Centro de Investigación en Matemáticas, A. C.

Automatic Camera Calibration and Structure Recovery of Multi-planar Scenes Using Two Views

TESIS

QUE PARA OBTENER EL GRADO ACADÉMICO DE

Maestro en Ciencias
Con orientación en
Ciencias de la Computación

PRESENTA

Flavio Mario Santés López

Director de Tesis:
Dr. Javier Flavio Vigueras Gómez

Guanajuato, Gto., Marzo de 2010

Agradecimientos

A todos aquellos que directa o indirectamente han contribuido al fortalecimiento de Instituciones como el Centro de Investigación en Matemáticas A. C. (CIMAT A.C.) y el Consejo Nacional de Ciencia y Tecnología (CONACYT).

A todos aquellos que con sus acciones coadyuvaron a la realización de este trabajo.

Mario.

Contents

List of Figures

List of Algorithms

Abstract

In man-made environments, the presence of planar objects is common. These surfaces are described by walls, doors and many objects such as books, desks and other furniture. Recently, *multiple view geometry* has been used by the Computer Vision Community as a tool in order to describe the elements of multi-planar scenes and recover the calibration coefficients of the cameras that are used to analyze the three dimensional scene. Calibration deals with the recovering of the focal distance (intrinsic calibration) and localization parameters of a camera (extrinsic calibration). Reconstruction of a multi-planar scene consists of determining the planar equation of each planar structure.

The aim of this work is to show how by using linear methods is possible to recover the calibration coefficients and describe the elements in the multi-planar scene using only two different views or images. Related work dealing with camera localization and structure recovery presents non-linear formulations that need an initialization guess in order to solve these problems. Our work introduces a multilinear form that captures all the coefficients needed to solve the calibration and structure recovery problems.

This work presents a novel approach to solve the calibration and structure recovery problems, based on linear methods that exploits geometrical and algebraic constraints induced by rigidity and planarity in the scene. Linearity allows our approach to be suitable for real-time camera calibration and scene reconstruction. Furthermore, we do not compute the two-view fundamental matrix. Therefore, we do not face stability problems commonly associated with explicit epipolar geometry computation.

Chapter 1

Introduction

This research addresses the problem of automatic calibration of two cameras that are observing the same three dimensional scene conformed by planar surfaces. We also want to know how these planes are located in the scene, i.e. we wish to describe the scene. Scenes with planar surfaces or planes such as walls, doors, books, desks or even buildings whose facade is *almost* planar, can be reconstructed using the theory developed in this work. These kind of surfaces are easily found in indoor environments such as offices or when walking out through streets in almost any city.

Something fascinating of working with planar surfaces is that every plane can be represented by three degrees of freedom. Determining these parameters is enough in order to represent all the points in the surface.

Given any three dimensional scene conformed by planes, it is easy to describe the relation between planes and their projection due to the camera. So far, camera models and transfer functions between multi-planar scenes and images or views are well-understood. Indeed, linear mathematical models have been developed by the computer vision community. These mathematical models allow us to analyze how objects inside a scene are mapped to an image. A natural way to solve these kind of problems may involve the inclusion of constraints that arise in a natural way as a consequence of the elements in the scene. These constrained problems still remain open and although they have been analyzed, there is not a general solution for all of them yet. Furthermore, linear methods that incorporate these constraints are rarely found in the current literature. Linear methods are interesting due to the fact that can be applied in real-time applications, they commonly offer a closed-form solution being free of an initialization guess and do not present problems with local minima as in non-linear formulations.

The use of planar surfaces for camera calibration and structure recovery have recently received attention from the Computer Vision Community [1, 6, 7, 10, 17, 23, 25]. Some research has been conducted in theoretical [2, 6, 11] and practical fields [1, 16, 17]. Other tasks such as image segmentation [1, 23] or camera calibration [25] have also been studied. Furthermore, map building is an area where multi-planar scenes have been used by roboticists [20].

Multiple view geometry is a branch of the computer vision field. This sub-field refers to research that is conducted by means of analyzing several images of the same scene acquired either by a moving camera or a camera array. Common tasks that can be solved by means of these images or views are the ones described above.

This work presents novel results derived of the analysis of two views of the same rigid scene that contains at least two planar structures or planes. Our approach makes use of key points or interest points in both views. Key points are picture elements with some outstanding quality when comparing with neighboring elements. The usual way to work with two views based on key points, consists of the following steps [10]:

1. Choose the camera model.
2. Determine how to conduct the matching process between key points.
3. Describe the transfer functions between the scene and each view and between views.
4. Recover the desired parameters (intrinsic and extrinsic camera parameters and scene reconstruction).

The camera model used in this work is the well-known pinhole camera model [10]. We define the *camera calibration problem* as the task that consists in determining the intrinsic coefficients of the camera[1] and its position with respect to a global three dimensional frame of reference. When considering square picture elements of pixels, only three parameters are computed in order to obtain the camera matrix. These parameters are the focal distance and two coefficients used to

[1] Also known as coefficients of the camera matrix.

describe the 2D coordinates of the principal point[2].

The *camera localization problem* consists in determining the camera *pose* where the images (scene projections) were taken. A 3D global frame of reference is imposed and is used to describe the elements in the scene (structure recovery or scene reconstruction problem). Although image segmentation is a widely studied topic, the case of images describing multi-planar structures remains open. Figure 1.1 shows two different views of the same multi-planar scene.

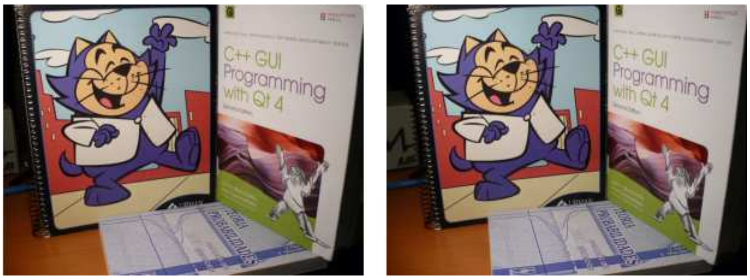

Figure 1.1: Two different views of the same multi-planar scene.

We propose an iterative linear algorithm exploiting geometrical and algebraic constraints induced by rigidity and planarity in the scene. Our approach allows us to deal with localization and reconstruction problems using only linear systems of equations, instead of solving a multi-linear problem or a non-linear problem with the corresponding instability due to errors in the initial localization guess.

The proposed framework makes use of features extracted from two images and matched by correlation. Image features are segmented considering coplanar constraints for point transferring. Furthermore, linear functions are used for epipolar geometry recovery, without explicit computation of the fundamental matrix.

One geometrical fact that is exploited in our work, consists in computing the intersections between all the planar surfaces that are projected in the views. These intersections are seen as straight lines on each image, and they allow us to carry out the two problems covered by this work: camera calibration and structure recovery.

[2] A formal description of these concepts is found in Chapter 2.

1.1 Document organization

This chapter ends by presenting the problem definition, gives a list of contributions of our work and makes a review of the most influential works in our research, including some comparisons between them. At the end of this chapter, we present an overview of our approach.

The mathematical foundations that are necessary in order to understand our approach are described in the second chapter. We also introduce the camera model that is used to represent the physical devices. This chapter also covers two well known linear transfer functions that are induced by the planar structures. Other mathematical objects derived from these transfer functions are presented, including the description of the intersection between two planes.

The third chapter exhibits the theory and the linear method developed in this research.

Experiments done on real data are presented in the fourth chapter. Conclusions and possible future lines of research are presented in the last chapter.

1.2 Problem definition

Given two images of the same multi-planar scene, we want to:

1. Determine the focal distance of the cameras.
2. Determine the position and orientation of each camera when these images are taken.
3. Reconstruct the scene, i.e. compute the parameters that describe each planar surface.
4. Segment the images, in order to label the image features that belong to each plane.

Points 1 and 2 refer to the camera calibration problem (intrinsic and extrinsic calibration, respectively), 3 and 4 belong to the structure recovery task (3D reconstruction and planar segmentation). Although the computation of intrinsic parameters of a camera commonly includes the principal point, we assume that it is fixed at the center of the image [25].

Focal distances of both cameras are needed in order to recover localization and reconstruction parameters.

1.3 Contributions

In [6], Faugeras and Lustman present a framework that allows to determine the localization of two cameras and recover the parameters of one planar surface that is observed by two cameras. This is one of the first attempts to show how linear methods can be used to recover motion and structure parameters.

Our research has been deeply influenced by [6]. The main differences between our work and [6] consists in considering more than one planar surface in the scene and consistency between the different projective planar transformations. This may seem as simple as adding one dimension to a linear least squares problem, but indeed when incorporating more than one plane, some considerations should be taken, i.e. a transfer function induced by a plane codify parameters of camera properties, camera localization and plane equation. Several transfer functions computed with the same pair of images, induced by an equal number of different planes, should codify the same intrinsic camera and localization parameters. This research addresses robust computation of the common parameters of transfer functions and robust recovery of the scene structure. Once all the transfer functions are computed and codify the same common parameters, it is possible to recover the focal distances of the cameras.

Our original contributions are:

1. A system of linear equations in order to compute the first epipole, i.e. the epipole in the first view, and the line of intersection between each pair of planes.
2. A linear system of equations that allow us to incorporate consistency in all of the computed inter-image homographies, and the algorithm to solve it. This means that all of the common parameters extracted from these homographies are the same. These parameters are related to pose estimation.

Inter-image homographies are transfer functions between two images and are induced by planes. The first epipole and all the intersections between any two planar surfaces can be used to compose these transfer functions. Camera calibration and scene reconstruction tasks are solved using these functions. Segmentation is achieved through the intersections of the planar surfaces in the scene.

1.4 Related work

The camera intrinsic parameters are needed to solve localization and reconstruction problems. The earliest works dealing with camera calibration make use of objects or models in the scene with well-known geometry. Tsai in [22] presents a method for camera calibration where an expert user gives the 3D-2D correspondences between the object in the scene and the image plane. No constraints are imposed in the geometry of the observed object. In [28] Zhang presents another model-based method for camera calibration using a planar object, usually a chessboard. [28] addresses the camera calibration problem as a non-linear formulation.

Other works such as [5] and [25] make use of camera motion with some constraints in order to recover internal and localization parameters but without considering known objects in the scene. In [5] the authors use views generated by a rotating camera and describe the relation between motion and the image of the absolute conic. In [25] the problem of camera calibration is analyzed assuming that the three dimensional scene is conformed by planar structures. Both of last methods are linear at first stage and require an additional bundle adjustment step. [5] and [25] present the non-linear formulation used to improve the original linear estimation.

Motion recovery and scene reconstruction are strongly linked tasks, because the first induces the 3D global frame of reference that is used to describe the elements in the scene. One of the first works that dealt with this problem was written by Faugeras and Lustman [6]; in this work they use only a planar surface and two views. Their approach makes use of features extracted from the two views. These features are used to fit a transfer function (inter-image homography) by means of a least-squares method. The main contribution of [6] is the linear method used for motion and structure recovery and the inclusion of geometrical and algebraic constraints that are induced by rigidity and planarity in the scene. With their approach, they do not compute the fundamental matrix associated to the views, but instead, they solve the problem of locating the camera by using one plane. The authors make an analysis of the singular values of the transfer function, showing that the problem has multiple solutions: eight in the general case, that can be reduced to two by using more information from the scene, four solutions when two of the singular values are equal and an indetermination when the three singular values are equal. With an additional observed plane, all ambiguities are dismissed.

For multi-planar scenes Bartoli et al. [2], introduce the motion and structure recovery problem as a non-linear cost function with explicit epipolar geometry computation, leading to the associated instability problems. Robust estimators are used for the minimization problem. [2] extends the epipolar geometry, and their authors analyze the numerical stability of the algorithm.

Lopez-Nicolas et al. [7], address the motion recovery problem in multi-planar scenes, introducing a linear algorithm that computes the fundamental and essential matrices. Pose recovering is carried out by means of inter-image homographies and the essential matrix. Planar homologies used in that work were applied by Malis and Cipolla in [11], for automatic calibration from multi-planar structures. The relation between homologies and inter-image homographies induced by planar homographies is provided by [11].

Segmentation of multiplanar scenes using the geometrical and algebraic constraints as applied in the previous works were used by Vigueras and Rivera [23]. They propose a non-linear cost function that when minimized gives the maximum likelihood of the inter-image homographies and the first epipole without explicit computation of the fundamental matrix. Their approach imposes that all the transfer functions have the same localization parameters ensuring projective coherence, a similar idea as in [2] but avoiding instability due to the explicit epipolar geometry computation.

In [17] Simon et al., present a user-assisted system for tracking several planes in a marker-less scene. In this work, a user delimits the boundary of each projected planar surface. Another contribution of [17] is the DLT-like[3] algorithm for computing inter-image homographies. In [16] Simon et al. show how to recover the localization and reconstruction parameters of multi-planar scenes with minimal user assistance, which consists of manual selection of a base plane. Additional

[3] DLT stands for *Direct Linear Transformation*.

planes are considered as *walls*, i.e. these additional planes are orthogonal to the base plane. In [16], all of the computed parameters are improved by means of an ad-hoc Hough transform applied to a video sequence.

1.5 Overview of the proposed method

When only two views of the same multi-planar scene are considered, epipolar geometry arises as a natural mathematical tool in order to determine motion parameters of the cameras [26]. In this work, we avoid explicit epipolar geometry computation and, instead, we use planar homologies, which are the product of two inter-image homographies. Homographies are calculated by using the RANSAC paradigm [10] to fit the linear model proposed in [17] on putative matches computed with the Zero-mean Normalized Cross-Correlation (ZNCC) measure [19].

We use intersections between planes [16] in order to improve the support of each homography obtained by RANSAC. The intersection of the planes is reflected in the first view as a line and is codified inside of the planar homology associated to these planes. The first epipole is also codified in this homology. The SVD Algorithm [14] is used for the extraction of these vectors and is applied on a stack of planar homologies.

Once the epipole and all intersections between planes are computed, the homography with the largest support is chosen as the reference homography, then an iterative linear method that computes an improved version of these vectors is triggered. All the non-reference homographies are rewritten using the reference homography, the first epipole and the vector that describes the intersection between the reference plane and the current plane. This stage is at the core of our method and ensures projective coherence between all the homographies.

With the epipole and the reference homography, both focal distances are estimated using the algorithm described in [25]. Once focal distances are determined, localization and reconstruction stages are carried out. Localization of the first camera is fixed at the origin of $\mathbb{R}^3$ with null orientation. For the second one we use a Faugeras-like algorithm for orientation recovery and later we use all the information computed up to now in order to recover the translation vector.

Dense segmentation of the projection of each plane is carried out with the improved version of the intersections between planes and the associated homography.

Chapter 2

Mathematical Foundations

This chapter presents the mathematical notation and formulations that are used in our work. Our research deals with a linear method for camera calibration and structure recovery, so that we should have a full understanding of numerical linear algebra.

In this work, the cross product operation ($\cdot \times \cdot$) is often represented as a matrix operator [10]. If vector $\mathbf{a}$ is defined as $(a_1, a_2, a_3)^T$ then $\mathbf{a} \times \cdot$ is written as $[\mathbf{a}]_\times \cdot$. The symbol $[\mathbf{a}]_\times$ is as follows:

$$[\mathbf{a}]_\times \equiv \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix}. \tag{2.1}$$

2.1 Camera Model

The pinhole camera model [10] is used in this work. The elements of this camera model can be observed in Figure 2.1 and they are:

- The camera center or center of projection, denoted by $\mathbf{C}$. Without loss of generality the coordinates of $\mathbf{C}$ for the first camera are at the origin $(0, 0, 0)^T$.

- The image plane with equation $Z = f \in \mathbb{R}^+$, is the place where points are projected.

- The line from the camera center perpendicular to the image plane is called the principal axis.
- The point $\mathbf{p}$ where the principal axis intersects the image plane is called the principal point.

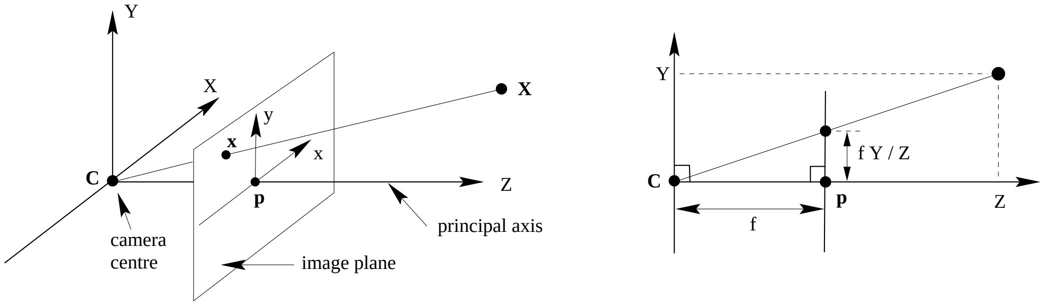

Figure 2.1: Pinhole camera geometry. $\mathbf{C}$ is the camera center and $\mathbf{p}$ is the principal point. The focal distance is denoted by f. The image plane is placed in front of the camera center.

The space point $\mathbf{X}$ is projected to the image plane at coordinates $\mathbf{x} = (f\frac{X}{Z}, f\frac{Y}{Z})^T$. This projection is determined by the intersection between the image plane and the line joining $\mathbf{X}$ and the camera center $\mathbf{C}$.

Image points can be expressed using normalized coordinates [26], this means that the principal point is at $(0,0)^T$ and the focal distance f equals to one. When the image coordinates are non-normalized, we need to incorporate the camera matrix $\mathbf{K}$ defined as:

$$\mathbf{K} = \begin{pmatrix} f & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix},$$

where f is the focal distance. In this work we are considering square pixels, this means that $f_y = f$, otherwise $f_y = \tau f$, where τ is a known aspect ratio. $(u_0, v_0)^T = (0,0)^T$ are the coordinates of the principal point, fixed at the center of the image plane.

In order to define matrix operations between points in the 3D space or in the image plane with common mathematical objects in their native spaces[1], we incorporate homogeneous coordinates. This means that all the points (in space or image coordinates) are represented by adding an entry different from zero, i.e. a space point $\mathbf{X} = (x, y, z)^T$ is represented in homogeneous coordinates by $\tilde{\mathbf{X}} = (X, Y, Z, W)^T$ with $W \neq 0$. The same applies to image coordinates, where $\mathbf{x} = (x, y)^T$ is represented by $\tilde{\mathbf{x}} = (X, Y, Z)^T$ where $Z \neq 0$.

One fact that is important to know when the pinhole camera model is used, is the equivalence class induced by the ray from $\mathbf{C}$ through $\mathbf{X}$. All the space points that lie on this ray are projected to the same point at the image plane. A 3D space point $\mathbf{X} = (X, Y, Z)^T$ is projected by the camera to a 2D point $\mathbf{x} = (fX/Z, fY/Z)^T$. This projection is done via a ray that starts at $\mathbf{C}$ through the space point $\mathbf{X}$. An infinity of points lie on this ray including one that is projected to the image plane. This ray is a sub-set of the line $\overline{\mathbf{CX}}$. All the points on this ray define an equivalence class whose representative element is $\tilde{\mathbf{x}} = (f\frac{X}{Z}, f\frac{Y}{Z}, f)^T$. Due to the equivalence class, in this work, $\mathbf{x} = (x, y, 1)^T$ is considered as the representative element[2] of the equivalence class defined by the projection of $\mathbf{X}$ at the image plane $\mathbf{Z} = f$.

2.2 Key Points

Given any two images of the same multi-planar scene, we are interested in knowing how to *align* the first to the second image. In other words, we wish to find the transfer function that allows us to describe matching points from the first to the second image. If these transfer functions are constrained to reflect rigidity and planarity properties induced by the planes in the scene, we will need as many transfer functions as there are planar surfaces are in the scene.

Not all the points in the scene that are coplanar will be considered to describe the transfer functions. We use *key points* also known as features or corners in the literature. These are picture elements that are representative under some criteria defined by the feature detector method, e. g. the Harris corner and edge detector [9], MIC [21] or SUSAN [18], among others.

[1] i.e. planes in $\mathbb{R}^3$, lines in $\mathbb{R}^3$ or $\mathbb{R}^2$ and so on.

[2] We write $\mathbf{x}$ instead of $\tilde{\mathbf{x}}$.

In this work, the method developed by Harris and Stephens is used to detect key points. This method makes use of the structure tensor $\mathbf{M}$:

$$\mathbf{M}(x,y) = \mathbf{g}_m(\sigma) \otimes \begin{pmatrix} \mathbf{I}_u^2(x,y) & \mathbf{I}_u\mathbf{I}_v(x,y) \\ \mathbf{I}_u(x,y)\mathbf{I}_v(x,y) & \mathbf{I}_v^2(x,y) \end{pmatrix}$$

where $\mathbf{g}_m(\sigma)$ is a Gaussian $m \times m$ mask with standard deviation σ, $\otimes$ is the convolution operator. The mask $\mathbf{g}_m(\sigma)$ acts as a weighting smoothing correlator. $\mathbf{I}(x,y)$ is the image at coordinates (x,y). $\mathbf{I}_u$ and $\mathbf{I}_v$ are the partial derivatives of $\mathbf{I}$ at pixel (x,y) in directions u and v respectively.

Derivatives in $\mathbf{M}$ can be calculated using Gaussian kernels or Sobel masks, among others [8]. The quality of the key points strongly depend on the method used to compute the derivatives and on the kernel size of the Gaussian mask (the value of σ in $\mathbf{g}_m$), as well as the value of m.

Cornerness measures the quality of a pixel to be considered as a key point or not. This classification is carried out by means of the structure tensor $\mathbf{M}$. An arbitrary and widely accepted cornerness measure was defined in [9]:

$$H(x,y) = det(\mathbf{M}(x,y)) - k \cdot trace(\mathbf{M}(x,y))^2,$$

where k is a tunable parameter (usually $k = 0.04$) but it depends on the parameters which were used to compute $\mathbf{M}$. Another measure that experimentally brings better results than the one presented above is:

$$N(x,y) = \frac{det(\mathbf{M}(x,y))}{trace(\mathbf{M}(x,y))^2}.$$

$N(x,y)$ may be undefined when no texture is present in the image, i.e. $trace(\mathbf{M}(x,y)) \rightarrow 0$, in such a case (x,y) is not key point. The cornerness measure N is reported by Noble in [12]. One advantage of this measure compared with H is that Noble's does not need the k parameter.

2.3 Matching key points

Once key points are computed, the next step consists in matching these points. In this work, points are matched using a modified version of the ZNCC correlation measure [19]:

$$ZNCC = \frac{\mathbf{N} \cdot \mathbf{S}_{IJ} - \mathbf{S}_I \mathbf{S}_J}{\sqrt{(\mathbf{N} \cdot \mathbf{S}_{II} - \mathbf{S}_I^2)(\mathbf{N} \cdot \mathbf{S}_{JJ} - \mathbf{S}_J^2)}}, \tag{2.2}$$

the elements in this formula are as follows:

- $\mathbf{W}$ is the window defining the patches, i.e. a rectangular region containing a certain number of pixels and centered at the reference feature,
- $\mathbf{N}$ is the number of pixels in the window $\mathbf{W}$,
- $\mathbf{I}$ is the first view,
- $\mathbf{J}$ is the second view,
- $\mathbf{S}_I = \sum_W \mathbf{I}$, where $\sum_W \mathbf{I} \equiv \sum_{(u,v) \in W} \mathbf{I}(u,v)$,
- $\mathbf{S}_J = \sum_W \mathbf{J}$,
- $\mathbf{S}_{II} = \sum_W \mathbf{I}^2$, where $\sum_W \mathbf{I}^2 \equiv \sum_{(u,v) \in W} \mathbf{I}(u,v)\mathbf{I}(u,v)$,
- $\mathbf{S}_{JJ} = \sum_W \mathbf{J}^2$,
- $\mathbf{S}_{IJ} = \sum_{W,W'} \mathbf{IJ}$, where $\sum_{W,W'} \mathbf{IJ} \equiv \sum_{(u,v) \in W_1, (u',v') \in W'} \mathbf{I}(u,v)\mathbf{J}(u',v')$,

The ZNCC measure takes values between $[-1, 1]$ and 1 indicates a perfect match. For real-time response, this measure has reported the best results when comparing with other similar correlation measures [19].

In this work, correspondences between two views are denoted by:

$$\{\mathbf{x} \leftrightarrow \mathbf{x}'\},$$

where $\mathbf{x} = (x, y, z)^T$ is a three dimensional vector expressed in homogeneous coordinates, representing a detected feature in the first view. $\mathbf{x}'$ is used for the second view.

In the literature, several other methods for key point matching can be found [4]. Some of them use linear methods with a final non-linear step as refinement. Others deal with the matching problem as a finite-combinatorial one.

2.4 Epipolar Geometry

Epipolar geometry is always well defined between two views that project the same physical scene [26] because it depends only on the motion between them. Epipolar geometry is depicted in Figure 2.2. The elements of this geometry are:

- The epipole $\mathbf{e}$ is the point of intersection of the line joining the camera centers $\mathbf{C}$ and $\mathbf{C}'$ with the image plane on the first view. For the second image the epipole is denoted by $\mathbf{e}'$.
- An epipolar plane containing the epipoles and the projected point $\mathbf{X}$. There is a one-parameter family of epipolar planes when the projected point moves.
- An epipolar line $\mathbf{l}$ is the intersection of an epipolar plane with the image plane. All epipolar lines intersect at the associated epipole.

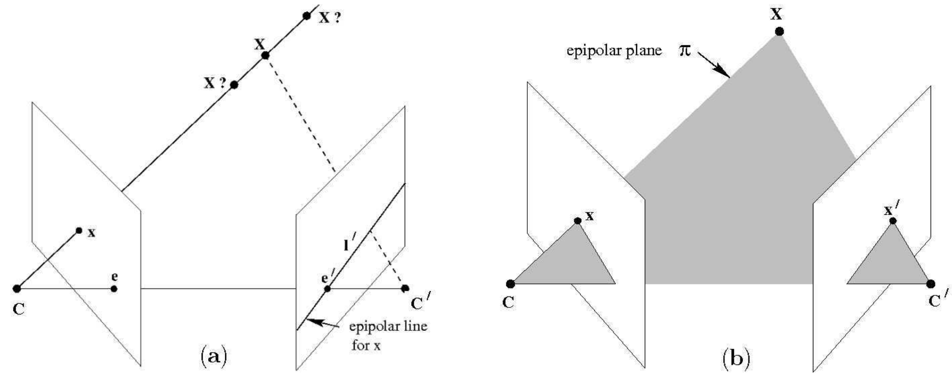

Figure 2.2: Geometry induced by point correspondences. (a) Camera centers are represented by $\mathbf{C}$ and $\mathbf{C}'$, $\mathbf{X}$ is a 3d-space point and is projected on the views as $\mathbf{x}$ and $\mathbf{x}'$ respectively. (b) The epipoles $\mathbf{e}$ and $\mathbf{e}'$ lie in the intersection of the baseline (line between $\mathbf{C}$ and $\mathbf{C}'$).

The fundamental matrix $\mathbf{F}$ is an algebraic representation of the epipolar geometry. For any pair of correspondences $\mathbf{x} \leftrightarrow \mathbf{x}'$ in the two images, this matrix satisfies:

$$\mathbf{x}'^T\mathbf{F}\mathbf{x} = 0. \tag{2.3}$$

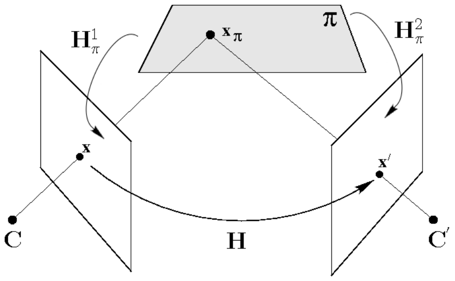

Figure 2.3: Inter-image homography induced by observing a planar surface. $\mathbf{H}$ is the inter-image homography induced by observing a planar surface π in a general position. $\mathbf{H}_\pi^1$ and $\mathbf{H}_\pi^2$ are the transfer functions between π and the first and second views, respectively.

2.5 Inter-image Homographies

A planar homography transfers coplanar space points to the image plane of a given view. This mapping is a 3×3 real matrix that takes a space point and gives us the projection of this point on the desired image plane. A DLT-like algorithm may be used to find this matrix by means of the least-squares method. Inside a noise-free environment, homographies are bijective maps.

Inter-image homographies are transfer functions between two image planes, and are based on planar homographies. Any inter-image homography can be decomposed into the product of two or more planar homographies [17], see Figure 2.3 for a sketch.

Given two planar homographies $\mathbf{H}_\pi^1$, $\mathbf{H}_\pi^2$, the inter-image homography between these views can be written as:

$$\mathbf{H} \equiv \mathbf{H}_1^2 = \mathbf{H}_\pi^2(\mathbf{H}_\pi^1)^{-1},$$

here $\mathbf{H}$ maps points from the first view to the second one, and π represents the observed planar surface.

According to [10], if we know the pose parameters $(\mathbf{R}, \mathbf{t})$ of the second camera, the parameters of the observed planar surface $(\mathbf{v})$ and the camera intrinsic matrices $(\mathbf{K}_1, \mathbf{K}_2)$, it is possible to write the inter-image homography as follows:

$$\mathbf{H} \sim \mathbf{K}_2(\mathbf{R} - \mathbf{t}\mathbf{v}^T)\mathbf{K}_1^{-1}, \tag{2.4}$$

$\mathbf{H}$ transfers the projected points from the observed planar surface on the first view to the second view, this is a 3×3 matrix. $\mathbf{R}$ is a 3×3 matrix, representing the rotation between the first and the second view. $\mathbf{t}$ is the translation vector from the first camera to the second, this vector has dimension three. $\mathbf{n}$ is the three dimensional normal vector associated to the surface such that the plane equation is:

$$\mathbf{n} \cdot \mathbf{X} + d = 0, \tag{2.5}$$

when $\mathbf{X}$ is a point on the planar surface and d a real number. $\mathbf{v} = \frac{\mathbf{n}}{d}$ is a vector used to describe the same plane equation such that:

$$\mathbf{v} \cdot \mathbf{X} + 1 = 0. \tag{2.6}$$

In order to compute an homography from correspondences (at least four non-collinear of them), we solve the next homogeneous linear system [17]:

$$\begin{pmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1x_1 & -x'_1y_1 & -x'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y'_1x_1 & -y'_1y_1 & -y'_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x'_2x_2 & -x'_2y_2 & -x'_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -y'_2x_2 & -y'_2y_2 & -y'_2 \\ & & & & & \vdots & & & \\ x_n & y_n & 1 & 0 & 0 & 0 & -x'_nx_n & -x'_ny_n & -x'_n \\ 0 & 0 & 0 & x_n & y_n & 1 & -y'_nx_n & -y'_ny_n & -y'_n \end{pmatrix} \begin{pmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \\ h_9 \end{pmatrix} = \mathbf{0} \tag{2.7}$$

where the entries h_i are part of the homography $\mathbf{H}$, as follows:

$$\mathbf{H} = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix}, \tag{2.8}$$

and the correspondence pairs are: $\{(x_i, y_i, 1) \leftrightarrow (x'_i, y'_i, 1)\}$ as defined above.

In our work, scenes contain many planar surfaces. It is expected that all the computed inter-image homographies share the same pose parameters [2] and only differ in the ones related to the plane equations.

We use $\mathbf{H}_j \sim \mathbf{K}_2(\mathbf{R} - \mathbf{t}\mathbf{v}_j^T)\mathbf{K}_1^{-1}$ as the inter-image homography which describes the transferring function induced by j-th planar surface and the associated views.

2.6 Consistency between homographies

Two homographies, $\mathbf{H}_1$ and $\mathbf{H}_2$, are used to define the associated planar homology [11] as follows:

$$\mathbf{M}_{12} = \mathbf{H}_1^{-1}\mathbf{H}_2, \tag{2.9}$$

consider Figure 2.4 for a sketch of the mapping $\mathbf{M}_{12}$.

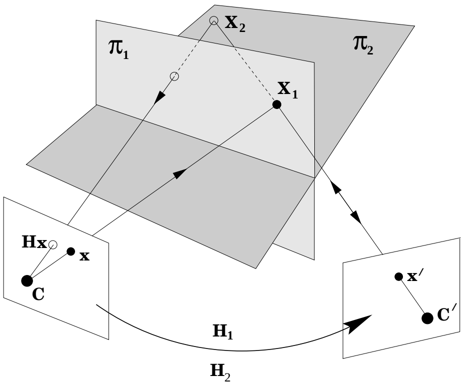

Figure 2.4: The action of the map $\mathbf{M}_{12} = \mathbf{H}_1^{-1}\mathbf{H}_2$ on a point $\mathbf{x}$ in the first view to transfer it to $\mathbf{x}'$ as thought if it were the image of the 3D point $\mathbf{X}_2$. After, map it to the first image as thought if it were the image of the 3D point $\mathbf{X}_1$. Points in the first view which lie on the intersection of the planes are mapped to themselves, so they are fixed points under this action. The epipole $\mathbf{e}$ is also a fixed point under this map.

Therefore, considering any homology $\mathbf{M}_{ij}$ and the relation (2.4), we obtain:

$$\mathbf{M}_{ij} \sim \mathbf{K}_1(\mathbf{R} - \mathbf{t}\mathbf{v}_i^T)^{-1}(\mathbf{R} - \mathbf{t}\mathbf{v}_j^T)\mathbf{K}_1^{-1},$$

using the Sherman-Morrison formula [14] we obtain:

$$\left(\mathbf{R} - \mathbf{t}\mathbf{v}_i^T\right)^{-1} = \mathbf{R}^{-1} + \frac{\mathbf{R}^{-1}\mathbf{t}\mathbf{v}_i^T\mathbf{R}^{-1}}{\alpha},$$

where:

$$\alpha = 1 - \mathbf{v}_i^T \mathbf{R}^{-1} \mathbf{t}.$$

From [26], we know that the first epipole is represented as $\mathbf{e} \sim \mathbf{K}_1 \mathbf{R}^{-1} \mathbf{t}$, using the Sherman-Morrison formula [14] and from that result, we obtain:

$$\mathbf{M}_{ij} \sim \mathbf{I} + \mathbf{e}\mathbf{s}_{ij}^T, \tag{2.10}$$

where

$$\mathbf{s}_{ij} \sim \mathbf{K}_1^{-T} (\mathbf{v}_i - \mathbf{v}_j). \tag{2.11}$$

Consequently, every homography can be written as: $\mathbf{H}_j \sim \mathbf{H}_i \mathbf{M}_{ij}$, this means that:

$$\mathbf{H}_j \sim \mathbf{H}_i (\mathbf{I} + \mathbf{e}\mathbf{s}_{ij}^T), \tag{2.12}$$

the equivalence $\sim$ vanishes in (2.12), if the second singular value of the right side is equal to one [27], hence (2.12) becomes an equality.

The reference homography is denoted by $\mathbf{H}_{ref}$, therefore all the remaining homographies are written as:

$$\mathbf{H}_j \sim \mathbf{H}_{ref} (\mathbf{I} + \mathbf{e}\mathbf{s}_{ref,j}^T). \tag{2.13}$$

From this equivalence class, we can realize that in order to maintain consistency between homographies, our setting needs a reference plane which induces a reference homography. In this work we choose the reference plane as the one with the greatest support, i.e. the planar surface with the biggest number of correspondences associated to it.

2.7 Fundamental matrix and inter-image homographies

When the epipolar geometry is induced by observing a planar surface, as shown in Figure 2.5, the inter-image homography $\mathbf{H}$ associated to that plane can be used in order to express the fundamental matrix:

$$\mathbf{F} \sim \mathbf{H}^{-T} [\mathbf{e}]_\times \sim [\mathbf{e}']_\times \mathbf{H}. \tag{2.14}$$

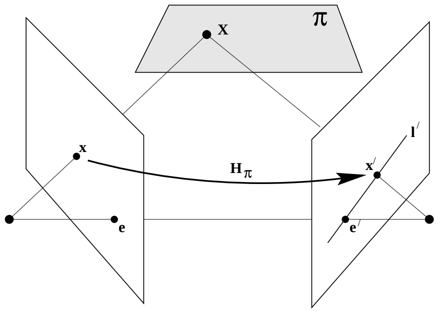

Figure 2.5: Epipolar geometry and inter-image homographies. A point $\mathbf{x}$ in the first image is transferred via the plane π to a matching point $\mathbf{x}'$ in the second image. The epipolar line through $\mathbf{x}'$ is obtained by joining $\mathbf{x}'$ to the epipole $\mathbf{e}'$.

The fundamental matrix has rank two with seven degrees of freedom and the inter-image homography has complete rank with eight degrees of freedom. $\mathbf{F}$ represents a mapping from the two-dimensional projective plane $\mathbb{P}^2$ of the first image to the pencil of epipolar lines through the epipole $\mathbf{e}'$. Thus, it represents a mapping from a two-dimensional onto a one-dimensional projective space[3], and hence, must have rank two.

Nevertheless, as it has been reported in [10], there exist instability problems when trying to explicitly compute the fundamental matrix from planar surfaces. In [25] the authors report that the fundamental matrix is numerically more stable when it is used for camera calibration in conjunction with inter-image homographies (equation 2.14) than when using point correspondences from planar surfaces.

The epipolar geometry is always well defined between any two views of the same three-dimensional scene. If the two camera centers are not coincident, it is uniquely defined. The main problem that arises when trying to determine the epipolar geometry via $\mathbf{F}$ is due to some camera configurations that do not allow to compute it from point correspondences. An important degeneracy is when all the points lie in a plane. Given any pair of correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ and the inter-image homography $\mathbf{H}$, such that $\mathbf{x}'_i \sim \mathbf{H}\mathbf{x}_i$, the fundamental matrix corresponding to the pair of cameras always satisfies the equation (2.3): $\mathbf{x}_i'^T\mathbf{F}\mathbf{x}_i = 0$,

[3] The projective plane $\mathbb{P}^2$ may be thought as $\mathbb{R}^2$ plus an *homogeneous dimension*, i.e. $\mathbb{P}^2$ is the set of all the homogeneous vectors of three entries. The same idea applies to the projective space $\mathbb{P}^n$ for all $n \in \mathbb{N}$.

i.e.:

$$\mathbf{x}_i'^T(\mathbf{F}\mathbf{H}^{-1})\mathbf{x}_i' = 0,$$

from relation (2.14) we know that $\mathbf{F} \sim [\mathbf{e}']_\times \mathbf{H}$ where $[\mathbf{e}']_\times$ is an asymmetric or skew-symmetric matrix as defined at the beginning of this chapter. This means that $\mathbf{F}\mathbf{H}^{-1} \sim [\mathbf{e}']_\times$ and the solution for $\mathbf{F}$ is any matrix of the form $\mathbf{S}\mathbf{H}$, where $\mathbf{S}$ is 3×3 skew-symmetric matrix.

If $\mathbf{F}$ is computed via $\mathbf{H}$, as mentioned above, $\mathbf{F}$ will have three degrees of freedom due to $\mathbf{S}$, and that is true because $\mathbf{S}$ could be written through any three-dimensional vector with three degrees of freedom. The epipole $\mathbf{e}'$ is a three-dimensional vector in homogeneous coordinates, and hence, the fundamental matrix will have two degrees of freedom when is computed via a homogeneous vector such as the epipole, i.e. if $\mathbf{S}$ is written in the form $\mathbf{S} = \lambda[\mathbf{e}']_\times$, where $\lambda \in \mathbb{R} - \{0\}$.

Furthermore, the two epipoles satisfy the next equations:

$$\begin{aligned} \mathbf{F}\mathbf{e} &= 0, \\ \mathbf{F}^T\mathbf{e}' &= 0, \end{aligned}$$

If we know the first epipole $\mathbf{e}$ and one inter-image homography $\mathbf{H}$, the second epipole $\mathbf{e}'$ is determined by:

$$\mathbf{e}' \sim \mathbf{H}\mathbf{e}. \tag{2.15}$$

2.8 Intersections between Planes

The intersection between planes [16] i and j implies that a given point $\mathbf{x}$ belonging to this intersection at the first image, satisfies both $\mathbf{x}' \sim \mathbf{H}_i\mathbf{x}$ and $\mathbf{x}' \sim \mathbf{H}_j\mathbf{x}$. Therefore $\mathbf{H}_i\mathbf{x} \sim \mathbf{H}_j\mathbf{x}$, i.e. $\mathbf{x} \sim \mathbf{H}_i^{-1}\mathbf{H}_j\mathbf{x}$, it follows from (2.9) that $\mathbf{x}$ is an eigenvector of $\mathbf{M}_{ij}$.

The eigenvectors of $\mathbf{M}_{ij}$ are (see Figure 2.4):

- The epipole $\mathbf{e}$, which is common for all the elements in the scene. From the definition of $\mathbf{M}_{ij}$, it follows that $(\mathbf{I} + \mathbf{e}\mathbf{s}_{ij}^T)\mathbf{e} = \alpha\mathbf{e}$, where $\alpha = (1 + \mathbf{s}_{ij}^T\mathbf{e})$ is a real number.

- Any vector $\mathbf{a}$ orthogonal to $\mathbf{s}_{ij}$: $(\mathbf{I} + \mathbf{e}\mathbf{s}_{ij}^T)\mathbf{a} = \mathbf{a} + \mathbf{s}_{ij}^T\mathbf{a}\mathbf{e}$, as $\mathbf{s}_{ij}^T\mathbf{a} = 0$ then $(\mathbf{I} + \mathbf{e}\mathbf{s}_{ij}^T)\mathbf{a} = \mathbf{a}$.

The intersection between planes i and j is the set of all such vectors $\mathbf{a}$. Furthermore, by orthogonality we know that $\mathbf{s}_{ij} \cdot \mathbf{a} \equiv \mathbf{s}_{ij}^T\mathbf{a} = 0$, then the equation of this intersection is a straight line given by vector $\mathbf{s}_{ij}$. Furthermore, vector $\mathbf{s}_{ij}$ is part of the first image plane [23]. This fact can be verified by means of equation (2.6) for planes i and j: $\mathbf{v}_i \cdot \mathbf{X} = -1$, $\mathbf{v}_j \cdot \mathbf{X} = -1$ where $\mathbf{X} \in \pi_i \cap \pi_j$. Thus:

$$\begin{aligned} \mathbf{v}_i \cdot \mathbf{X} - \mathbf{v}_j \cdot \mathbf{X} &= 0 \\ (\mathbf{v}_i - \mathbf{v}_j) \cdot \mathbf{X} &= 0, \end{aligned}$$

remember $\mathbf{s}_{ij} \sim \mathbf{K}_1^{-T}(\mathbf{v}_i - \mathbf{v}_j)$. Applying $\mathbf{s}_{ij}$ to last equation we obtain:

$$\mathbf{s}_{ij} \cdot \mathbf{X} = 0.$$

We know from Section 2.1 that if $\mathbf{x}$ is the projection of the space point $\mathbf{X}$, then $\mathbf{x}$ is an element of the equivalence class defined by the ray that starts at $\mathbf{C}$ through $\mathbf{X}$. Thus $\mathbf{x} \sim \mathbf{X}$ and there exists a scalar $\lambda \neq 0$ such that $\mathbf{x} = \lambda\mathbf{X}$, i.e.:

$$\begin{aligned} \mathbf{s}_{ij} \cdot (\lambda\mathbf{x}) &= 0 \\ \mathbf{s}_{ij} \cdot \mathbf{x} &= 0. \end{aligned}$$

This means that $\mathbf{s}_{ij}$ is a vector orthogonal to all the points $\mathbf{x}$ whose space point $\mathbf{X}$ lies in $\pi_i \cap \pi_j$. Points $\mathbf{x}$ are in the first image and $\mathbf{s}_{ij}$ is a vector normal to $\mathbf{X} \in \pi_i \cap \pi_j$. This intersection is a straight line when planes are in general position, otherwise it is a point or an empty set (a degenerated line). The same procedure does not apply to the second view and projection $\mathbf{x}'$ of $\mathbf{X}$ due to the elements of the rigid transformation between $\mathbf{C}$ and $\mathbf{C}'$, even so $\mathbf{s}_{ij}$ may be drawn in the second view applying an inter-image function.

The intersection between the reference plane and the j-th planar surface, observed at the first image, is given by $\mathbf{s}_{ref,j} \cdot \mathbf{x} = 0$. The homology between any other planes i and j is given by $\mathbf{M}_{ij} = \mathbf{M}_{ref,i}^{-1}\mathbf{M}_{ref,j}$, and by using (2.10) we obtain:

$$\mathbf{M}_{ij} = (\mathbf{I} + \mathbf{e}\mathbf{s}_{ref,i}^T)^{-1}(\mathbf{I} + \mathbf{e}\mathbf{s}_{ref,j}^T) = \mathbf{I} + \frac{\mathbf{e}(\mathbf{s}_{ref,j} - \mathbf{s}_{ref,i})^T}{1 + \mathbf{s}_{ref,i}^T\mathbf{e}},$$

therefore, the intersection line is defined by the equation: $(\mathbf{s}_{ref,j} - \mathbf{s}_{ref,i}) \cdot \mathbf{x} = 0$. This means that:

$$\mathbf{s}_{ij} = \mathbf{s}_{ref,j} - \mathbf{s}_{ref,i}.$$

2.9 The Faugeras-Lustman Algorithm

This section deals with a well known method for motion and structure recovery using one planar surface and two views: the Faugeras-Lustman Algorithm [6]. This method uses inter-image homographies and assumes that image coordinates are normalized (see Section 2.1). If the transfer functions are computed with non-normalized coordinates, we should provide the internal parameters of both cameras in order to normalize the inter-image homographies (see Section 2.1).

This algorithm obtains localization or motion parameters as well as reconstruction parameters by means of an analysis of the singular values of the inter-image homography. Singular values are computed through the SVD Algorithm [14].

From equation (2.4), we know that an inter-image homography $\mathbf{H}$, induced by a planar surface with normal vector $\mathbf{n}$, can be written as:

$$\mathbf{H} \sim \mathbf{K}_2(\mathbf{R} - \mathbf{t}\frac{\mathbf{n}^T}{d})\mathbf{K}_1^{-1}$$

in order to apply the Faugeras-Lustman Algorithm we only use the collineation matrix $\mathbf{C} = \mathbf{R} - \mathbf{t}\frac{\mathbf{n}^T}{d}$, where $\mathbf{R}$, $\mathbf{t}$ represent the relative motion parameters between the first and the second camera, and d is the distance between the optical center of the first camera and the observed plane in the space. $\mathbf{C}$ can be decomposed by the SVD algorithm as: $\mathbf{C} = \mathbf{U}\mathbf{S}\mathbf{V}^T$. The elements of $\mathbf{S}$ are the square roots of the eigenvalues of $\mathbf{C}\mathbf{C}^T$. These eigenvalues λ_i are positive and can be sorted in decreasing order: $\lambda_1 \geq \lambda_2 \geq \lambda_3$. These values are used to recover $\mathbf{R}$, $\mathbf{t}$, $\mathbf{n}$ and d. Introducing auxiliary variables we obtain:

$$
\begin{aligned}
\mathbf{R} &= s\mathbf{U}\mathbf{R}'\mathbf{V}^T \\
\mathbf{t} &= \mathbf{U}\mathbf{t}' \\
\mathbf{n} &= \mathbf{V}\mathbf{n}' \\
d &= sd' \\
s &= det(\mathbf{U})det(\mathbf{V}),
\end{aligned}
$$

where $d' = \pm\lambda_2$.

We have no knowledge about the sign of d', thus several solutions are possible: eight when all the singular values are distinct, four when two singular values are equal and an indetermination when the three are equal.

Once we know $\mathbf{C}$, we compute $\mathbf{U}$, $\mathbf{S}$ and $\mathbf{V}$ by means of the SVD Algorithm. The new unknown parameters are entries of matrices $\mathbf{R}'$, $\mathbf{t}'$, $\mathbf{n}'$. Further, d' and s have also to be computed. Writing $\mathbf{n}' = (a_1, a_2, a_3)^T$ where a_i are as follows:

$$
\begin{aligned}
a_1 &= \pm\sqrt{\frac{\lambda_1^2 - \lambda_2^2}{\lambda_1^2 - \lambda_3^2}} \\
a_2 &= 0 \\
a_3 &= \pm\sqrt{\frac{\lambda_2^2 - \lambda_3^2}{\lambda_1^2 - \lambda_3^2}}.
\end{aligned}
$$

So far, we can not compute $\mathbf{n}$ from the a_i's. If we know that a point $\mathbf{X}$ is projected by the first camera as $\mathbf{x} = (x, y, 1)^T$, the entries of $\mathbf{n}$ should satisfy the following inequality:

$$\frac{\mathbf{n}^T\mathbf{x}}{d} > 0,$$

and d holds:

$$\frac{h_7x + h_8y + h_9}{d} > 0.$$

These inequalities are the main result of Proposition 4 in [6] and are used to reduce the number of solutions. The knowledge of the point $\mathbf{x}$ allows to compute the sign of d', therefore d is determined. Instead of using $\mathbf{X}$ in these inequalities, we use $\mathbf{x}$ because this point is an element of the equivalence class generated by the ray that begins at the optical center through $\mathbf{X}$.

Up to now $\mathbf{n}'$ and d' are already known, this implies that it is possible to determine $\mathbf{n}$ and d. It is time to analyze the singular values of the collineation $\mathbf{C}$. When $\lambda_1 > \lambda_2 > \lambda_3$ and $d' > 0$ we obtain:

$$\mathbf{R}' = \begin{pmatrix} cos(\theta) & 0 & -sin(\theta) \\ 0 & 1 & 0 \\ sin(\theta) & 0 & cos(\theta) \end{pmatrix}$$

where:

$$\begin{aligned} sin(\theta) &= \frac{(\lambda_1 - \lambda_3)a_1a_3}{\lambda_2} \\ cos(\theta) &= \frac{\lambda_1 a_3^2 + \lambda_3 a_1^2}{\lambda_2} \end{aligned}$$

and $\mathbf{t}' = (\lambda_3 - \lambda_1)(a_1, 0, -a_3)^T$. When $\lambda_1 > \lambda_2 > \lambda_3$ and $d' < 0$ we obtain:

$$\mathbf{R}' = \begin{pmatrix} cos(\theta) & 0 & sin(\theta) \\ 0 & -1 & 0 \\ sin(\theta) & 0 & -cos(\theta) \end{pmatrix}$$

where:

$$\begin{aligned} sin(\theta) &= \frac{(\lambda_1 + \lambda_3)a_1a_3}{\lambda_2} \\ cos(\theta) &= \frac{\lambda_3 a_1^2 - \lambda_1 a_3^2}{\lambda_2}. \end{aligned}$$

The translation vector $\mathbf{t}'$ is equal to: $-(\lambda_1 + \lambda_3)(a_1, 0, a_3)^T$.

2.9.1 Degenerate configurations

Faugeras and Lustman in [6] describe some camera motions that generate configurations classified as *degenerations*, mainly due to the mathematical structures used in that work. Degenerate configurations are present when there are singular values λ_i with an order of multiplicity greater than 1. These degenerate cases are as follows:

- Double singular value: the camera displacement is normal to the reference plane, in front of the physical reference plane structure.
- Triple singular value: the camera displacement is normal to the reference plane. The second camera is located behind of the physical reference plane structure.
- Triple singular value: caused by a null displacement of the camera. In this case, we get a *pure rotation* displacement.
- Numerical errors arising from calculations performed by the corner detector, point matching algorithm or homography fitting stage that can lead to any of the previous cases, see [10, 14].

2.10 The Xu Algorithm

In this section, we describe an algorithm that allows us to determine the focal distances for both cameras using two inter-image homographies. In [25], Xu et al. describe an algorithm to compute camera calibration and structure parameters. Using the theory developed in [25], we are able to compute the focal distances of the two cameras in our setting.

Xu's method assumes that only two inter-image homographies are available and computed by hand: selecting two clusters of correspondence pairs in the two views and later determining the two transfer functions with these pairs. The two transfer functions are needed in order to compute the second epipole $\mathbf{e}'$ by solving a generalized eigenvector problem derived from equation (2.15): $\mathbf{e}' = \mathbf{H}_1\mathbf{e}$. The generalized eigenvector problem is as follows:

$$\mathbf{H}_1\mathbf{e} \sim \mathbf{H}_2\mathbf{e} \sim \mathbf{e}' \tag{2.16}$$

This algorithm follows the same idea from Faugeras and Lustman, i.e. it makes use of relation (2.4): $\mathbf{H} \sim \mathbf{K}_2(\mathbf{R} - \mathbf{t}\mathbf{v}^T)\mathbf{K}_1^{-1}$ to extract both focal distances f_1 and f_2, motion parameters $\mathbf{R}$ and $\mathbf{t}$, and structure parameters $\mathbf{v} = \frac{\mathbf{n}}{d}$. The intrinsic camera matrices are $\mathbf{K}_1 = diag(f_1, f_1, 1)$ and $\mathbf{K}_2 = diag(f_2, f_2, 1)$ respectively. From relation (2.4) we obtain:

$$\mathbf{H}_1 \sim \mathbf{K}_2\mathbf{R}\mathbf{K}_1^{-1} - \mathbf{K}_2\mathbf{t}\mathbf{v}^T\mathbf{K}_1^{-1}.$$

To eliminate the $\sim$ symbol, we use a scalar value $s \in \mathbb{R} - \{0\}$ as follows:

$$s\mathbf{H}_1 = \mathbf{K}_2\mathbf{R}\mathbf{K}_1^{-1} - \mathbf{K}_2\mathbf{t}\mathbf{v}^T\mathbf{K}_1^{-1},$$

from [25] we know that $\mathbf{e}' \sim \mathbf{K}_2\mathbf{t}$ and using a not-null scalar a we obtain $\mathbf{e}' = a\mathbf{K}_2\mathbf{t}$, then:

$$s\mathbf{H}_1\mathbf{K}_1 - \mathbf{e}'\mathbf{m}^T = \mathbf{K}_2\mathbf{R}, \tag{2.17}$$

where $\mathbf{m} = a\mathbf{v}$ and $\mathbf{m} = (m_1, m_2, m_3)^T$. Multiplying each side of the equation by its transpose, we get:

$$(s\mathbf{H}_1\mathbf{K}_1 - \mathbf{e}_2\mathbf{m}^T)(s\mathbf{H}_1\mathbf{K}_1 - \mathbf{e}_2\mathbf{m}^T)^T = (\mathbf{K}_2\mathbf{R})(\mathbf{K}_2\mathbf{R})^T,$$

therefore:

$$(s\mathbf{H}_1\mathbf{K}_1 - \mathbf{e}_2\mathbf{m}^T)(s\mathbf{K}_1^T\mathbf{H}_1^T - \mathbf{m}\mathbf{e}_2^T) = (\mathbf{K}_2\mathbf{R})(\mathbf{R}^T\mathbf{K}_2^T).$$

We know that $\mathbf{K}_1^2 = \mathbf{K}_1\mathbf{K}_1^T = \mathbf{K}_1^T\mathbf{K}_1$, then:

$$s^2\mathbf{H}_1\mathbf{K}_1^2\mathbf{H}_1^T + m\mathbf{e}_2\mathbf{e}_2^T - s(\mathbf{H}_1\mathbf{K}_1\mathbf{m}\mathbf{e}_2^T + \mathbf{e}_2\mathbf{m}^T\mathbf{K}_1\mathbf{H}_1^T) = \mathbf{K}_2^2, \tag{2.18}$$

where $m = ||\mathbf{m}||^2 = \mathbf{m}^T\mathbf{m}$.

Equation (2.16) is linear with respect to a seven-dimensional vector:

$$\mathbf{p} = \begin{pmatrix} s^2 f_1^2 \\ s f_1 m_1 \\ s f_1 m_2 \\ s m_3 \\ m \\ s^2 \\ f_2^2 \end{pmatrix}, \tag{2.19}$$

therefore $\mathbf{L}\mathbf{p} = \mathbf{q}$. Matrix $\mathbf{L}$ is defined as follows:

$$\mathbf{L} = \begin{pmatrix} h_1^2 + h_2^2 & -2e_1h_1 & -2e_1h_2 & -2e_1h_3 & e_1^2 & h_3^2 & -1 \\ h_1h_4 + h_2h_5 & -e_1h_4 - e_2h_1 & -e_1h_5 - e_2h_2 & -e_1h_6 - e_2h_3 & e_1e_2 & h_3h_6 & 0 \\ h_1h_7 + h_2h_8 & -e_1h_7 - e_3h_1 & -e_1h_8 - e_3h_2 & -e_1h_9 - e_3h_3 & e_1e_3 & h_3h_9 & 0 \\ h_4^2 + h_5^2 & -2e_2h_4 & -2e_2h_5 & -2e_2h_6 & e_2^2 & h_6^2 & -1 \\ h_4h_7 + h_5h_8 & -e_2h_7 - e_3h_4 & -e_2h_8 - e_3h_5 & -e_2h_9 - e_3h_6 & e_2e_3 & h_6h_9 & 0 \\ h_7^2 + h_8^2 & -2e_3h_7 & -2e_3h_8 & -2e_3h_9 & e_3^2 & h_9^2 & 0 \end{pmatrix},$$

where h_i are the entries of the inter-image homography $\mathbf{H}_1$, and $\mathbf{q} = (0, 0, 0, 0, 0, 0, 1)^T$. $\mathbf{L}$ and $\mathbf{q}$ are uniquely defined by $\mathbf{H}_1$ and $\mathbf{e}'$.

Due to the non-linearity in the entries of $\mathbf{p}$, it is not possible to obtain a unique solution for $\mathbf{p}$. Therefore, we use the following linear system:

$$\mathbf{p} = \mathbf{L}^+\mathbf{q} + \lambda\mathbf{g}, \tag{2.20}$$

with $\mathbf{p} = (p_i)$, $\mathbf{L}^+ = \mathbf{L}^T(\mathbf{L}\mathbf{L}^T)^{-1}$ and $\mathbf{g} = (g_i)$ is the null-vector of $\mathbf{L}$. Using this linear system and (2.19), we build the following equation:

$$p_2^2 + p_3^2 + \frac{p_1p_4^2}{p_6} = p_1p_5, \tag{2.21}$$

that holds:

$$s^2f_1^2(m_1^2 + m_2^2 + m_3^2) = s^2f_1^2m,$$

remember that $m = m_1^2 + m_2^2 + m_3^2$. Using equation (2.21) we obtain:

$$p_6\left[p_2^2 + p_3^2 - p_1p_5\right] + p_1p_4^2 = 0, \tag{2.22}$$

this is a cubic equation with respect to λ and it is solved for λ using the algorithm described in [14], page 183. Assuming that all the three solutions are real, we obtain λ_1, λ_2 and λ_3 as candidate solutions. These values are used to determine $\mathbf{p}$. In [25], it is reported that only two solutions are considered and the one that is chosen is that which satisfies that $\mathbf{R}$ has a positive determinant in equation (2.17). As it has been established in [25], the remaining solution of the cubic equation is not considered, as it has been experimentally observed that it tends to zero, i.e. the cubic equation degenerates in a quadratic equation.

Chapter 3

Proposed multi-linear approach

Linear methods are important due to the fact that they do not need an initialization guess and are more stable than non-linear methods whose performance strongly depends on the election of the initial approach. Furthermore, some of the non-linear methods require to compute first or second derivatives that can be approximated by several methods [13] and when the model to approximate is highly non-linear, the right choice of the optimization method depends on the experience of the user. On the other hand, the method that we describe in this chapter allows us to face our problem without derivatives, initialization guess or advanced programming skills to ensure convergence [14].

As it has been mentioned in the first chapter, we have developed a novel linear mathematical method in order to determine the camera matrices, the relative localization of the cameras, a scene reconstruction and a dense segmentation from the views. Our method is divided into two stages, the first one computes the epipole $\mathbf{e}$ and all the required vectors $\mathbf{s}_{ref,j}$ from $\mathbf{M}_{ij}$ matrices. The second one is an iterative numerical method that incorporates the support[1] of each homography in order to improve the previous estimation.

The epipole and vectors $\mathbf{s}_{ref,j}$ are used to rewrite all the non-reference homographies. Using the reference homography we compute the focal distances by means of the algorithm developed by Xu et al. Once the camera matrices are computed, localization and reconstruction parameters are calculated using a Faugeras-like algorithm.

[1] Pairs of correspondences associated to a plane.

Our work introduces three linear systems in order to compute the first epipole and the family of vectors $\mathbf{s}_{ref,j}$. The first linear system is used to extract the epipole and the intersection between planes. The last two are formed by the pairs of correspondences in such a way that the previous estimations of the epipole and vectors $\mathbf{s}_{ref,j}$ are improved.

In the literature, two non-linear formulations are used to solve the localization and reconstruction problems. These are as follows:

- In [2], computation of the fundamental matrix and all the homographies is carried out by means of a non-linear cost function:

  $$E = \sum_{j=1}^{m}\sum_{i=1}^{n}(d^2(\mathbf{x}_i^{'}, \mathbf{H}_j\mathbf{x}_i) + d^2(\mathbf{x}_i, \mathbf{H}_j^{-1}\mathbf{x}_i')),$$

  where π_j stands for the j-th planar surface and $\mathbf{x}_i \leftrightarrow \mathbf{x}_i^{'} \in \pi_j$. In this work, the authors propose a parametrization of the fundamental matrix that depends on the finite or infinite nature of the epipoles. The goal is to minimize E, by incorporating constraints in the homographies induced by the epipolar geometry and by the planar surfaces.

- In [23], another non-linear formulation is presented in order to recover all the homographies from dense intensity image registering, the epipole $\mathbf{e}$ and the vectors $\mathbf{s}_{ref,j}$. This formulation is as follows:

  $$E(\mathbf{H}_{ref}, \mathbf{e}, \mathbf{s}_{ref,1}, \cdots, \mathbf{s}_{ref,n}) = \sum_{k=1}^{n}\sum_{\mathbf{x}} w_k(\mathbf{x})||\mathbf{I}_2(\mathbf{x}_k^{'}) - \mathbf{I}_1(\mathbf{x}_k)||^2,$$

  where $w_k(\mathbf{x}) \in [0, 1]$ acts as a membership variable for $\mathbf{x}$, with respect to the k-th planar surface. $\mathbf{I}_1$ and $\mathbf{I}_2$ are the two views containing the projection of a multi-planar scene. The goal is the minimization of E by means of the Levenberg-Marquardt method.

On the other hand, our work deals with the extraction of the epipole and the vectors $\mathbf{s}_{ref,j}$ from the following linear form[2]:

[2] The origin of this stack of matrices will be explaining below.

$$[\mathbf{es}^T_{ref,1} \quad \mathbf{es}^T_{ref,2} \quad \cdots \quad \mathbf{es}^T_{ref,m}],$$

after recovery of the above parameters, the iterative stage is triggered. In this stage, two additional linear systems are solved. Our goal is to incorporate the support of the planar surfaces in order to extract the epipole and the family of vectors $\mathbf{s}_{ref,j}$ from the relation: $\mathbf{H}_j \sim \mathbf{H}_{ref}(\mathbf{I} + \mathbf{es}^T_{ref,j})$. This is a multi-linear form for the epipole $\mathbf{e}$ and vectors $\mathbf{s}_{ref,j}$, and because of this, two linear systems are presented in order to recover those vectors.

A sketch of our algorithm is depicted in Algorithm 1.

Algorithm 1 Localization, reconstruction and segmentation of multi-planar scenes using two views.

```
Require: Two views of the same multi-planar scene. The RANSAC threshold.
    The maximum number of iterations.
Ensure: Localization, reconstruction and dense support of the planar surfaces.
  Using a corner detector, image features are obtained and matched using the
    ZNCC correlation measure (2.2).
  RANSAC is used to fit the linear system in (2.7).
  The SVD Algorithm is used to recover the epipole and the intersections between
    planes (Section 3.1).
  All the non-reference homographies are rewritten (equation 2.13).
  Call Algorithm 2 for the iterative stage (section 3.2).
  The Faugeras-Lustman Algorithm is used to recover the orientation of the
    second camera (Section 3.4).
  Recovery of translation and reconstruction parameters using all the homographies
    (Section 3.5).
  Dense segmentation is conducted at the first view for each planar surface
    (Section 3.6).
```

3.1 Epipole and intersections between planes from planar homologies

Planar homologies are calculated between any two planes, i. e. these matrices $\mathbf{M}_{i,j}$ are determined for all $i < j \in \{1, \cdots, n\}$, where $n > 1$ is the number of

Algorithm 2 Iterative stage

```
Require: Input parameters: correspondence pairs, vectors $\mathbf{e}$ and $\mathbf{s}'_{ref,j}$, homographies. The maximum number of iterations.
Ensure: Improved estimation of the input parameters.
repeat
    The correspondence pairs are refined using the segmented matches obtained by means of RANSAC and vectors $\mathbf{s}_{ref,j}$.
    The vectors $\mathbf{s}_{ref,j}$ and the first epipole are re-estimated.
    Reference homography is re-estimated using all the matches.
    All the non-reference homographies are composed using the reference homography, the first epipole and the vectors $\mathbf{s}_{ref,j}$.
until maximum number of iterations reached or solution converges.
```

planes in the scene. From (2.10) we know that $\mathbf{M}_{ref,j} = \mathbf{I} + \mathbf{e}\mathbf{s}_{ref,j}^T$, thus every planar homology codifies the localization parameters inside the epipole and the reconstruction data with the vector $\mathbf{s}_{ref,j}$.

In order to obtain the epipole and the vectors $\mathbf{s}_{ref,j}$, we define the matrix $\mathbf{G}_{ref,j}$ as $\mathbf{G}_{ref,j} = \mathbf{M}_{ref,j} - \mathbf{I}$, this matrix has rank equal to one. We use the following useful facts:

$$\mathbf{G}_{ref,j}\mathbf{G}_{ref,j}^T = \alpha^2\mathbf{e}\mathbf{e}^T$$

$$\mathbf{G}_{ref,j}^T\mathbf{G}_{ref,j} = \beta^2\mathbf{s}_{ref,j}\mathbf{s}_{ref,j}^T,$$

where $\alpha^2 = ||\mathbf{s}_{ref,j}||^2$ and $\beta^2 = ||\mathbf{e}||^2$ are real numbers. From [14] we know that the singular value decomposition of $\mathbf{G}_{ref,j}$ is $\mathbf{USV}^T$, where $\mathbf{S} = diag(\lambda, 0, 0)$ is a diagonal 3×3 matrix, $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices of the same dimension as $\mathbf{S}$. The SVD decomposition of $\mathbf{G}_{ref,j}$ is:

$$\mathbf{G}_{ref,j} = [\mathbf{U}_1\mathbf{U}_2\mathbf{U}_3]\begin{pmatrix}\lambda & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{pmatrix}[\mathbf{V}_1^T\mathbf{V}_2^T\mathbf{V}_3^T],$$

where $\mathbf{U}_k$ and $\mathbf{V}_k$ are the k-th columns of matrices $\mathbf{U}$ and $\mathbf{V}$, respectively.

Under ideal conditions (noise free data) all the above results hold; but when working with real images, $\mathbf{S}$ does not always have unitary rank. In this case we take the first singular value and the remaining are forced to be zero [10, 14]. In fact, the

SVD Algorithm applied to $\mathbf{G}_{ref,j}\mathbf{G}_{ref,j}^T$ gives us $\mathbf{U}\mathbf{S}^2\mathbf{U}^T$, and for $\mathbf{G}_{ref,j}^T\mathbf{G}_{ref,j}$ we obtain $\mathbf{V}\mathbf{S}^2\mathbf{V}^T$. These results are derived from the fact that $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices. Using this trick we recover $\mathbf{e}$ and $\mathbf{s}_{ref,j}$ from the planar homology $\mathbf{G}_{ref,j}$, as follows:

$$\begin{aligned} \mathbf{e} &= \lambda \mathbf{U}_1 \\ \mathbf{s}_{ref,j} &= \lambda \mathbf{V}_1, \end{aligned} \tag{3.1}$$

where λ is the largest element of $\mathbf{S}$.

For all the possible planar homologies that can be computed, we compose the following stack of matrices:

$$\mathbf{G} = [\mathbf{G}_{ref,2}\mathbf{G}_{ref,3} \cdots \mathbf{G}_{ref,n}], \tag{3.2}$$

in this setting, $\mathbf{G}$ has dimension $3 \times 3(n-1)$ and rank equal to one. The singular value decomposition of $\mathbf{G}$ gives us $\mathbf{G} = \mathbf{USV}^T$, so that $\mathbf{U}$ has size $3 \times 3(n-1)$. $\mathbf{S} = diag(\lambda, 0, \cdots, 0)$ and $\mathbf{V}$ both have dimension $3(n-1) \times 3(n-1)$. Without loss of generality, we suppose that the first planar surface is the reference plane.

The epipole is calculated in the same way as above, by applying the SVD Algorithm to $\mathbf{GG}^T$. Matrix $\mathbf{V} = [\mathbf{V}_1 \cdots \mathbf{V}_n]$ is computed applying the SVD Algorithm to $\mathbf{G}^T\mathbf{G} = \mathbf{VS}^2\mathbf{V}^T$. Vectors $\mathbf{s}_{ref,j}$ are stored into $\mathbf{V}_1$ as follows:

$$\mathbf{V}_1 = \frac{1}{\lambda}\begin{pmatrix} \mathbf{s}_{ref,2} \\ \mathbf{s}_{ref,3} \\ \vdots \\ \mathbf{s}_{ref,n} \end{pmatrix}.$$

So far, we have described the first part of our method, in the next lines we will depict the second stage: the iterative step.

3.2 Iterative method incorporating the support of the homographies

This stage uses the support of each homography to improve the epipole and the family of vectors $\mathbf{s}_{ref,j}$. Computation of this support is obtained by means of the RANSAC paradigm and could have some outliers, i. e. pairs that belong to another plane or are not part of the multi-planar scene.

3.2.1 Improving the support of the homographies.

The family of vectors $\mathbf{s}_{ref,j}$ help us to identify and reject some outliers, because these vectors lie in the intersection between planes in the first view. When the whole image only captures planes represented by the computed homographies, all the outliers belong to any other planar surface.

The first step for outliers rejection consists in computing signed distances to lines $\mathbf{s}_{ref,j}$ for the center of mass of the support of each homography. The center of mass is computed once, at the beginning of the iterative stage. Suppose that we are working with the reference plane, then the center of mass tested on a particular $\mathbf{s}_{ref,j}$ give us a not null value if this point does not vanishes in the line $\mathbf{s}_{ref,j}$.

The vector $\mathbf{s}_{ref,j}$ is a straight line, and when the signed distance of the center of mass to this line equation is computed, a real number is obtained. We are only interested in the sign of this number. The sign of this point on the line, is used to evaluate all the support of the reference plane. The points in the first view, which have a different sign than the obtained by the center of mass, are rejected or classified as outliers for this planar surface. This outlier rejection method is very simple but effective in our setting. Figure 3.1 shows the previous idea.

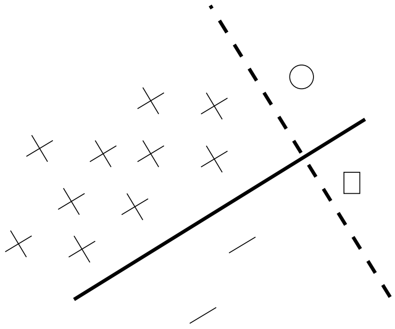

Figure 3.1: Clustering correspondence pairs. Suppose that these straight lines are any pair of vectors $\mathbf{s}_{ref,j}$, then for the support of the reference plane, some points are treated as inliers ($+$ mark) and the remaining points as outliers.

The sign of the center of mass is computed considering all the vectors $\mathbf{s}_{ref,j}$. Each point in their support is evaluated over the complete set of vectors $\mathbf{s}_{ref,j}$. A point is rejected if a sign is different than the one of the center of mass at least once. The outliers are not deleted in the original support. They are only removed in the current iteration of our method.

For each available support, all of their elements are evaluated as mentioned above until convergence. In our setting, five or less iterations were necessary.

Once the support of each planar surface is refined, the next task consists in incorporating this updated support to improve the epipole and the family of plane intersections.

3.2.2 Computing the intersection between planes.

From equation (2.13), we know that each homography can be written as $\mathbf{H}_j \sim \mathbf{H}_{ref}(\mathbf{I} + \mathbf{e}\mathbf{s}^T_{ref,j})$, where $\mathbf{H}_{ref}$ is the reference homography. Considering any planar surface and its associated support, we can write the transfer equation as:

$$\mathbf{x}^{'} \sim \mathbf{H}_{ref}(\mathbf{I} + \mathbf{e}\mathbf{s}^T_{ref,j})\mathbf{x}, \tag{3.3}$$

in order to compute the $\mathbf{s}_{ref,j}$ vector from this equivalence class, we need to do some algebraic operations. Our goal is to transform (3.3) into a design matrix for a linear system. The only allowed operation under the equivalence class operator $\sim$ is the product by any non null scalar factor. Exploiting this idea we do the following algebraic operations:

$$\begin{aligned} \mathbf{x}^{'} &\sim \mathbf{H}_{ref}(\mathbf{I} + \mathbf{e}\mathbf{s}^T_{ref,j})\mathbf{x}, \\ \mathbf{H}^{-1}_{ref}\mathbf{x}^{'} &\sim (\mathbf{I} + \mathbf{e}\mathbf{s}^T_{ref,j})\mathbf{x}, \\ \mathbf{H}^{-1}_{ref}\mathbf{x}^{'} &\sim \mathbf{x} + \mathbf{e}\mathbf{x}^T\mathbf{s}_{ref,j}. \end{aligned}$$

Let $\mathbf{p} = \mathbf{H}^{-1}_{ref}\mathbf{x}^{'}$, and by using the $[\cdot]_\times$ operator as defined in (2.1), it follows:

$$\begin{aligned} [\mathbf{p}]_\times\mathbf{p} &\sim [\mathbf{p}]_\times(\mathbf{x} + \mathbf{e}\mathbf{x}^T\mathbf{s}_{ref,j}), \\ \mathbf{0} &\sim [\mathbf{p}]_\times(\mathbf{x} + \mathbf{e}\mathbf{x}^T\mathbf{s}_{ref,j}). \end{aligned}$$

When dealing with equivalence classes, $\mathbf{a} \sim \mathbf{0}$ implies that $\mathbf{a} = \mathbf{0}$, therefore:

$$([\mathbf{p}]_\times \mathbf{e}\mathbf{x}^T)\mathbf{s}_{ref,j} = -[\mathbf{p}]_\times \mathbf{x}. \tag{3.4}$$

From this equation, we realize that we have formed a linear system $\mathbf{A}\mathbf{s}_{ref,j} = \mathbf{b}$, where $\mathbf{A} = [\mathbf{p}]_\times \mathbf{e}\mathbf{x}^T$ is a 3×3 matrix and $\mathbf{b} = -[\mathbf{p}]_\times \mathbf{x}$ is a column vector with three elements.

This last result holds for any pair $\mathbf{x} \leftrightarrow \mathbf{x}^{'}$. In order to incorporate all the elements in the current support, we form an over determined linear system that can be solved for $\mathbf{s}_{ref,j}$ by means of the SVD Algorithm.

Let $\mathbf{x}_i \leftrightarrow \mathbf{x}_i^{'}$ be any pair in the support, with their associated matrices $\mathbf{A}_i$ and $\mathbf{b}_i$ defined as above, then the over determined linear system is:

$$\begin{pmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_n \end{pmatrix} \mathbf{s}_{ref,j} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_n \end{pmatrix}, \tag{3.5}$$

where n is the number of elements inside the support.

3.2.3 Computing the epipole.

Heretofore, we have depicted a linear system for computing all the $\mathbf{s}_{ref,j}$ vectors, the next step consists in computing the epipole. Remember that the epipole is common for all the elements in the scene.

We use the same equivalence as above:

$$\mathbf{H}_{ref}^{-1}\mathbf{x}^{'} \sim (\mathbf{I} + \mathbf{e}\mathbf{s}_{ref,j}^T)\mathbf{x},$$

which can be simplified as follows:

$$\mathbf{H}_{ref}^{-1}\mathbf{x}^{'} \sim \mathbf{x} + \mathbf{e}(\mathbf{s}_{ref,j}^T\mathbf{x}),$$

if $\mathbf{a}, \mathbf{b}$ are two vectors with real entries, it follows that $\mathbf{a} \cdot \mathbf{b}$ is a real number, therefore defining $\gamma \equiv \mathbf{s}_{ref,j} \cdot \mathbf{x}$ we obtain:

$$\mathbf{H}_{ref}^{-1}\mathbf{x}^{'} \sim \mathbf{x} + \gamma\mathbf{e},$$

following the same previous ideas, we define $\mathbf{p} = \mathbf{H}_{ref}^{-1}\mathbf{x}^{'}$, we rewrite this equivalence as:

$$\mathbf{0} \sim [\mathbf{p}]_{\times}(\mathbf{x} + \gamma\mathbf{e}),$$

that becomes the following equation:

$$-[\mathbf{p}]_{\times}\mathbf{x} = \gamma[\mathbf{p}]_{\times}\mathbf{e}. \tag{3.6}$$

So far we have obtained a linear system for epipole recovery. We extend our mathematical notation. Let

$$\mathbf{A}_i^j = \mathbf{s}_{ref,j} \cdot \mathbf{x}_{i,j}[\mathbf{H}_{ref}^{-1}\mathbf{x}_{i,j}^{'}]_{\times},$$

be a 3×3 matrix, where $\mathbf{x}_{i,j} \leftrightarrow \mathbf{x}_{i,j}^{'}$ is the i-th pair in the support for the j-th planar surface. And considering:

$$\mathbf{b}_i^j = -[\mathbf{H}_{ref}^{-1}\mathbf{x}_{i,j}^{'}]_{\times}\mathbf{x}_{i,j},$$

as a three dimensional column vector, thus the linear system including all the support obtained by RANSAC is as follows:

$$\begin{pmatrix} \mathbf{A}_1^1 \\ \vdots \\ \mathbf{A}_{n_1}^1 \\ \vdots \\ \mathbf{A}_1^m \\ \vdots \\ \mathbf{A}_{n_m}^m \end{pmatrix} \mathbf{e} = \begin{pmatrix} \mathbf{b}_1^1 \\ \vdots \\ \mathbf{b}_{n_1}^1 \\ \vdots \\ \mathbf{b}_1^m \\ \vdots \\ \mathbf{b}_{n_m}^m \end{pmatrix}, \tag{3.7}$$

where m is the total number of planar surfaces in our setting, and n_j is the total number of elements in the j-th support. Each pair $\mathbf{x}_{i,j} \leftrightarrow \mathbf{x}_{i,j}^{'}$ induces a 3×3 matrix $\mathbf{A}_i^j$ and a vector $\mathbf{b}_i^j$ with three entries. As has been mentioned above, equation (3.7) can be solved by means of the SVD Algorithm for vector $\mathbf{e}$.

Heretofore we have developed a novel mathematical tool in order to improve the estimation for the epipole and all the vectors $\mathbf{s}_{ref,j}$ associated to m homographies obtained by means of RANSAC. As commented above, localization parameters are codified by the epipole relation $\mathbf{e} \sim \mathbf{K}_1\mathbf{R}^{-1}\mathbf{t}$ and reconstruction parameters by $\mathbf{s}_{ref,j} \sim \mathbf{K}^{-T}(\mathbf{v}_{ref} - \mathbf{v}_j)$.

3.2.4 Re-estimating homographies.

The reference homography $\mathbf{H}_{ref}$ is computed using all the available support obtained by RANSAC. The associated support that belongs to the reference plane is directly used to compute $\mathbf{H}_{ref}$. For all the remaining planar surfaces, their associated support satisfies relation (3.3), i. e.

$$\mathbf{x}'_{i,j} \sim \mathbf{H}_{ref}(\mathbf{I} + \mathbf{e}\mathbf{s}^T_{ref,j})\mathbf{x}_{i,j}.$$

This relation is used to compute $\mathbf{H}_{ref}$. For each pair $\mathbf{x}_{i,j} \leftrightarrow \mathbf{x}'_{i,j}$, the point $\mathbf{y}'_{i,j}$ is defined as: $\mathbf{y}_{i,j} = (\mathbf{I} + \mathbf{e}\mathbf{s}^T_{ref,j})\mathbf{x}_{i,j}$. The new pair is $\mathbf{x}'_{i,j} \leftrightarrow \mathbf{y}_{i,j}$. This procedure is used for all the pairs that are outside the reference plane.

The linear system (2.7) for computing $\mathbf{H}_{ref}$ is used with the set of correspondences: $\mathbf{x}'_{i,j} \leftrightarrow \mathbf{y}_{i,j}$.

All the remaining homographies should contain the same localization parameters, therefore these homographies are composed by using relation (2.13): $\mathbf{H}_j \sim \mathbf{H}_{ref}(\mathbf{I} + \mathbf{e}\mathbf{s}^T_{ref,j})$.

3.3 Computing focal distances

Once the epipole and the reference homography have been determined by our method, we use the algorithm in [25] by Gang Xu, et al., to recover the focal distances of both cameras. Remember that in Section 2.1 we said that the principal point on each camera is at $(0,0)^T$. With our method, we do not need to cluster the key points by hand. Furthermore, we do not face stability problems associated to the solution of the generalized eigen-vector problem. This procedure is applied by Xu to determine the epipole.

In [24], Vigueras et al. present experimental results associated to calibration tasks in real-time applications. [24] also analyzes the implications of assuming that the principal point is fixed at the center of image plane. This assumption is fundamental in our work.

3.4 Computing the orientation

Given the camera matrices $\mathbf{K}_1$, $\mathbf{K}_2$, we obtain the associated collineation for each homography, i. e. we compute homographies expressed in canonic coordinates. Collineations are denoted by $\mathbf{C}_j$, and are defined as follows:

$$\mathbf{C}_j = \mathbf{K}_2^{-1}\mathbf{H}_j\mathbf{K}_1.$$

The Faugeras-Lustman Algorithm [6] can be applied to the reference homography only, but this requires some knowledge about the camera motion. The theory developed by Faugeras and Lustman shows how to reduce the number of solutions to two when considering only one planar surface, indeed two planes are necessary in order to estimate in an automatic way the orientation of the second camera.

The SVD Algorithm is applied to all the collineations $\mathbf{C}_j$. Analyzing their singular values helps us to reject non-feasible solutions. The last step consists in composing the rotation matrix using only singular values.

In fact, the translation and reconstruction parameters can be obtained by means of the Faugeras-Lustman Algorithm, but in our experiments with real data, these parameters were not always computed with the desired precision. Thus, we propose a new approach in Section 3.5.

3.5 Translation and reconstruction

Using all the computed collineations $\mathbf{C}_j$ and relation (2.4), we can depict:

$$\mathbf{C}_j = \mathbf{R} - \mathbf{t}\mathbf{v}_j^T.$$

Defining $\mathbf{D}_j = \mathbf{R} - \mathbf{C}_j$, i. e. $\mathbf{D}_j = \mathbf{t}\mathbf{v}_j^T$, we wish to extract vectors $\mathbf{t}$ and $\mathbf{v}_j$ from $\mathbf{D}_j$, this goal is achieved by applying the ideas described in Section 3.1. Extraction of $\mathbf{t}$ is the same problem as the epipole recovery and vectors $\mathbf{v}_j$ are computed in the same way as vectors $\mathbf{s}_{ref,j}$.

3.6 Segmentation on the first view

The vectors $\mathbf{s}_{ref,j}$ are used in order to create a partition on the first view. This partition allows us to recover the dense support for all the detected planar surfaces.

This simple strategy gives a well-delimited segmentation, in the cases where only the detected planar surfaces were projected by both cameras. On the other hand, if non-planar surfaces are projected by the cameras, the whole dense support of each detected planar surface is recovered, but this may include points that belong to non-planar structures in the scene. Thus, dense segmentation is only reliable when applied to polyhedral scenes.

Chapter 4

Results

All the datasets used in this work are home-made. In the current literature there is no datasets available to compare our results with previous research. Most benchmark databases are oriented to the problem of stereo views with well aligned epipolar lines. However, these sets of data [15] may be suitable for comparing some of the segmentation problems but not the camera calibration stage that is the basis to compute the other data: camera localization and scene structure. The problem is that rectified epipolar lines imply that the two camera focal axis are concurrent which is a degenerate case for Xu's method (and for all the self-calibration methods actually). Furthermore we do not have a physical reference frame that provide us with reliable data that allows to test the accuracy of our results.

As it has been established in previous sections, our work computes focal distances of both cameras that are used to determine localization and reconstruction parameters. Dense segmentation is carried out directly from the output of the iterative stage (second stage of our method), i.e. dense segmentation makes use of the reference homography and non-reference homographies that are composed using the epipole and vectors $\mathrm{s}_{ref,j}$.

In this section, focal distances are also computed with the method described in [28] by Z. Zhang in order to make a practical comparison. The output of Zhang's method is used as a reference for our method. We can not use [28] as ground-truth for the method developed in this work, due to Zhang's method is model-based, i.e. it needs prior knowledge of the observed planar object, in contrast with our theory that only depends on the movement of the camera and on the initial structure of the scene, i.e. that it contains planar structures.

4.1 Experiments

We performed several experiments with our own datasets with the following features:

- fixed scene, i.e. no mobile objects in the scene,
- no objects occluding between the camera and the surfaces,
- same camera with fixed parameters for both views.

We arrange our experiments in the following way: in the first experiment we introduce a complete execution of our method, and we also present localization and reconstruction parameters computed with Zhang's method for the same dataset. The second experiment computes the focal distance of the camera with 26 pairs of images. The following two experiments show degenerate configurations in which our method can not be applied. The last three experiments run with datasets recollected in outdoor environments.

4.1.1 Interpreting results

All the experiments in this section implement the algorithms described in chapters 2 and 3. We compute translation and rotation vectors that assume the global reference frame at the first camera center. Figure 2.1 shows the reference frame used in this work, we reproduce this image in Figure 4.1.
The translation vector $\mathbf{t} = (t_1, t_2, t_3)^T$ must be interpreted as:

- t_1 is the displacement of the camera on the $\mathbf{X}$ axis. If we are observing the scene, a positive value in t_1 means that a step to our left was made.
- The displacement of the camera on a vertical line, perpendicular to the $\mathbf{Y}$ axis, is reflected in t_2. A positive value in t_2 means that the camera shifted upwards.
- t_3 measures the displacement of the camera on the $\mathbf{Z}$ axis. t_3 is positive if we got closer to the scene, with normal direction respect to the $\mathbf{XY}$ plane.

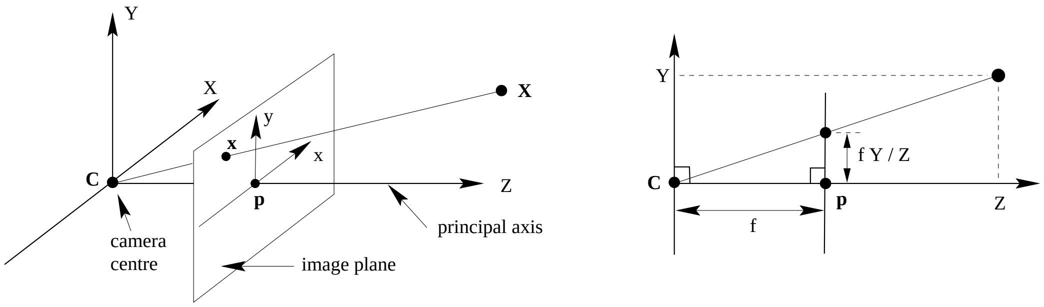

Figure 4.1: Pinhole camera geometry. $\mathbf{C}$ is the camera center and $\mathbf{p}$ is the principal point. The focal distance is denoted by f. The image plane is placed in front of the camera center.

The rotation matrix $\mathbf{R}$ is directly computed by means of the Faugeras-Lustman Algorithm, see Section 3.4. In our experiments we do not show the rotation matrix, instead we use a rotation vector $\mathbf{r}$ with a rotation angle θ, see [10] for a theoretical description of the algorithm used to obtain $(\mathbf{r}, \theta)$ from $\mathbf{R}$ and [3] for implementation notes. A rotation vector $\mathbf{r}$ is used to indicate that we turn the camera around this axis with an angle θ. One advantage of using the rotation axis instead of the rotation matrix is due to the easy interpretation of results.

Execution time

The execution time of each experiment depends on:

- I/O functions.
- Images size.
- Number of detected features and matching stage.
- Number of plane structures detected in the scene.
- Number of iterations, among other implementation and external factors.

In this work, there is no high-level algorithm with a fixed execution time, even so the current version of our implementation brings the desired results in less than two seconds when loading the input images from the file system. This execution time is the average of several experiments, and can be reduced if the image data is acquired from frame-buffer devices.

4.1.2 Experiment 1: Benny the Ball

This experiment is carried out with the two views shown in Figure 4.2 (a,b). Algorithm 1 is used in the following experiment. The first step consists of running a feature detector algorithm on the two views. In this work, the Harris corner detector is used with standard values [3]. For the ZNCC, the window $\mathbf{W}$ is a region of 11×11 pixels. The RANSAC threshold is 2.0 as defined in [10].

The original size of these views is 640×480. The iterative stage of our method only needs of five iterations to reach the desired visual accuracy[1]. The vectors that are calculated in this experiment are: $\mathbf{e}$, $\mathbf{s}_{ref,2}$, $\mathbf{s}_{ref,3}$ and $\mathbf{s}_{2,3}$. The $\mathbf{s}_{2,3}$ vector is used to describe the intersection between the two non-reference planes. Figure 4.2 shows the images resulting from this experiment.

Localization and reconstruction using focal distances computed with our algorithm by means of Xu's method

Using the algorithm developed in this work, we obtained this focal distance: $f = 996.630$, therefore the camera matrix is as follows:

$$\mathbf{K} = \begin{pmatrix} 996.630 & 0 & 0 \\ 0 & 996.630 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Localization and reconstruction parameters are computed with this matrix by means of the Faugeras-Lustman Algorithm:

- rotation axis between views: $(0.00761, 0.11469, 0.01837)^T$,
- angle of rotation between views: 6.66955 degrees,
- translation vector: $(0.48372, -0.03769, -0.04429)^T$,
- normal vector to the reference plane: $\mathbf{v}_{ref} = (-0.15566, -0.20573, 0.96615)^T$,
- normal vector to the 2nd plane: $\mathbf{v}_2 = (0.63066, -0.14145, 0.76306)^T$,
- normal vector to the 3rd plane: $\mathbf{v}_3 = (-0.13737, 0.84533, 0.51628)^T$,

[1] Visual accuracy means that we stop our method when the segmentation stage is finished.

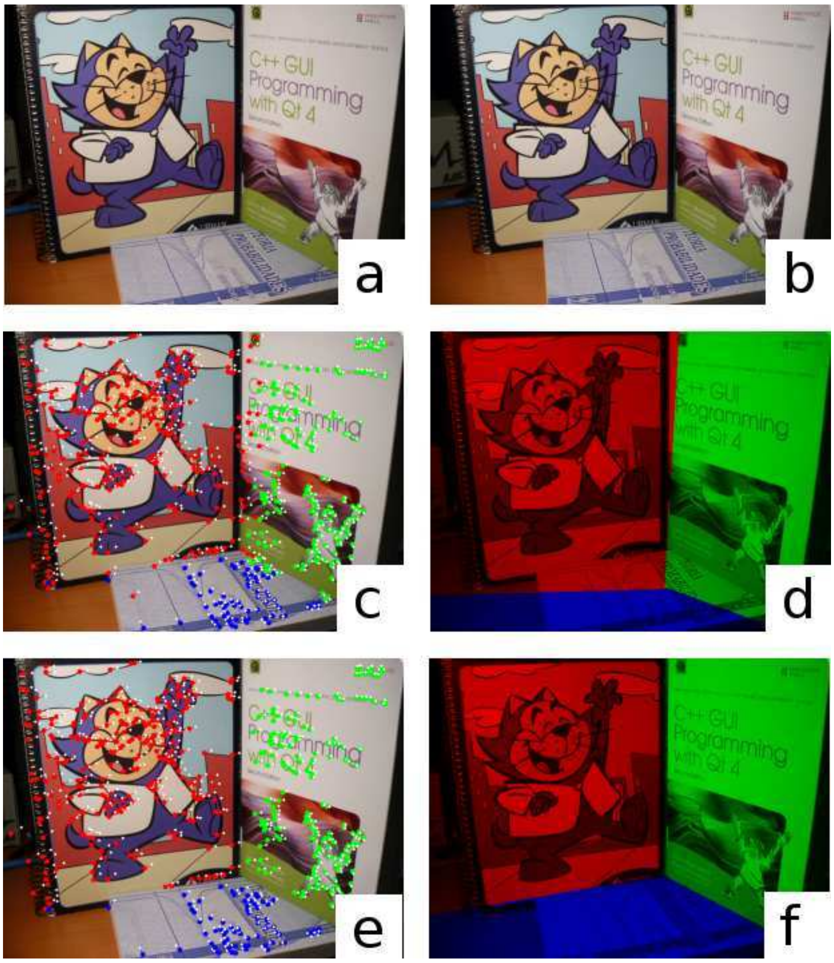

Figure 4.2: Experiment 1: Benny the Ball. (a,b) are two different views of the same multi-planar scene. (c) segmented correspondences for the detected planar surfaces. Pairs in the reference plane are red. Pairs for the second and third plane are green and blue respectively. (d) first approximation of the segmentation stage. (e) improvement of the support of each planar surface. (f) final segmentation.

- angle between the reference plane and the 2nd plane: 122.0268 degrees,
- angle between the reference plane and the 3rd plane: 110.2554 degrees,
- angle between the 2nd and 3rd plane: 101.8432 degrees.

Localization and reconstruction using focal distances from Zhang's method

Using Zhang's method we compute the next camera matrix:

$$\mathbf{K} = \begin{pmatrix} 709.676 & 0 & 0 \\ 0 & 731.667 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

This camera matrix is used to compute the localization and reconstruction parameters by means of the Faugeras-Lustman Algorithm:

- rotation axis between views: $(0.00810, 0.09255, 0.02354)^T$,
- angle of rotation between views: 5.49100 degrees,
- translation vector: $(0.42713, -0.02740, 0.01432)^T$,
- normal vector to the reference plane: $\mathbf{v}_{ref} = (-0.34047, -0.26459, 0.90226)^T$,
- normal vector to the 2nd plane: $\mathbf{v}_2 = (0.84644, -0.17245, 0.50379)^T$,
- normal vector to the 3rd plane: $\mathbf{v}_3 = (0.04125, 0.97386, 0.22335)^T$,
- angle between the reference plane and the 2nd plane: 102.2390 degrees,
- angle between the reference plane and the 3rd plane: 94.02559 degrees,
- angle between the 2nd and 3rd plane: 91.17444 degrees.

These last two angles should be 90 degrees, roughly. Figure 4.2 (c) shows the support of each planar surface. This support is segmented via the RANSAC paradigm. Figure 4.2 (e) shows the improved support obtained by means of the vectors $\mathrm{s}_{ref,j}$. The estimation of the epipole and the family of vectors $\mathrm{s}_{ref,j}$ is improved using this support. Figure 4.2 (d) shows the dense support of each detected planar surface. This segmentation is improved in Figure 4.2 (f) via the new iterative estimation of the epipole and vectors $\mathrm{s}_{ref,j}$.

As it can be observed from this experiment and from [10, 24, 25] the accuracy of the focal distance is a fundamental parameter for the localization and reconstruction tasks. Better results, in these two tasks, were obtained when using the focal distance computed with the Zhang's method.

4.1.3 Experiment 2

Experimentally it can be observed that the accuracy of Xu's method strongly depends on the quality of the segmentation of features. Xu uses a bundle adjustment in order to improve all the computed parameters [25]. We do not use that resource to improve the focal distance.

In this experiment we only compute the focal distance using 26 pairs of images. The mean and the standard deviation of these parameters are as follows:

μ	868.70
σ	208.28

Comparing this result with the focal distance obtained by means of Xu's method, we observe that we are far from the ground-truth value: 709.676. Using μ as focal distance for the same dataset used in the first experiment, we obtain:

- rotation axis between views: $(0.00758, 0.09968, 0.01664)^T$,
- angle of rotation between views: 5.80664 degrees,
- translation vector: $(0.47700, -0.04196, -0.03565)^T$,
- normal vector to the reference plane: $\mathbf{v}_{ref} = (-0.16581, -0.22964, 0.95905)^T$,
- normal vector to the 2nd plane: $\mathbf{v}_2 = (0.70908, -0.14175, 0.69073)^T$,
- normal vector to the 3rd plane: $\mathbf{v}_3 = (-0.12112, 0.91702, 0.38002)^T$,
- angle between the reference plane and the 2nd plane: 125.26958 degrees,
- angle between the reference plane and the 3rd plane: 79.98242 degrees,
- angle between the 2nd and 3rd plane: 87.32818 degrees.

Localization parameters are closer to those from ground-truth focal distance than the data resulted from our method using Xu's method. We can also conclude that reconstruction parameters were improved when comparing with our first result.

4.1.4 Experiment 3: Books corner

For the images in Figure 4.3, it is not possible to segment the correspondences when the camera movement is minimal. Although equation 2.4 $\mathbf{H} \sim \mathbf{K}_2(\mathbf{R} - \mathbf{t}\mathbf{v}^T)\mathbf{K}_1^{-1}$ can deal with zero translation and null rotation, it is not possible to segment pairs of correspondences. Thus, we can not apply our method to this pair of *similar* images. The resulting reference homography tends to the identity matrix when using almost the same camera for both views.

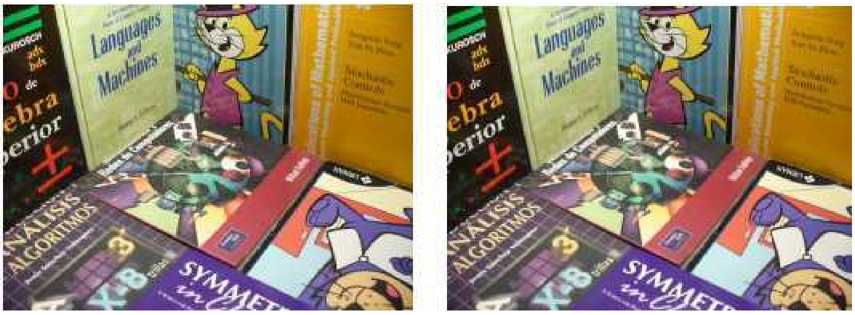

Figure 4.3: Experiment 3: Books corner. Two different views of the same multiplanar scene. Segmentation of correspondences fails due to the minimal camera displacement.

4.1.5 Experiment 4

From [6] we know that a camera translation on the camera axis is a degenerate movement that does not allow to do localization and reconstruction tasks. In Figure 4.4, the second image was obtained by means of a movement on the camera axis with minimal rotation.

All the degenerate configurations described in [6] are inherited by our method, see Sub-section 2.9.1.

4.1.6 Experiment 5: House's facade

This experiment has been conducted in an outdoor environment, partially capturing the facade of a house (Figure 4.5). For this experiment, only two planes are detected with a RANSAC threshold of 1.0. The computed parameters are:

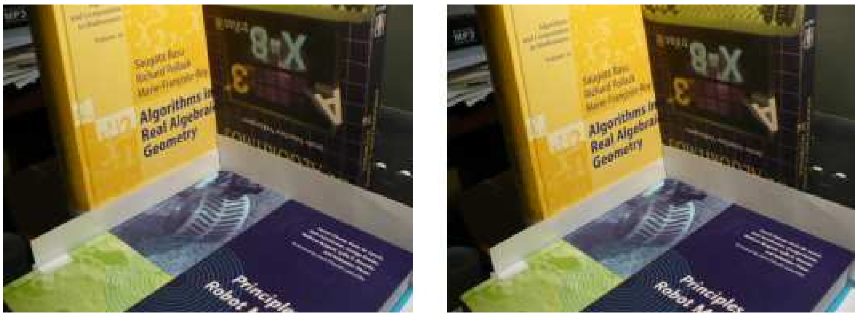

Figure 4.4: Experiment 4. Two different (but very similar) views of the same multi-planar scene. Segmentation of correspondences fails due to the type of translation movement of the camera.

- rotation axis between views: $(0.01264, -0.02440, 0.00899)^T$,
- angle of rotation between views: 1.65649 degrees,
- translation vector: $(0.21152, 0.10370, -0.02991)^T$,
- normal vector to the reference plane: $\mathbf{v}_{ref} = (-0.11049, 0.33498, -0.93573)^T$,
- normal vector to the 2nd plane: $\mathbf{v}_2 = (-0.50111, -0.73967, -0.44920)^T$,
- angle between the reference plane and the 2nd plane: 76.82519 degrees.

As can be seen in Figure 4.5, both images are close. This fact is reflected in the computed parameters and only one iteration of our method was necessary. Reconstruction parameters were calculated using the focal distances obtained by means of Zhang's method. The angle between the normals should be 90 degrees.

Segmentation results for this experiment are shown in Figure 4.6, reference plane is red and second plane is green.

4.1.7 Experiment 6

This experiment has a similar dataset as before, in which only two planes are detected, see Figure 4.7. RANSAC threshold was set to 1.0, the computed parameters are:

- rotation axis between views: $(-0.01633, -0.07638, -0.04577)^T$,

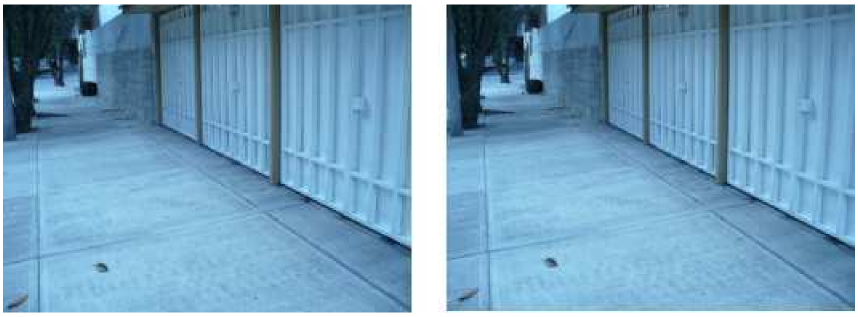

Figure 4.5: Experiment 5: House's facade. Two different images partially capturing the facade of a house.

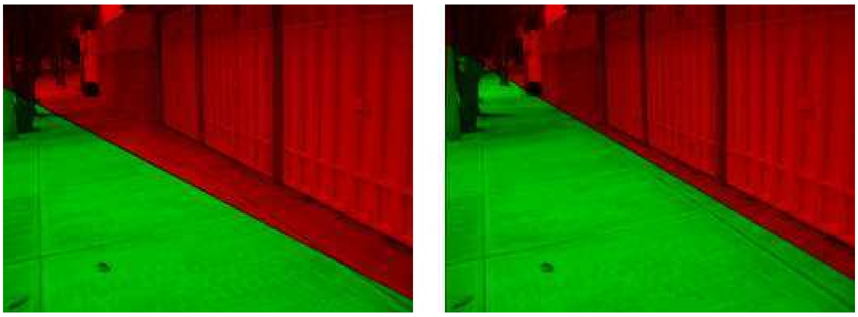

Figure 4.6: Segmentation results for experiment 5: House's facade. (left) Segmentation of planes obtained at the first stage of our method. (right) Segmentation is improved after one iteration of our method.

- angle of rotation between views: 5.18679 degrees,
- translation vector: $(0.26049, -0.01733, -0.06164)^T$,
- normal vector to the reference plane: $\mathbf{v}_{ref} = (-0.79572, 0.32820, -0.50903)^T$,
- normal vector to the 2nd plane: $\mathbf{v}_2 = (-0.17599, -0.93412, -0.31055)^T$,
- angle between the reference plane and the 2nd plane: 90.48458 degrees.

The angle between the normals is almost 90 degrees. With this dataset, only one iteration of our method was necessary. Focal distances were calculated by means of Zhang's method.

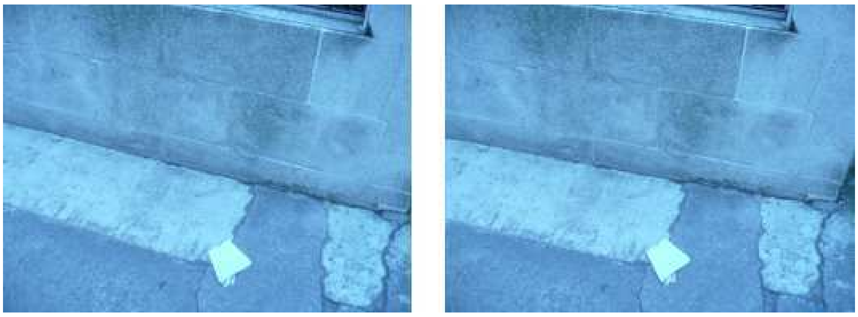

Figure 4.7: Experiment 6. Another outdoor experiment, wall and floor with almost the same color and texture.

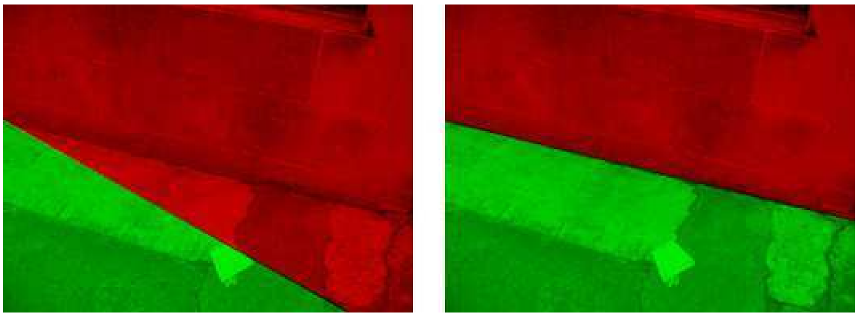

Figure 4.8: Segmentation results for experiment 6. (left) Segmentation of planes obtained at the first stage of out method. (right) Segmentation is improved after one iteration of our method.

Figure 4.8 shows segmentation results for this experiment. In this figure, reference plane is red and second plane is green. First segmentation is carried out with vector $s_{ref,2}$ computed at first stage of our method. Segmentation is improved after one iteration at second stage.

4.1.8 Experiment 7: Wall and grass

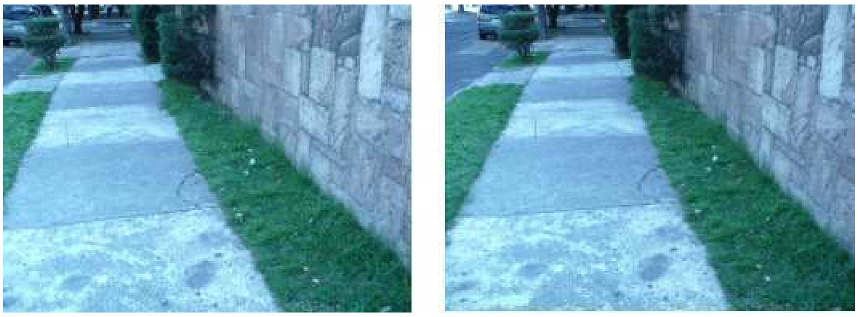

Figure 4.9: Experiment 7: Wall and grass. In this experiment, several iterations were necessary in order to align vector $s_{ref,2}$ with the physical intersection of planes.

Our last experiment, as in experiments 5 and 6, captures the facade of a house. In this experiment the RANSAC threshold is 1.2. In this experiment two planes were detected, see Figure 4.9. Figure 4.10 shows segmentation results: reference plane is red and second plane is green.

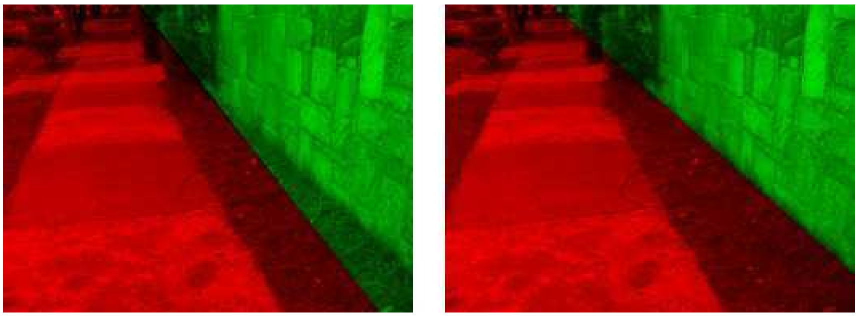

Figure 4.10: Segmentation results for Experiment 7: Wall and grass. (left) Segmentation computed at the first stage of our method. (right) Segmentation of planes is improved after seven iterations of our method.

Seven iterations were necessary in order to align vector $\mathbf{s}_{ref,2}$ with the projection of the physical intersection of the planes. The computed parameters are:

- rotation axis between views: $(0.04163, 0.04834, 0.05576)^T$,
- angle of rotation between views: 4.85450 degrees,
- translation vector: $(-0.46257, -0.11179, -0.48106)^T$,
- normal vector to the reference plane: $\mathbf{v}_{ref} = (-0.74298, 0.63503, 0.21146)^T$,
- normal vector to the 2nd plane: $\mathbf{v}_2 = (-0.75771, 0.16569, 0.63120)^T$,
- angle between the reference plane and the 2nd plane: 143.289 degrees.

Zhang's method was used to compute the focal distances involved in the reconstruction stage. The computed angled between planes should be 90 degrees roughly.

Chapter 5

Conclusions and Future Research

In this research, we have proposed a linear method to recover the camera calibration parameters of two cameras and the structure of the main planar surfaces projected by these cameras. Experimentally, we have observed that the accuracy of the solution depends on the segmentation of the correspondence pairs, both at the initial and iterative stages. The use of a feature detector in both views constrains our work to scenes in which texture is present. Another limitation of our work consists of the segmentation of the dense support. An exact segmentation is carried out only if the scene is perfectly polyhedral. In other cases, wrong pixels may be considered as part of the support of each plane.

If a planar surface is not detected at the first stage of our method, it is not possible to recover it at the iterative stage. Goodness of key point detection stage influences significantly the matching process, therefore, accurate segmentation of correspondences is conditioned by these two tasks. Determining the focal distances is not an easy task, even if both cameras have the same parameters. Xu, in [25], attacks the stability problem of camera calibration with a non-linear bundle adjustment, carried out after determining the linear approximation of the focal distances.

The mathematical theory developed so far allows us to use minimal information, i.e. two images. It only requires that the user gives the two views, the RANSAC threshold and the maximum number of iterations for the iterative stage. This is an advantage over the previous works, where an expert user supplies the correspondence pairs [6, 25] or others where previous knowledge of the dense support of each planar surface is required [17, 23, 25]. All previous works that deal with localization and reconstruction problems require the camera matrix.

Ensuring homography consistency is one of the most important contributions of our work. Although there exist previous works intending to deal partially or totally with all the tasks that we cover, our work is theoretically more stable because it is not based in explicit epipolar geometry computation.

The linear systems presented in this work allow us to use this framework in real-time systems for tasks such as map building, augmented reality and robot localization. As has been used in other referred works, a *bundle adjustment* can be conducted in order to improve the precision of the calibration and reconstruction parameters. The use of our framework for video processing is direct and may require non-linear filtering theory as has been shown in [6, 16]. In order to obtain a more user-independent system, radial distortion correction [10] might be considered for future work.

Bibliography

[1] Adrien Bartoli. A Random Samping Strategy For Piecewise Planar Scene Segmentation. *Computer Vision and Image Understanding*, 2006.

[2] Adrien Bartoli, Peter Sturm, and Radu Horaud. A Projective Framework for Structure and Motion Recovery from Two Views of a Piecewise Planar Scene. Research Report RR-4070, INRIA, 2000.

[3] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media, Inc., 2008.

[4] Roberto Brunelli. *Template Matching Techniques in Computer Vision: Theory and Practice*. Wiley, 2009.

[5] L. de Agapito, R. I. Hartley, and E. Haymen. Linear self-calibration of a rotating and zooming camera. In *CVPR*, pages 15–21, 1999.

[6] Olivier Faugeras and F. Lustman. Motion and Structure from Motion in a Piecewise Planar Environment. Technical Report RR-0856, INRIA, 06 1988.

[7] C. Sagues G. Lopez-Nicolas and J. J. Guerrero. Automatic Matching and Motion Estimation From Two Views of a Multiplane Scene. *Pattern Recognition and Image Analysis*, LNCS 3522:68–76, 2005.

[8] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (2nd Edition)*. Prentice Hall, January 2002.

[9] C. Harris and M. Stephens. A combined corner and edge detection. In *Proceedings of The Fourth Alvey Vision Conference*, pages 147–151, 1988.

[10] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[11] Ezio Malis and Roberto Cipolla. Multi-View Constraints Between Collineations: Application to Self-Calibration From Unknown Planar Structures. In *6th European Conference on Computer Vision, Dublin, Ireland*, volume LNCS 1843, pages 610–624. Springer-Verlag, 2000.

[12] J. Alison Noble. Finding corners. *Image Vision Comput.*, 6(2):121–128, 1988.

[13] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, New York, USA, August 1999.

[14] William Press, Saul Teukolsky, William Vetterling, and Brian Flannery. *Numerical Recipes in C*. Cambridge University Press, Cambridge, UK, 2nd edition, 1992.

[15] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47:7–42, 2002.

[16] Gilles Simon and Marie-Odile Berger. Detection of the Intersection Lines in Multiplanar Environments: Application to Real-Time Estimation of the Camera-Scene Geometry. In *19th International Conference on Pattern Recognition - ICPR 2008*, 2008.

[17] Gilles Simon, Andrew W. Fitzgibbon, and Andrew Zisserman. Markerless Tracking using Planar Structures in the Scene. In *Proc. International Symposium on Augmented Reality*, 2000.

[18] Stephen M. Smith and J. Michael Brady. SUSAN - A New Approach to Low Level Image Processing. *Int. J. Comput. Vision*, 23(1):45–78, 1997.

[19] Joan Solà. *Towards Visual Localization, Mapping and Moving Objects Tracking by a Mobile Robot: a Geometric and Probabilistic Approach.* PhD thesis, LAAS-CNRS Toulouse, France, 2007.

[20] S. Thrun, W. Burgard, D. Chakrabarti, R. Emery, Y. Liu, and C. Martin. A Real-time Algorithm for Acquiring Multi-Planar Volumetric Models with Mobile Robots. In *Proceedings of the 10th International Symposium of Robotics Research (ISRR'01)*, Lorne, Australia, 2001. Springer.

[21] Miroslav Trajkovic and Mark Hedley. Fast corner detection. *Image Vision Comput.*, 16(2):75–87, 1998.

[22] Roger Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *Radiometry*, pages 221–244, 1992.

[23] Javier Flavio Vigueras and Mariano Rivera. Registration and Iteractive Planar Segmentation for Stereo Images of Polyhedral Scenes. *Elsevier Pattern Recognition, special edition on Interactive Image Processing*, 2009.

[24] Javier Flavio Vigueras, Gilles Simon, and Marie-Odile Berger. Calibration Errors in Augmented Reality: a Practical Study. In *4th IEEE and ACM International Symposium on Mixed and Augmented Reality - ISMAR'05*, pages 154–163, Vienna Austria, 10 2005. IEEE.

[25] Gang Xu, Jun ichi Terai, and Heung-Yeung Shum. A linear algorithm for Camera Self-Calibration, Motion and Structure Recovery for Multi-Planar Scenes from Two Perspective Images. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 474–479, 2000.

[26] Gang Xu and Zhengyou Zhang. *Epipolar Geometry in Stereo, Motion, and Object Recognition: A Unified Approach*. Kluwer Academic Publishers, Norwell, MA, USA, 1996.

[27] Z. F. Zhang and A. R. Hanson. Scaled Euclidean 3d Reconstruction Based on Externally Uncalibrated Cameras. In *SCV95*, pages 37–42, 1995.

[28] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:1330–1334, 1998.