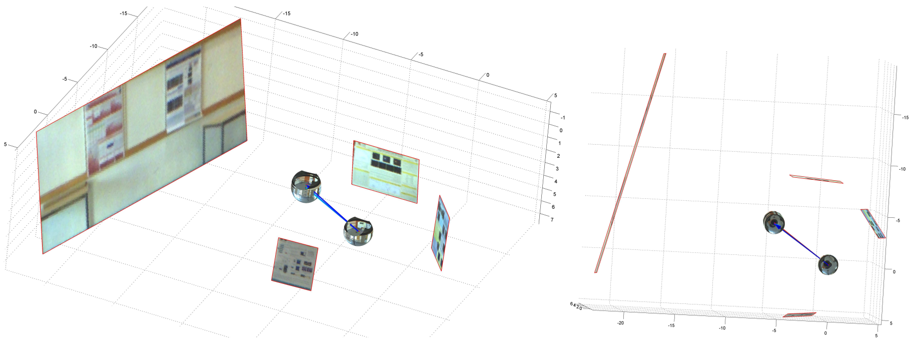


Figure 60: Reconstruction of the scene and translation vectors in a unique frame.

by exploiting the available information from such senses such as vanishing points and the planar nature of the surfaces.

We do not establish many feature correspondences (only one line correspondence), nor do we estimate the optical flow or normal flow (in fact such methods do not work for long-range motion) but we rely on image intensities of the flat surface. Our method works on perspective as well as omnidirectional images; it does not have multiple solution ambiguity and it guaranties one solution as long as the surface/es are well textured.

An interesting application of the proposed algorithm is the extension of the method to estimating the trajectory of a robot along with reconstructing all possible surrounding planar surfaces as the robot moves inside a scene such as a street and takes images from its surrounding, assuming consecutive images share at least one planar surface.

8 GENERAL SUMMARY, FINAL REMARKS AND FUTURE WORKS

In this chapter, we conclude with a short summary and some remarks on possible future works. The primary goal of this thesis was to develop automatic SfM methods for images taken from constructed scenes by any type of central imaging systems including perspective, fish-eye or catadioptric systems.

The work started, in chapter 2, by investigating image formation process for different types of imaging systems by considering basic pinhole camera and its geometry of forming images followed by a brief examination and classification of large FOV cameras and some of their examples. Finally, the unitary sphere and some common calibration techniques for central imaging systems concluded the chapter.

We continued, in chapter 3, by presenting the state of the art on various approaches of line matching along with their classification based on the kind of motion which can be handled by each method.

The rest of the manuscript was then devoted to four main contributions of the thesis as follows:

1. A generic and simple line matching approach for all types of central images including omnidirectional images in constructed scenes under a short baseline motion :

   This part was covered in chapter 4, where we started our objective of developing a generic line matching method for constructed scenes specially applicable to omnidirectional images by tackling the simplified problem where the motion of the system is mainly an arbitrary rotation and the translation of the camera between two views with respect to its distance to the imaged scene is negligible.

   The chapter dealt with the problem of matching lines for all types of central imaging system under a short baseline motion by presenting a generic and simple line matching approach. The method is composed of two main steps of extracting line segments and estimating vanishing directions followed by simultaneously recovering the rotation R and matching lines. Also, two methods for retrieving R, one based on matching vanishing points and the other based on matching any two feature points were proposed. Finally, various experimental results on both synthetic and real images taken by different central cameras as well as an application of the algorithm for creating high resolution panoramic images from high resolution perspective images were also presented.

The state of the art line matching methods use demanding techniques to match lines between images with short baseline and due to deformation of omnidirectional images, they do not even work on these kind of images at all. On the other hand, we developed a very simple and intuitive method which is generic and it works for both perspective and omnidirectional images. It is based on the fact that the motion of the system for a short base-line movement is mainly consist of rotation and in constructed scenes, the rotation can be estimated by matching vanishing points which are easily available in such scenes.

2. A fast and original geometric constraint for matching lines for central images including omnidirectional images in planar constructed scenes insensible to the motion of the camera :

   This part was covered in chapter 5. The problem which we tried to tackle in this chapter was to match two sets of randomly oriented but parallel to a scene plane lines using the location of their intersection with the line at infinity of the plane. Eventually, we proposed a method which has several advantages. It exploits a geometric constraint (the angle between two lines) based on the structure of the scene, without need for the motion of the camera to be known. It is, therefore, also capable of handling disparate views since it employ a constraint which is independent of the motion of the camera. Finally it is computationally very fast and can be run in real-time. Despite all these advantages, the stand-alone algorithm presented here for matching lines between two views of a planar surface is sensitive to the noise and the output of the algorithm is altered as the preset tolerance regions are loosen or tighten.

   The geometric constraints presented in this chapter narrow down the whole set of possible matches (*i.e.* each line in one image is a potential match for each and every line in the other image) to a much smaller set including correct matches. Therefore, the subject is still open and one proper direction to improve the result of current algorithm could be to further separate the correct matches from the rest by considering the spatial and topological configuration of group of lines on unitary spheres or the image planes.

   As a future work, an interesting application of the constraint presented in this chapter for perspective images can be to use it as a filter/booster constraint for leveraging the result of any chosen line matching technique currently available rather than using it as a stand-alone algorithm. To do so, consider the output of the selected technique as a set of putative matches. During the filtering stage, whenever vanishing points are available, the ambiguous or false matches can be detected and filtered out by verifying whether their symmetric homographic transfer error is bigger than a chosen threshold. Similarly, during the boosting stage, for any unmatched segment in one image, its possible correspondence can be found by looking for the segment in the other image with transfer error less than a chosen threshold.

3. A unique and efficient way of computing overlap between two segments on perspective images :

   In the chapter covering this part of the contributions, chapter 6, we introduced a new measure of overlap which increases the speed of the calculating the overlap between two line segments in correspondence. It also allows a denser sampling of the motion space for finding initial guesses for optimization of the non-linear objective function for recovering motion based on line segment correspondences and therefore facilitating the search for a good solution where due to the nature of the scene, sparse sampling followed by optimization does not converge to a good solution. We also suggested, whenever possible, to estimate the rotation between two views using vanishing points and use it as an initial guess in order to reduce the sampling space to two dimensions. We demonstrated this situation by giving the results on two real data sets including the scene where the original algorithm fails to recover the motion.

   As a future work, we can adapt Zhang method to omnidirectional images. Though the original definition of the overlap and our new proposed definition are based on 2d Cartesian coordinates on the image plane, the adaption of these definitions to image sphere should not be very complicated.

4. A simple motion estimation and surface reconstruction algorithm for piece-wise planar scenes applicable to all kinds of central images including omnidirectional images :

   Finally in chapter 7, covering the last but not the least part of this thesis's contributions, we built a novel and efficient interactive interface especially suitable for piece-wise planar scenes, proving that architecture modeling can be made very simple by exploiting the available information from such senses such as vanishing points and the planar nature of the surfaces.

   We do not establish many feature correspondences (only one line correspondence), nor do we estimate the optical flow or normal flow (in fact such methods do not work for long-range motion) but we rely on image intensities of the flat surface. Our method works on perspective as well as omnidirectional images; it does not have multiple solution ambiguity and it guaranties one solution as long as the surface/es are well textured. Also, the simultaneous motion estimation and reconstruction of our method makes it likely that errors are nicely spread over the whole 3D model, compared to more sequential approaches.

   An interesting application of the proposed algorithm is the extension of the method to estimating the trajectory of a robot along with reconstructing all possible surrounding planar surfaces as the robot moves inside a scene such as a street and takes images from its surrounding, assuming consecutive images share at least one planar surface.

Part IV

APPENDIX

A

LINE EXTRACTION; VANISHING POINT EXTRACTION AND MATCHING; GENERALIZED HAUSDORFF DISTANCE

A.1 LINE EXTRACTION

For perspective images, there are two main approaches for extracting line segments. Both methods start by an edge detection step. The first approach starts by an edge detection step followed by detecting the infinite support lines by applying the Hough transform. This method directly inherits the disadvantage of the Hough transform which is many false lines in highly-textured images due to accidental linear arrangements of edge pixels. Furthermore, the location accuracy of the detected lines is not very good. The second approach also starts by an edge detection step followed by fitting the detected edge pixels into straight versus curvilinear structures. The ratio of the length of the line segment divided by the maximum deviation of any point from the line is then used to estimate the quality of the fitting.

For omnidirectional images, however, the straight lines appear as curvilinear structures and none of the perspective methods can be directly applied to extract lines. Methods for extracting lines in catadioptric images can be divided into three groups [13]. The first group is based on fitting the best conic to the points of the line in the image. These algorithms are very sensitive to the occlusion. The second group try to adapt the Hough transform methods for the perspective images to the sphere space and they suffer from the same limitations as in perspective case. The last category is based on specific geometric constraints of paracatadioptric sensors and therefore cannot be applied to other catadioptric systems.

Recently, Bazin et al. [13] proposed a fast central catadioptric line detection algorithm which can be seen as an extension of polygonal approximation of perspective case towards sphere space. This is the algorithm which we have used in our experiments to detect and extract lines.

A.2 A FAST CENTRAL CATADIOPTRIC LINE EXTRACTION

The main idea is based on this geometrical property that a line in space is projected as a great circle on the equivalent sphere. Therefore, after detecting edges in the image and building chains of connected edge pixels (by applying an edge detection method such as Canny edge detector), in order to verify whether these chains correspond to projection of world straight lines, they are projected on the sphere and the great circle constraint is verified. For this, a split and merge algorithm based on the distance between chain points and the plane defining a great circle is applied as follows.

A.2.1 *Division criteria*

Let $n = P_1 \times P_N$ be the normal of the unique great circle which is passing through P_1 and P_N, two endpoints of a chain composed of N pixels after projection onto the unitary sphere. Any other point P_s of the chain is considered to belong to this great circle if the distance between this point and the plane defined by the great circle is less than a threshold $|P_s \centerdot n| \leq Threshold$. If at least 95% of the chain points belong to the great circle, then this chain is considered a line, otherwise, the chain is cut into two sub-chains at the point $argmax(|P_i \centerdot n|)$ which is the furthest point from the great circle. This splitting step stops when the chain is considered a line or the chain length is smaller than a certain threshold.

It often occurs that a line segments split into several smaller, more or less collinear line fragments during building chains of connected edge pixels. These line fragments can be merged based on the fact that they should share the same great circle on the unitary sphere.

A.2.2 *Fusion criteria*

If n_1 and n_2 are the normals of two great circles associated to two extracted lines by the above stage, these two lines are co-linear if they define the same plane: $1 - |n_1 \centerdot n_2| \leq MergeThreshold$. The normal of the great circle associated to merged line is then is obtained from the SVD of matrix containing the pixels of the chains which belong to two line segments.

Thought the fusion step can merge the fragmented segments but it also can merge those segments which may just be accidentally collinear or almost collinear while they are actually belonging to two different lines. See Figure 61(c) for an example. Therefore we skip the merging step during the line extraction and instead, during the line matching process, we merge fragmented line segments by checking co-linearity of the segments in both images sine fragmented segments have to be collinear in both images.

A.3 EXTRACTING AND MATCHING VANISHING POINTS

Vanishing points are points on the plane at infinity and therefore they are invariant to translation. A rotation matrix has three degree of freedom and each vanishing point correspondence provides two rotation constraints. Therefore for estimating R, it is sufficient to have two enough distinct vanishing point correspondences in two views. Reference Bazin et al. [15] has exploited above facts to estimate the rotation of an imaging system in two steps, extraction of vanishing points followed by recovering R by matching these points. For the

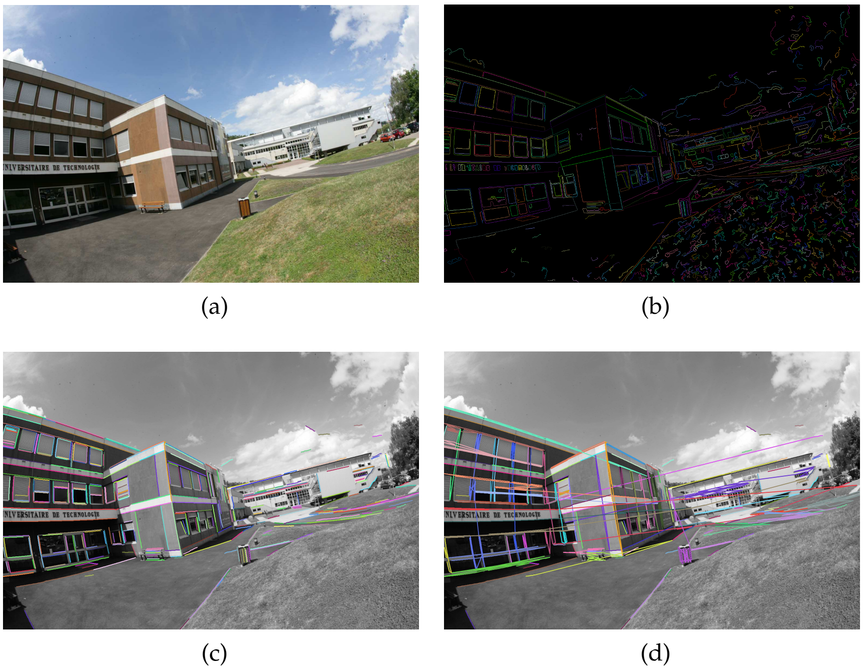


Figure 61: (a) Original image, (b) Extracted chains, (d) Line detection results after division step, (e) The results after fusion step. Note that many merged lines are accidentally collinear or almost collinear.

sake of completeness, we briefly explain their method for extracting vanishing point and matching them.

A.3.1 *Extracting Vanishing Points*

Consider two great circles corresponding to two parallel 3D lines. The intersection of these two great circles (say vector u) corresponds to the common direction of the lines. u should also point at the direction of any other line parallel to these two lines (inside a similarity threshold). Therefore checking for all lines, we can find the number of lines that may share the same direction u. By repeating this procedure for each combination of two great circles, we can compute the vector that corresponds to the highest number of parallel lines (the vanishing point of those lines). To detect the second dominant direction, we remove the lines belong to the first dominant direction and repeat the above steps.

A.3.2 *Matching Vanishing Points*

In their paper, Bazin et al. [14] present a fast and robust method for finding the correspondences of vanishing points in catadioptric images based on comparing the histograms of the spherical regions defined by the vanishing points in the unitary sphere. They propose two ways for splitting the sphere into regions. An

Algorithm A.1 Splitting the unitary sphere using two vanishing points.

```
Given an image taken by a central imaging system and two extracted vanishing points $V_1$ and $V_2$.
for each image pixels $P$ projected on the unitary sphere do
    $pos_1 = P \centerdot V_1$.
    $pos_2 = P \centerdot V_2$.
    if $pos_1 > 0$ and $pos_2 > 0$ then
        $Region_1 \Leftarrow Region_1 + P$;
    end if
    if $pos_1 > 0$ and $pos_2 > 0$ then
        $Region_1 \Leftarrow Region_1 + P$;
    end if
    if $pos_1 > 0$ and $pos_2 < 0$ then
        $Region_2 \Leftarrow Region_2 + P$;
    end if
    if $pos_1 < 0$ and $pos_2 > 0$ then
        $Region_3 \Leftarrow Region_3 + P$;
    end if
    if $pos_1 < 0$ and $pos_2 < 0$ then
        $Region_4 \Leftarrow Region_4 + P$;
    end if
end for
return $Region_1, Region_2, Region_3, Region_4$;
```

intuitive region definition is in clustering each pixel with respect to its nearest vanishing point and the second approach (which we employ in our experiments) is to use the planes defined by vanishing directions and the center of the unitary sphere to split the sphere. The image pixels projected on the unitary sphere are then clustered into regions depending on their position with respect to the planes defined by each vanishing point. The algorithm for two vanishing point is given in Algorithm A.1.

Each region is then represented by a histogram using the intensity of every pixel inside the region and then some distances are used to compute the similarity between two histograms and therefore matching vanishing points. For more details, interested readers are referred to the original paper.

A.4 GENERALIZED HAUSDORFF DISTANCE

Given two sets of points $A = \{a_1, ..., a_m\}$ and $B = \{b_1, ..., b_n\}$, the Hausdorff distance is defined as $H(A, B) = max(h(A, B)\,,\, h(B, A))$ where

$$h(A, B) = \max_{a \in A} \min_{b \in B} ||a - b||$$

The function $h(A, B)$ is called the directed Hausdorff distance from A to B. It identifies the point $a \in A$ that is farthest from any point of B, and measures the distance from a to its nearest neighbor in B. Thus the Hausdorff distance, $H(A, B)$, measures the degree of mismatch between two sets, as it reflects the distance of the point of A that is farthest from any point of B and vice versa.

The Hausdorff distance as defined above is very sensitive to even a single outlying point of A or B and especially for the application considered in this thesis, it can not be directly used. Therefore we use a generalization of the Hausdorff distance given by taking the k-th ranked distance rather than the maximum:

$$h_k(A, B) = \underset{a \in A}{k\text{th}} \min_{b \in B} ||a - b||$$

where kth denotes the k-th ranked value. We used $k = m/2$ for all our experiments which means that the median of the m individual point distances determines the overall distance.

B THE INTERACTIVE 3D RECONSTRUCTION INTERFACE

After automatic estimation of vanishing points and recovering the rotation, the user is asked to select two points on a pair of corresponding line segments on the surface under reconstruction between two views (points number 1 and 2 in the figure 62).

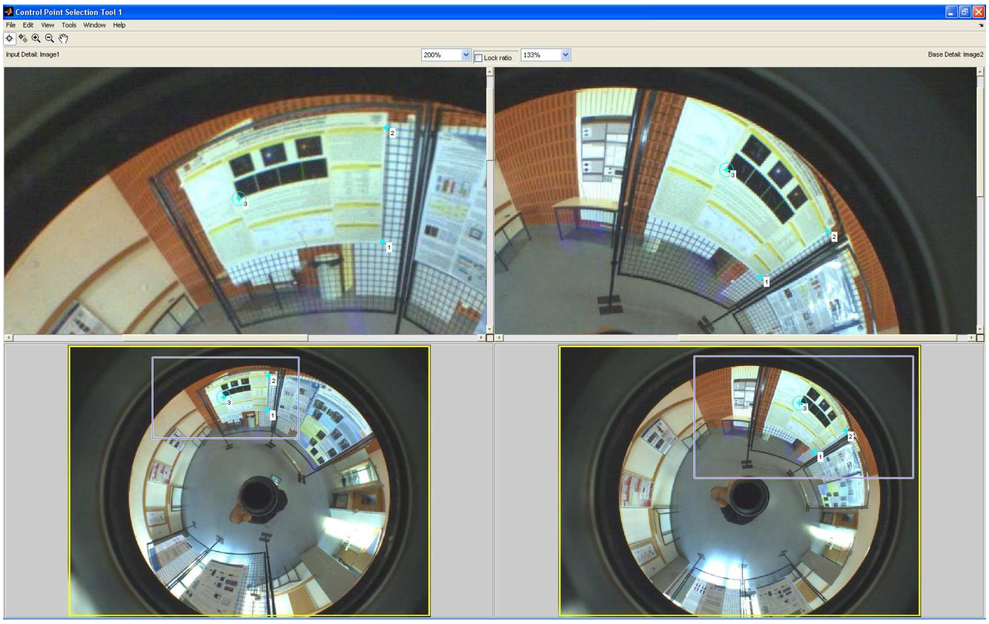
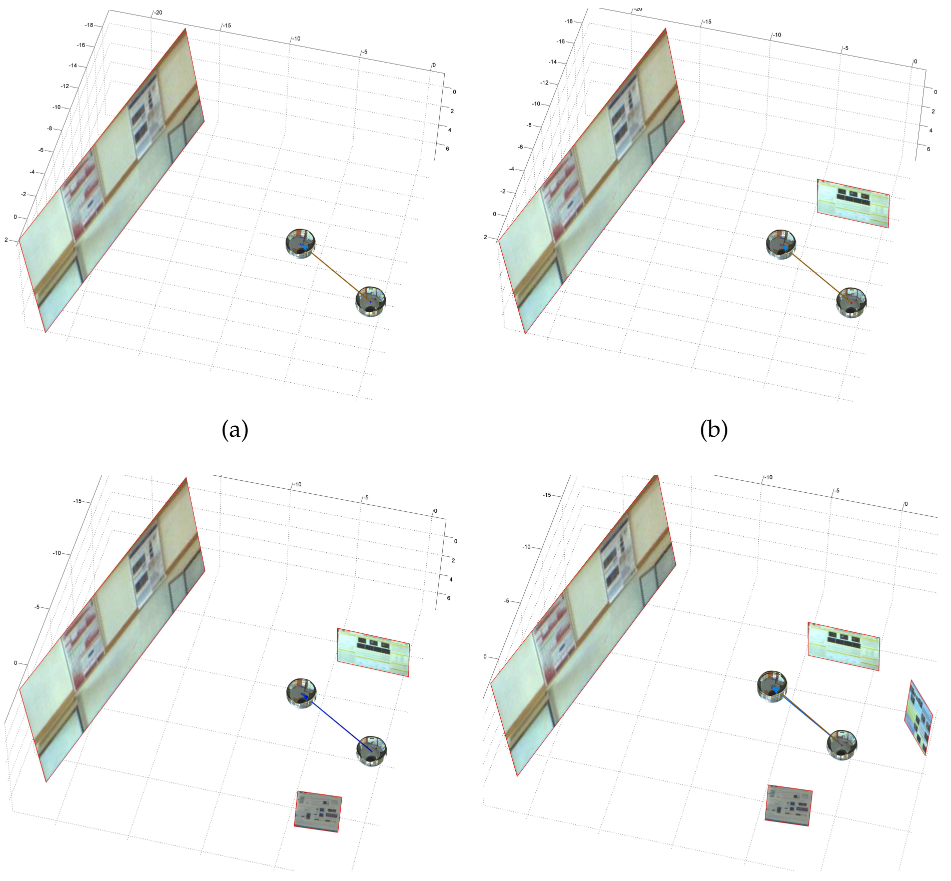


Figure 62: Input interface. The user is asked to select two points on a pair of corresponding line segments on the surface under reconstruction between two views

The lower windows can be used to locate the planar surfaces more easily. Optionally, the user can select a third point to indicate the planar side of the line (point number 3). As the necessary point correspondences are input to the software, it reconstructs the surface and estimate the related translation. The estimated translation related to the first reconstructed surface is considered as the reference and the rest of the reconstructions are scaled to have the same norm for their estimated translations. Figure 62(a-d) shows the progressive reconstruction of the planar scene previously shown in figure 58(a).

Finally for a better dense visualization, wherever appropriate, the reconstructed planes can be extended to intersect other planes. Figure 64 shows another pair of paracatadioptric images from an exterior scene which is mainly composed of three planes.



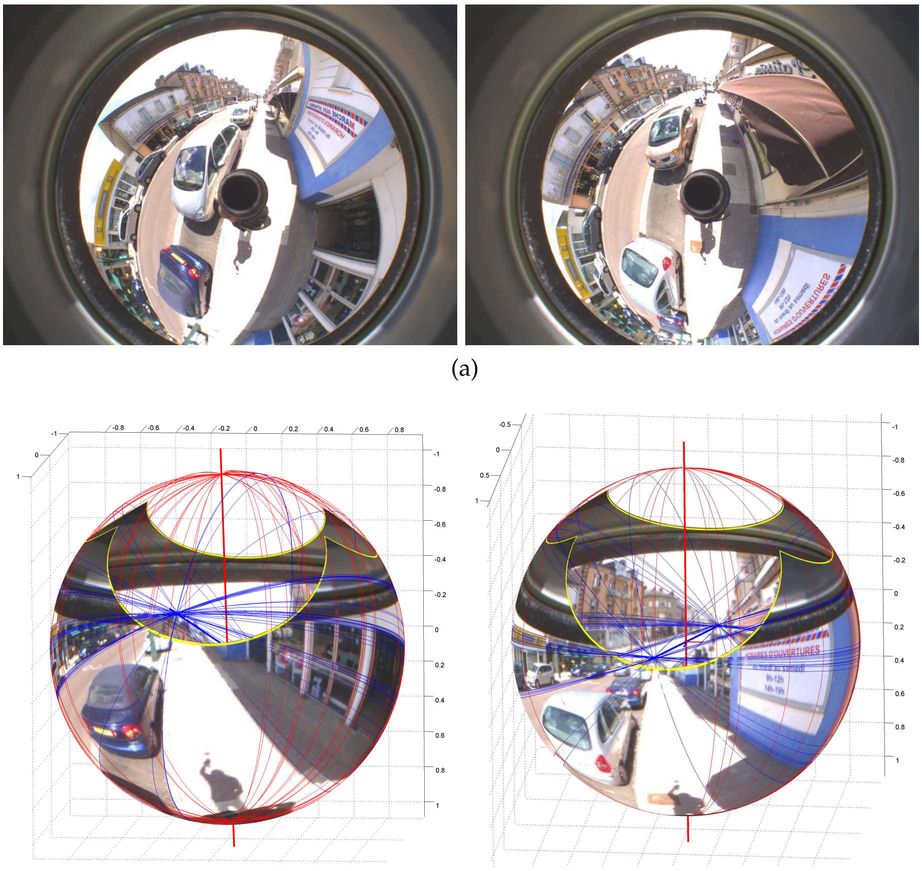
(a)

(b)

(c)

(d)

Figure 63: (a-d) Progressive reconstruction of the scene.



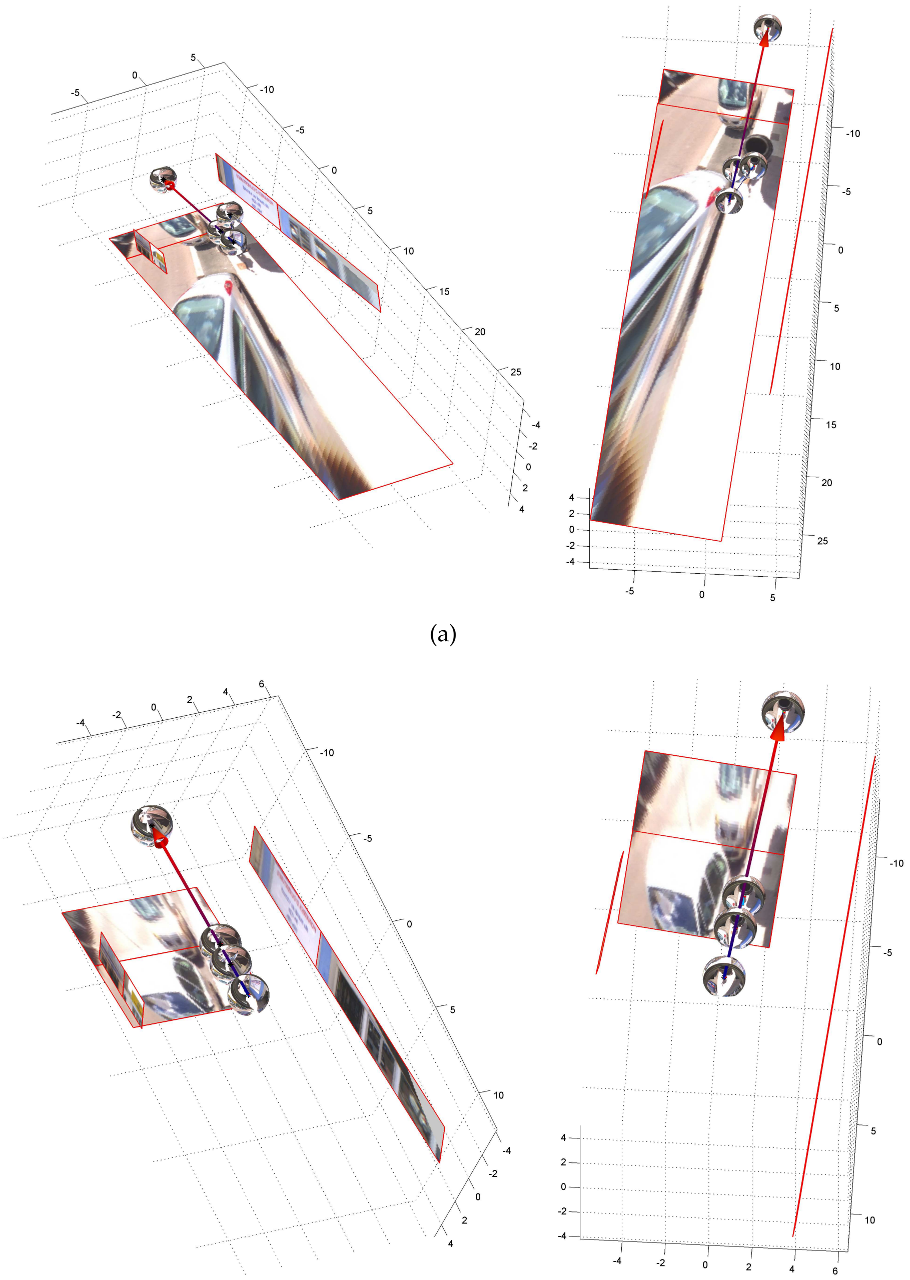
(a)

(b)

Figure 64: (a) Two exterior catadioptric images and (b) their extracted vanishing directions.

During the automatic construction of the plane corresponding to the street floor, the similarity measuring function (Generalized Hausdorff Distance) failed to correctly pick the right mesh images due to the fact that the street can not be approximated anymore as a planar surfaces since big objects such as cars are occupying a large part of the street. After successfully constructing some surfaces, such wrong estimation of translation can be automatically detected by simply verifying whether the estimated translation is coherent with the rest of translation estimated so far and in case of failur, the user is asked to interfere and select the correct mesh images. Figure 65(a) shows the initial results including the wrong reconstruction of the street floor versus the corrected one (b).

All four surfaces plotted in one uniform framework are shown in Fig. 66. For a better visualization, the planes are extended to intersect with each other.



(a)

(b)

Figure 65: Reconstruction of three main planar surfaces of the street scene and 3 estimated translation vectors related to each surfaces from two catadioptric images. (a) The street floor plane reconstruction and the related estimated translation is wrong. (b) The street floor reconstruction is corrected.

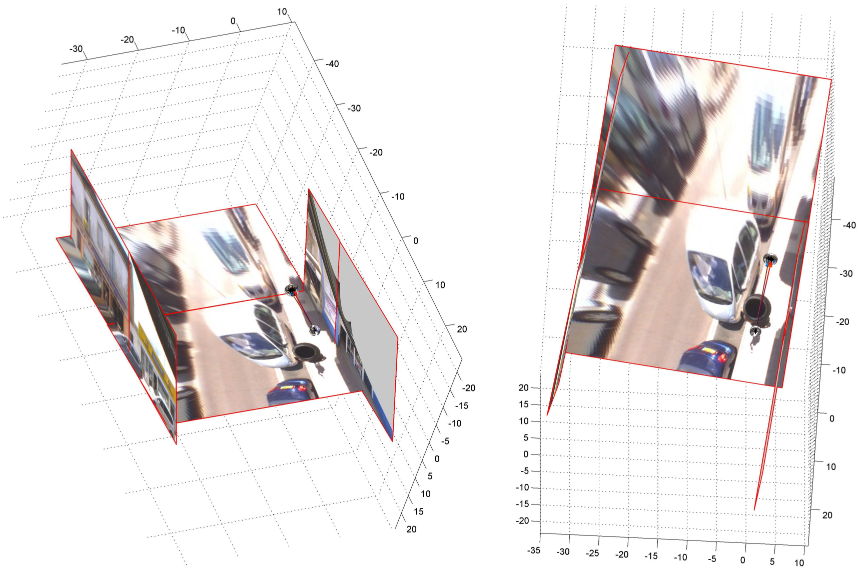


Figure 66: Reconstruction of the street scene and translation vectors in one uniform framework.

BIBLIOGRAPHY

[1] Ayache, N., 1991. Artificial Vision for Mobile Robots: Stereo Vision and Multisensory Perception. MIT Press.

[2] Ayache, N., Faverjon, B., 1987. Efficient registration of stereo images by matching graph descriptions of edge segments. International Journal of Computer Vision 1 (2), 107–132.

[3] Baillard, C., Zisserman, A., 1999. Automatic reconstruction of piecewise planar models from multiple views. In: CVPR. pp. 559–565.

[4] Baker, P.; Ogale, A. F. C., 2004. The argus eye: A new tool for robotics. IEEE Robotics & Automation Magazine 11(4), 31 – 38.

[5] Baker, S., Nayar, S., 1998. A theory of catadioptric image formation. In: IEEE International Conference on Computer Vision. pp. 35–42.

[6] Barreto, J., Araujo, H., 2001. Issues on the geometry of central catadioptric image formation. In: IEEE Computer Vision and Pattern Recognition. pp. II:422–427.

[7] Barreto, J., Araujo, H., August 2005. Geometric properties of central catadioptric line images and their application in calibration. IEEE Trans. Pattern Analysis and Machine Intelligence 27 (8), 1327–1333.

[8] Bartoli, A., Sturm, P., May 2004. The 3d line motion matrix and alignment of line reconstructions. International Journal of Computer Vision 57 (3), 159–178.

[9] Bartoli, A., Sturm, P., 2005. Structure-from-motion using lines: Representation, triangulation, and bundle adjustment. Computer Vision and Image Understanding 100 (3), 416–441.

[10] Basu, A., Licardie, S., 1995. Alternative models for fish-eye lenses. Pattern Recognition Letters 16, 433–441.

[11] Bay, H., Ess, A., Tuytelaars, T., Van Gool, L., 2008. Speeded-up robust features (SURF). Computer Vision and Image Understanding 110 (3), 346–359.

[12] Bay, H., Ferrari, V., Van Gool, L., 2005. Wide-baseline stereo matching with line segments. In: IEEE Computer Vision and Pattern Recognition. pp. I: 329–336.

[13] Bazin, J., Demonceaux, C., Vasseur, P., 2007. Fast central catadioptric line extraction. In: IbPRIA. pp. II: 25–32.

[14] Bazin, J., Kweon, I., Demonceaux, C., Vasseur, P., 2008. Spherical region-based matching of vanishing points in catadioptric images. In: OMNIVIS. pp. xx–yy.

[15] Bazin, J., Kweon, I., Demonceaux, C., Vasseur, P., 2008. Uav attitude estimation by vanishing points in catadioptric images. In: IEEE International Conference on Robotics and Automation. pp. 2743–2749.

[16] Bazin, J. C., Demonceaux, C., Vasseur, P., Kweon, I. S., 2010. Motion estimation by decoupling rotation and translation in catadioptric vision. CVIU 114, 254–273.

[17] Benosman, R., Devars, J., 1998. Panoramic stereovision sensor. In: International Conference on Pattern Recognition. pp. 767–769.

[18] Benosman, R., Maniere, T., Devars, J., 1996. Multidirectional stereovision sensor, calibration and scenes reconstruction. In: International Conference on Pattern Recognition. pp. I: 161–165.

[19] Brassart, E., Delahoche, L., Cauchois, C., Drocourt, C., Pegard, C., Mouaddib, E., 2000. Experimental results got with the omnidirectional vision sensor: Syclop. In: IEEE International Conference on Computer Vision - OMNIVIS. pp. xx–yy.

[20] Brauer Burchardt, C., Voss, K., 2001. A new algorithm to correct fish-eye- and strong wide-angle-lens-distortion from single images. In: IEEE International Conference on Image Processing. pp. I: 225–228.

[21] Burns, J. B., Weiss, R. S., Riseman, E. M., 1992. The non-existence of general-case view-invariants, 120–131.

[22] Cauchois, C.; Brassart, E. D. C. V. P., 1999. Calibration of the omnidirectional vision sensor: Syclop. In: ICRA. Vol. 2. pp. 1287–1292.

[23] Chen, S. E., 1995. Quicktime vr: an image-based approach to virtual environment navigation. In: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques. ACM, New York, NY, USA, pp. 29–38.

[24] Clowes, M., 1971. On seeing things. Vol. 2. pp. 79–116.

[25] Cutler, R., Rui, Y., Gupta, A., Cadiz, J., Tashev, I., He, L.-w., Colburn, A., Zhang, Z., Liu, Z., Silverberg, S., 2002. Distributed meetings: a meeting capture and broadcasting system. In: Proceedings of the tenth ACM international conference on Multimedia. ACM, New York, NY, USA, pp. 503–512.

[26] Drocourt, C. Delahoche, L. M. B. C. A., 2002. Simultaneous localization and map construction method using omnidirectional stereoscopic information. In: ICRA. Vol. 1. pp. 894–899.

[27] Faugeras, O., 2001. Three-dimensional computer vision : a geometric viewpoint. MIT Press, Cambridge, Mass.

[28] Faugeras, O., Luong, Q., 2004. The Geometry of Multiple Images: The Laws That Govern the Formation of Multiple Images of a Scene and Some of Their Applications. MIT Press.

[29] Fermuller, C., Aloimonos, Y., Baker, P., Pless, R., Neumann, J., Stuart, B., 2000. Multi-camera networks: Eyes from eyes. In: IEEE International Conference on Computer Vision - OMNIVIS. pp. xx–yy.

[30] Ferrari, V., Tuytelaars, T., Van Gool, L., 2003. Wide-baseline multiple-view correspondences. In: IEEE Computer Vision and Pattern Recognition. pp. I: 718–725.

[31] Fiala, M., Basu, A., 2002. Line segment extraction in panoramic images. p. 179.

[32] Fiala, M., Basu, A., June 2005. Panoramic stereo reconstruction using non-svp optics. Computer Vision and Image Understanding 98 (3), 363–397.

[33] Fischler, M., Bolles, R., June 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.

[34] Fitzgibbon, A., 2001. Simultaneous linear estimation of multiple view geometry and lens distortion. In: IEEE Computer Vision and Pattern Recognition. pp. I:125–132.

[35] Franceschini, N.; Pichon, J. B. C., 1991. Real time visuomotor control: from flies to robots. In: Fifth International Conference on Advanced Robotics,Robots in Unstructured Environments. Vol. 2. pp. 931–935.

[36] Gallup, D., Frahm, J., Mordohai, P., Yang, Q., Pollefeys, M., 2007. Real-time plane-sweeping stereo with multiple sweeping directions. In: CVPR. pp. 1–8.

[37] Gao, C., Hua, H., Ahuja, N., February 2010. A hemispherical imaging camera. Computer Vision and Image Understanding 114 (2), 168–178.

[38] Gaspar, J., Winters, N., Santos-victor, J., 2000. Vision-based navigation and environmental representations with an omnidirectional camera. IEEE Transactions on Robotics and Automation 16, 890–898.

[39] Georgis, N., Petrou, M., Kittler, J., January 1998. On the correspondence problem for wide angular separation of noncoplanar points. Image and Vision Computing 16 (1), 35–41.

[40] Geyer, C., Daniilidis, K., 2000. A unifying theory for central panoramic systems and practical implications. In: European Conference on Computer Vision. pp. II: 445–461.

[41] Gluckman, J., Nayar, S., 1998. Ego-motion and omnidirectional cameras. In: ICCV. pp. 999–1005.

[42] Gluckman, J., Nayar, S., 1999. Planar catadioptric stereo: Geometry and calibration. In: IEEE Computer Vision and Pattern Recognition. pp. I: 22–28.

[43] Gluckman, J., Nayar, S., February 2002. Rectified catadioptric stereo sensors. IEEE Trans. Pattern Analysis and Machine Intelligence 24 (2), 224–236.

[44] Gluckman, J., Nayar, S., Thoresz, K., 1998. Real-time omnidirectional and panoramic stereo. pp. 299–303.

[45] Goedeme, T., Tuytelaars, T., Van Gool, L., 2004. Fast wide baseline matching for visual navigation. In: IEEE Computer Vision and Pattern Recognition. pp. I: 24–29.

[46] Greguss, P., 1986. Panoramic imaging block for three-dimensional space.

[47] Gros, P., Bournez, O., Boyer, E., February 1998. Using local planar geometric invariants to match and model images of line segments. Computer Vision and Image Understanding 69 (2), 135–155.

[48] Grossberg, M., Nayar, S., 2001. A general imaging model and a method for finding its parameters. In: IEEE International Conference on Computer Vision. pp. II: 108–115.

[49] Gu, W., Yang, J., Huang, T., May 1987. Matching perspective views of a polyhedron using circuits. IEEE Trans. Pattern Analysis and Machine Intelligence 9 (3), 390–400.

[50] Guzman-Arenas, A., Guzmaan, A., 1968. Computer recognition of three-dimensional objects in a visual scene. Ph.D. thesis, Cambridge, MA, USA.

[51] Hartley, R., 1995. A linear method for reconstruction from lines and points. In: IEEE International Conference on Computer Vision. pp. 882–887.

[52] Hartley, R., Zisserman, A., June 2004. Multiple View Geometry in Computer Vision. Cambridge University Press.

[53] Heikkila, J., October 2000. Geometric camera calibration using circular control points. IEEE Trans. Pattern Analysis and Machine Intelligence 22 (10), 1066–1077.

[54] Heikkila, J., Silven, O., 1997. A four-step camera calibration procedure with implicit image correction. In: IEEE Computer Vision and Pattern Recognition. pp. 1106–1112.

[55] Holt, R., Netravali, A., June 1996. Uniqueness of solutions to structure and motion from combinations of point and line correspondences. In: JVCIR. Vol. 7. pp. 126–136.

[56] Hong, J.; Tan, X. P. B. W. R. R. E., 1991. Image-based homing. In: ICRA. Vol. 1. pp. 620–625.

[57] Horaud, R., Skordas, T., November 1989. Stereo correspondence through feature grouping and maximal cliques. IEEE Trans. Pattern Analysis and Machine Intelligence 11 (11), 1168–1180.

[58] Hua, H., Ahuja, N., 2001. A high-resolution panoramic camera. In: IEEE Computer Vision and Pattern Recognition. pp. I:960–967.

[59] Hua, H., Ahuja, N., Gao, C., February 2007. Design analysis of a high-resolution panoramic camera using conventional imagers and a mirror pyramid. IEEE Trans. Pattern Analysis and Machine Intelligence 29 (2), 356–361.

[60] Huang, T., Lee, C., 1989. Motion and structure from orthographic projections. IEEE Transactions on Pattern Analysis and Machine Intelligence 11 (5), 536–540.

[61] Huffman, D., 1977. Impossible objects as non-sense sentences. In: CMetImAly. pp. 338–366.

[62] Huttenlocher, D., Klanderman, G., Rucklidge, W., 1992. Comparing images using the hausdorff distance under translation. In: CVPR. pp. 654–656.

[63] INRIA, 2002. Syntim stereo images.
URL http://perso.lcpc.fr/tarel.jean-philippe/syntim/paires.html

[64] Iwerks, U., Oaks, S., 1964. Panoramic motion picture camera arrangement.

[65] Jiang, N., Tan, P., Cheong, L.-F., December 2009. Symmetric architecture modeling with a single image. ACM Trans. Graph. 28, 113:1–113:8.

[66] Kannala, J., Brandt, S., August 2006. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. IEEE Trans. Pattern Analysis and Machine Intelligence 28 (8), 1335–1340.

[67] Kawanishi, T., Yamazawa, K., Iwasa, H., Takemura, H., Yokoya, N., 1998. Generation of high-resolution stereo panoramic images by omnidirectional imaging sensor using hexagonal pyramidal mirrors. In: ICPR. pp. 485–489.

[68] Krishnan, A., Ahuja, N., 1996. Panoramic image acquisition. In: Proceedings of the 1996 Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, Washington, DC, USA, pp. 379–.

[69] Liebowitz, D., Zisserman, A., 1999. Combining scene and auto-calibration constraints. In: ICCV. pp. 293–300.

[70] Lin, S., Bajcsy, R., May 2006. Single-view-point omnidirectional catadioptric cone mirror imager. IEEE Trans. Pattern Analysis and Machine Intelligence 28 (5), 840–845.

[71] Lourakis, M., Halkidis, S., Orphanoudakis, S., June 2000. Matching disparate views of planar surfaces using projective invariants. Image and Vision Computing 18 (9), 673–683.

[72] Lowe, D., November 2004. Distinctive image features from scale-invariant keypoints. Vol. 60. pp. 91–110.

[73] Majumder, A., Seales, W. B., Gopi, M., Fuchs, H., 1999. Immersive teleconferencing: a new algorithm to generate seamless panoramic video imagery. In: Proceedings of the seventh ACM international conference on Multimedia (Part 1). ACM, New York, NY, USA, pp. 169–178.

[74] Mauthner, T., Fraundorfer, F., Bischof, H., 2006. Region matching for omnidirectional images using virtual camera planes. In: Proc. of Computer Vision Winter Workshop. pp. 93–98.

[75] McIntosh, J., Mutch, K., 1988. Matching straight lines. Vol. 43. pp. 386–408.

[76] McIntosh, J., Mutch, K., September 1988. Matching straight lines. Computer Vision Graphics and Image Processing 43 (3), 386–408.

[77] McMillan, L., Bishop, G., 1995. Plenoptic modeling: An image-based rendering approach. In: SIGGraph. pp. 39–46.

[78] Medioni, G., 1983. Matching linear features of images and maps. Ph.D. thesis.

[79] Medioni, G., Nevatia, R., November 1984. Matching images using linear features. IEEE Trans. Pattern Analysis and Machine Intelligence 6 (6), 675–685.

[80] Medioni, G., Nevatia, R., July 1985. Segment-based stereo matching. Computer Vision Graphics and Image Processing 31 (1), 2–18.

[81] Mei, C., Rives, P., April 2007. Single view point omnidirectional camera calibration from planar grids. In: IEEE International Conference on Robotics and Automation.

[82] Micusik, B., Pajdla, T., 2003. Estimation of omnidirectional camera model from epipolar geometry. In: IEEE Computer Vision and Pattern Recognition. pp. I: 485–490.

[83] Mosaddegh, S., 2008. Line matching across catadioptric images. Master's thesis.

[84] Mosaddegh, S., Fofi, D., Vasseur, P., 2010. Simultaneous ego-motion estimation and reconstruction of piecewise planar scenes from two views. Tech. Rep. 10-01, Le2i, UMR CNRS 5158, Universite de Bourgogne.

[85] Mosaddegh, S., Fofi, D., Vasseur, P., Ainouz, S., 2008. Line matching across catadioptric images. In: OMNIVIS. pp. xx–yy.

[86] Mouaddib, E.M.; Marhic, B., 2000. Geometrical matching for mobile robot localisation. In: ICRA. pp. 542–552.

[87] Mouaddib, E. M., Sagawa, R., Echigo, T., Yagi, Y., 2005. Stereovision with a single camera and multiple mirrors. In: ICRA. pp. 800–805.

[88] Murray, D., March 1995. Recovering range using virtual multicamera stereo. Computer Vision and Image Understanding 61 (2), 285–291.

[89] Nagahara, H., Yoshida, K., Yachida, M., 2007. An omnidirectional vision sensor with single view and constant resolution. In: IEEE International Conference on Computer Vision. pp. 1–8.

[90] Nagao, K., Grimson, W., April 1998. Affine matching of planar sets. Computer Vision and Image Understanding 70 (1), 1–22.

[91] Nalwa, V. S., 1996. A true omnidirectional viewer. Bell Labs Technical Journal.

[92] Navab, N., Faugeras, O., 1997. The critical sets of lines for camera displacement estimation: A mixed euclidean-projective and constructive approach. IJCV 23, 17–44.

[93] Nayar, S., 1997. Catadioptric omnidirectional camera. In: IEEE Computer Vision and Pattern Recognition. pp. 482–488.

[94] Nayar, S., Peri, V., 2001. Folded catadioptric cameras. pp. 103–119.

[95] Nene, S., Nayar, S., 1998. Stereo with mirrors. In: IEEE International Conference on Computer Vision. pp. 1087–1094.

[96] Netravali, A., Huang, T., 2002. Motion and structure from feature correspondences: A review. In: AIPU. pp. 331–348.

[97] Nicholas J. Wade, S. F., 2001. The eye as an optical instrument: from camera obscura to helmholtz's perspective. Perception 30, 1157 – 1177.

[98] Onoe, Y., Yamazawa, K., Yokoya, N., Takemura, H., 1998. Visual surveillance and monitoring system using an omnidirectional video camera. In: International Conference on Pattern Recognition. pp. Vol I: 588–592.

[99] Orghidan, R., 2005. Catadioptric stereo based on structured light projection. Ph.D. thesis, Department of Electronics, Computer Science and Automatic Control, University of Girona.

[100] Orphanoudakis, S., Halkidis, S., Lourakis, M., 1998. Matching disparate views of planar surfaces using projective invariants. pp. I: 94–104.

[101] Peer, P., Solina, F., April 2002. Panoramic depth imaging: Single standard camera approach. International Journal of Computer Vision 47 (1-3), 149–160.

[102] Pegard, C.; Mouaddib, E., 1996. A mobile robot using a panoramic view. Vol. 1. pp. 89–94.

[103] Pollefeys, M., Koch, R., Van Gool, L., August 1999. Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters. Vol. 32. pp. 7–25.

[104] Powell, I., 1997. Panoramic fish-eye imaging system.

[105] Ramalingam, S., Sturm, P., 2008. Minimal solutions for generic imaging models. In: IEEE Computer Vision and Pattern Recognition. pp. 1–8.

[106] Ramalingam, S., Sturm, P., Boyer, E., 2006. A factorization based self-calibration for radially symmetric cameras. In: 3DPVT. pp. 480–487.

[107] Ramalingam, S., Sturm, P., Lodha, S., February 2010. Generic self-calibration of central cameras. Computer Vision and Image Understanding 114 (2), 210–219.

[108] Roberts, L. G., 1963. Machine perception of three-dimensional solids. Ph.D. thesis, Massachusetts Institute of Technology.

[109] Rothwell, C., Zisserman, A., Forsyth, D., Mundy, J., September 1995. Planar object recognition using projective shape representation. International Journal of Computer Vision 16 (1), 57–99.

[110] Saleh Mosaddegh, David Fofi, P. V., 2011. Line matching between disparate views of planar surfaces. In: International Conference on Computer Applications and Network Security (ICCANS 2011).

[111] Scaramuzza, D., Criblez, N., Martinelli, A., Siegwart, R., 2008. Robust feature extraction and matching for omnidirectional images. In: Laugier, C., Siegwart, R. (Eds.), Field and Service Robotics. Vol. 42 of Springer Tracts in Advanced Robotics. Springer Berlin-Heidelberg, pp. 71–81.

[112] Scaramuzza, D., Martinelli, A., Siegwart, R., 2006. A Toolbox for Easily Calibrating Omnidirectional Cameras. In: Iros. Benjing China.

[113] Scaramuzza, D., Martinelli, A., Siegwart, R., 2006. A flexible technique for accurate omnidirectional camera calibration and structure from motion. p. 45.

[114] Scaramuzza, D., Siegwart, R., Martinelli, A., February 2009. A robust descriptor for tracking vertical lines in omnidirectional images and its use in mobile robotics. Int. J. Rob. Res. 28, 149–171.

[115] Schmid, C., Zisserman, A., 1997. Automatic line matching across views. In: IEEE Computer Vision and Pattern Recognition. pp. 666–671.

[116] schon Lin, S., 2003. High resolution catadioptric omni-directional stereo sensor for robot vision. In: ICRA. pp. 1694–1699.

[117] Shabayek, A. E. R., 2009. Non-central catadioptric sensors auto-calibration. Master's thesis.

[118] Shah, S., Aggarwal, J., November 1996. Intrinsic parameter calibration procedure for a (high-distortion) fish-eye lens camera with distortion model and accuracy estimation. Pattern Recognition 29 (11), 1775–1788.

[119] Southwell, D. Vandegriend, B. B. A., 1996. A conical mirror pipeline inspection system. In: ICRA. Vol. 4. pp. 3253–3258.

[120] Southwell, D., Basu, A., Fiala, M., Reyda, J., 1996. Panoramic stereo. In: ICPR. Vol. A. pp. 378–382.

[121] Spacek, L., 2005. A catadioptric sensor with multiple viewpoints. Vol. 51. pp. 667–674.

[122] Sturm, P., 2000. A method for 3d reconstruction of piecewise planar objects from single panoramic images. In: IEEE International Conference on Computer Vision - OMNIVIS. pp. xx–yy.

[123] Sturm, P., Maybank, S., 1999. A method for interactive 3d reconstruction of piecewise planar objects from single images. In: British Machine Vision Conference. pp. 265–274.

[124] Sturm, P., Maybank, S., 1999. On plane-based camera calibration: A general algorithm, singularities, applications. pp. I: 432–437.

[125] Sturm, P., Ramalingam, S., 2004. A generic concept for camera calibration. In: European Conference on Computer Vision. pp. Vol II: 1–13.

[126] Sturm, P., Ramalingam, S., Tardif, J.-P., Gasparini, S., 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.

[127] Tardif, J., 2009. Non-iterative approach for fast and accurate vanishing point detection. In: IEEE International Conference on Computer Vision. pp. 1250–1257.

[128] Tardif, J., Sturm, P., Trudeau, M., Roy, S., September 2009. Calibration of cameras with radially symmetric distortion. IEEE Trans. Pattern Analysis and Machine Intelligence 31 (9), 1552–1566.

[129] Taylor, C., Kriegman, D., 1995. Structure and motion from line segments in multiple images. T-PAMI 17, 1021–1032.

[130] Tell, D., Carlsson, S., 2000. Wide baseline point matching using affine invariants computed from intensity profiles. In: European Conference on Computer Vision. pp. I: 814–828.

[131] Thirthala, S., Pollefeys, M., 2005. Multi-view geometry of 1d radial cameras and its application to omnidirectional camera calibration. In: IEEE International Conference on Computer Vision. pp. II: 1539–1546.

[132] Tsai, F., 1993. A probabilistic approach to geometric hashing using line features. Ph.D. thesis.

[133] Tsai, F., March 1994. Geometric hashing with line features. Pattern Recognition 27 (3), 377–389.

[134] Tsai, F., January 1996. A probabilistic approach to geometric hashing using line features. Computer Vision and Image Understanding 63 (1), 182–195.

[135] Tsai, R., 1987. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. IEEE Trans. Robotics and Automation 3 (4), 323–344.

[136] Ullman, S., 1979. The Interpretation of Visual Motion. MIT Press.

[137] Vasseur, P., Demonceaux, C., 2010. Central catadioptric line matching for robotic applications. In: ICRA. pp. 2562–2567.

[138] Wang, L., Neumann, U., You, S., 2009. Wide-baseline image matching using line signatures. In: IEEE International Conference on Computer Vision. pp. 1311–1318.

[139] Wang, Z., Wu, F., Hu, Z., May 2009. Msld: A robust descriptor for line matching. Pattern Recognition 42 (5), 941–953.

[140] Wei, S.-C., Yagi, Y., Yachida, M., 1998. Building local floor map by use of ultrasonic and omni-directional vision sensor. In: ICRA. pp. 2548–2553.

[141] Weng, J., Huang, T., Ahuja, N., 1992. Motion and structure from line correspondences; closed-formsolution, uniqueness, and optimization. T-PAMI 14 (3), 318–336.

[142] Winters, N., Gaspar, J., Lacey, G., Santos-Victor, J., 2000. Omni-directional vision for robot navigation. In: Proceedings of the IEEE Workshop on Omnidirectional Vision. IEEE Computer Society, Washington, DC, USA, pp. 21–.

[143] Wolfson, H., Lamdan, Y., 1988. Geometric hashing: A general and efficient model-based recognition scheme. In: IEEE International Conference on Computer Vision. pp. 238–249.

[144] Xiong, Y., Turkowski, K., 1997. Creating image based vr using a self-calibrating fisheye lens. In: IEEE Computer Vision and Pattern Recognition. pp. 237–243.

[145] Yachida, M., 1998. Omnidirectional sensing and combined multiple sensing. p. Sensing and Rendering Real Scenes.

[146] Yagi, Y., Kawato, S., 1990. Panoramic scene analysis with conic projection. pp. xx–yy.

[147] Yagi, Y., Yachida, M., 2002. Omnidirectional sensing for human interaction. In: IEEE International Conference on Computer Vision - OMNIVIS. pp. 121–127.

[148] Yamazawa, K., Yagi, Y., Yachida, M., 1993. Omindirectional imaging with hyperboliodal projection. pp. 1029–1034.

[149] Yamazawa, K., Yagi, Y., Yachida, M., 1995. Obstacle detection with omnidirectional image sensor hyperomni vision. pp. 1062–1067.

[150] Ying, X., Hu, Z., October 2004. Catadioptric camera calibration using geometric invariants. IEEE Trans. Pattern Analysis and Machine Intelligence 26 (10), 1260–1271.

[151] Zhang, Z., 1994. A new and efficient iterative approach to image matching. In: International Conference on Pattern Recognition. pp. A:563–565.

[152] Zhang, Z., March 1994. Token tracking in a cluttered scene. Image and Vision Computing 12 (2), 110–120.

[153] Zhang, Z., 1995. Estimating motion and structure from correspondences of line segments between two perspective images. In: IEEE International Conference on Computer Vision. pp. 257–262.

[154] Zhang, Z., 1998. A flexible new technique for camera calibration. Tech. rep.

[155] Zhang, Z., Weiss, R., Riseman, E., 1991. Feature matching in 360 waveforms for robot navigation. In: IEEE Computer Vision and Pattern Recognition. pp. 742–743.

[156] Zheng, J., Tsuji, S., 1990. Panoramic representation of scenes for route understanding. In: International Conference on Pattern Recognition. pp. I: 161–167.