Universitat
de Girona

Department of Electronics, Computer Science and Automatic Control

PhD Thesis

MODELLING STEREOSCOPIC VISION SYSTEMS FOR ROBOTIC APPLICATIONS

Thesis presented by Xavier Armangué Quintana,
to obtain the degree of:
PhD in Computer Engineering.

Supervisors:
Dr. Joaquim Salvi Mas
Dr. Helder Araújo

Girona, July 2003

Modelling Stereoscopic Vision Systems for Robotic Applications

by

Xavier Armangué Quintana

To those who have helped me.
Als qui m'han ajudat.

Modelling Stereoscopic Vision Systems for Robotic Applications

Abstract

Human eyes have been widely studied by the scientific community so that its operation principle is widely known. Computer vision tries to copy the way human beings perceive visual information by means of using cameras acting as eyeballs and computers aspiring to process this information in an "intelligent way". The complex task of being conscious of reality is obviously divided into a set of simpler problems which covers from image acquisition to scene description. One of the main applications is robot perception in which a mobile robot is equipped with a computer vision system. Robots may be able to navigate around an unknown structured environment acquiring visual information of their surroundings with the aim of estimating the position and orientation of every obstacle. Moreover, the pose of the vehicle has to be estimated as accurate as possible. Hence, the motion of the vehicle might be also computed allowing the localization of the vehicle with respect to the 3D map.

This thesis is focused on the study of the geometry involved in stereo vision systems composed by two cameras with the aim of obtaining 3D geometric information of the vehicle surroundings. This objective deals to the study of *camera modelling and calibration* and the comprehension of the epipolar geometry. Then, the computation of the *fundamental matrix* of a stereoscopic system is surveyed with the aim of reducing the correspondence problem between both image planes. An accurate estimation of the fundamental matrix allow us not only to compute 3D information of the vehicle environments, but to validate it. Nevertheless, the traditional case of the epipolar geometry has some limitations in the common case of a single camera attached to a mobile robot. Disparities between two consecutive images are rather small at common image rates leading to numerical inaccuracies on the computation of the fundamental matrix. Then, another objective is the study of general vision-based egomotion estimation methods based on the *differential epipolar constraint* with the aim of perceiving the robot movement instead of its position.

The study of the geometry involved in stereo vision systems leads us to present a computer vision system mounted on a vehicle which navigates in an unknown environment. Two main tasks are faced: a) the localization of the vehicle; and b) the building of an absolute 3D map.

Agraïments

Aquesta tesi va dedicada a tots aquells que d'una manera o altre m'han ajudat a arribar fins aquí. Sense la seva ajuda m'hauria quedat pel camí.

Primer de tot vull donar les gràcies al meu director en Joaquim Salvi qui em va introduir al món de la percepció 3D marcant el camí a seguir, he anat tocant diferents temes que m'han permès tenir una punt de vista amplia de la visió per computador. També agrair la seva paciència a l'hora d'escriure els diversos articles com d'aquesta tesi.

Vull agrair a l'Helder Araújo, co-director d'aquesta tesi, que acceptar-me en el seu grup durant sis mesos (en dues estades diferents). Aquest temps m'ha permet conèixer una altre manera de treballar, disposar d'un espai, recursos i de la tranquil·litat necessària.

Agraeixo el suport rebut pels membres del Grup de Visió per Computador i Robòtica però també altres membres del departament. Gràcies per acompanyar-me a la platja o per esquivar-nos mentre anàvem a la platja!! Gràcies per venir a jugar a tennis per ensenyar-me'n, per jo intentar ensenyar-vos o simplement per practicar. Gràcies per portar-me de concert i conèixer altres estils de música. Gràcies per venir-me a visitar, encara que només sigui per fer una capbussada. Gràcies per fer sortides en bicicleta tot i que després quedi destrossat. Gràcies per jugar a ping-pong i no haver-me deixat guanyar mai!! Gràcies per acompanyar-me en el viatge tan divertit que vam passar fent amics entre la comunitat asiàtica. Gràcies per tots aquells moments davant la màquina del cafè. Gràcies per organitzar les partits de voleibol on més que jugar es tracta de passar una estona divertida. Gràcies pels correus amb missatges divertits per passar la estona. Gràcies per venir als sopars per passar-ho bé especialment a aquells sopars més memorables a base de pizza o altres sopars més reduïts per falta de quòrum!!! Gràcies per acompanyar-me al cinema a veure tantes i tantes pel·lícules. Però no tot han estat bons moments i també voldria recordar quan les coses no han anat tan bé i m'han ajudat a superar-ho encara que només fos escoltant les meves misèries.

Dono les gràcies a tots els companys del ISR de la Universitat de Coimbra que em van acollir en el seu grup, que em van facilitar l'adaptació a la seva ciutat i em van ajudar en diverses coses. Em sap una mica de greu no haver pogut correspondre aprenent una mica de portuguès ja què no vaig arribar a parlar-lo mai tot hi que el vaig arribar a entendre força.

També vull donar les gràcies a tots aquells que han compartit durant aquests anys, des del pis destartalat del carrer Cardenal Margarit, passant pel pis de la zona de Lluís Pericot fins al pis del carrer Migdia. Amb tots ells viscut bons moments permetent-me desconnectar de la feina diària, fent-me riure molt cadascú a la seva manera.

També vull recordar a tots els meus amics, tan els del sector Roses com als del sector Figueres, que tot i que no es veiem gaire sovint, moltes vegades per culpa meva per treballar massa, sempre s'han interessat per la meva feina i fent-me veure altres maneres de viure la vida.

Dono les gràcies a la meva família que m'han donat suport tot aquests anys, tot hi que no han acabat mai d'entendre que estic fent han confiat amb mi i m'han deixat escollir el meu propi camí.

Acknowledgements

This thesis is dedicated to everybody who have helped me in this long travel. Without their help I wouldn't have reach it.

First of all, I would like to thank a to my director Dr. Joaquim Salvi, who has introduced me in the 3D perception world. Along these years I have known several subjects that provide me a wider vision of the computer vision world.

I would like to thank Dr. Helder Araújo, co-director of this thesis, for offering me the opportunity to work in his group during my stages in Coimbra. All along this time allowed me to know another way of working and to enjoy the sufficient space, resources and required tranquility.

I thank Dr. Z. Zhang for the implementation of the correspondence problem solution utilized in chapter 3 to fundamental matrix estimation with real images.

I greatly appreciate Dr. W. Chojnacki and Dr. A. van den Hengel who gave me detailed information and the code of their methods (FNS and CFNS), as well as Dr. P. H. S. Torr who left the implementation of the MAPSAC and MLESAC methods in its web page.

I appreciate a lot Dr. Tina Y. Tian, Dr. Carlo Tomasi and Dr. David J. Heeger who implemented the methods explained in section 4.4.2 which have been compared with the rest of methods explained in the thesis, specially Dr. Heeger who gave us insightful information and the source code of such methods.

Finally, I want to thank Dr. Rafael García for providing me me the mosaicking application developed by the Underwater Vision Lab, which has been very useful to design the localization and mapping application described in chapter 5.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter describes the motivation leading to the presentation of this thesis and how it is related to a CICYT project funded by the Spanish government. The objectives of the project and the subjects included in this document are briefly explained. The chapter ends describing the structure and contents of the remaining of the thesis.

1.1 The Human Eye versus Computer Vision

Human beings own five different senses: vision, taste, smell, touch and hearing. These senses keep us in touch with our surrounding, allowing us to perceive and interact with our environment. Albert Einstein defined reality in the following way: “Reality is nothing more than the capability of our senses to be wrong”. However, human senses provide the only way of acquiring information from “reality”.

Human senses can be classified into two groups. First, taste and touch are considered touching senses because a direct contact with the sensing object is required. Vision, smell and hearing are considered indirect senses due to the ability of such senses to perceive information at a certain distance from the object. The information acquired by the different sensors gives a different perceptual view of the surroundings. However, of the five vision is considered the most developed human sense.

The perception of the environment is completed in two steps. First, the information is acquired through at least one of the human senses. Second, this information is processed to get an interpretation of the environment perceived. With human vision, the information is acquired by means of the eyes and interpreted by the human brain.

Human eyes have long been the subject of medical and biological studies so that their functioning is well documented. Basically, light entering the cornea is focused onto the retinal surface by a lens which under muscular control changes shape, to perform proper focusing on near and distant objects. The iris acts as a shutter to control the amount of light entering the eye. The retina is composed of two types of receptors: rods and cones. Nerves, carrying the visual message, leave the retina and pass through the optic nerve bundle to the brain. The rods are long slender receptors extremely sensitive to light, while cones are only sensitive to high level of illumination. We perceive three-dimensional information by using our eyes and the knowledge of the most common objects which we achieved through living. Overall, the behavior of our brain is very complex and the manner in which information is processed is still largely unknown, and remains one of the most exciting research subjects for the scientific community.

Computer vision tries to copy the way how human beings perceive visual information by means of using cameras acting as eyeballs and computers to process the information in an “intelligent way” as does the human brain. The cameras are designed based on the knowledge of the eyeball operation so that some comparisons can be established: the shutter corresponds to the human iris, the camera lenses and focus correspond to the muscular human lens, and the CCD[1] array corresponds to the retina. The computer vision community has been studying ways to process the visual information acquired by these cameras with the aim of perceiving and being conscious of reality.

The complex task of understanding of reality has been divided into a set of simpler problems starting from what is considered low-level image processing such as thresholding, edge detection and filtering and proceeds through to high-level image processing tasks such as texture analysis, feature extraction and motion detection ending with the description and interpretation level for shape identification and scene description.

---

[1] Charge-Coupled Device

1.2 Context and Motivation

This thesis was partially funded by the CICYT[2] project TAP[3] 1999-0443-C05-01. The main objective of this project is the design, implementation and accuracy evaluation of mobile robots fitted with distributed control, sensing and a communicating network. The robots must be able to navigate around an unknown structured environment estimating the position and orientation of every obstacle in their surroundings. This information must allow the robots to compute their localization in the environment so they can avoid any potential obstacle and accomplish the mission. Each robot must be able to generate a dynamic map of the environment which will be used to localize the robot. The main objective of this project is to allow the robots to explore and build a map of the environment so that no modification of the environment is needed.

In order to achieve accurate navigation of the vehicle, it is quite important to equip the robot with a sensing system, permitting the acquisition of information about the vehicle's surroundings. When considering an easy task of obstacle avoidance, the use of range systems such us ultrasonic and infrared sensors might be enough. Also, with the aim of acquiring more accurate information, it is interesting to study the computer vision problem as applied to mobile vehicles. The main objective of this project is to fit every robot with a computer vision sensor which must allow the vehicle to acquire information of the environment so that a 3D map might be built dynamically. Moreover, the position of the vehicle has to be estimated as accurate as possible by using the 3D map alone. Hence, the motion of the vehicle should also be computed allowing the localizing of the vehicle with respect to the 3D map.

1.3 Objectives

The main objectives of this thesis are described in the following paragraphs.

The study of the geometry involved in stereo vision systems composed of two cameras with the aim of obtaining 3D geometric information of the vehicle's surroundings. This objective deals with the study of camera modelling and calibration

---

[2] Centro de Investigación Científica y Tecnológica

[3] Tecnologías Avanzadas de Producción

and the comprehension of epipolar geometry. Then, the computation of the fundamental matrix of a stereoscopic system is surveyed with the aim of reducing the correspondence problem between both image planes. An accurate estimation of the fundamental matrix allows us not only to compute 3D information of the vehicle's environment but to validate it as well.

The traditional case of epipolar geometry has some limitations in the common case of a single camera attached to a mobile robot. Disparities between two consecutive images are rather small at the usual image rates, leading to numerical inaccuracies on the computation of the fundamental matrix. Therefore, a second objective should be the study of general vision-based egomotion estimation methods based on the *differential epipolar constraint* with the aim of perceiving a 6-DOF movement. However, the geometry should be adapted to the common case of a mobile robot moving on a plane constraining the movement to 2-DOF, leading to more accurate results.

The study of the geometry involved in stereo vision systems should allow us to present an application which permits a vehicle to move in an unknown environment. Two main tasks should be faced: a) the localization of the vehicle and b) the building of an absolute map while the robot is moving.

1.4 Thesis Outline

This research survey is structured in 7 chapters, two appendices and a bibliography section detailed at the end of the document.

Chapter 2 surveys five camera calibration methods. This chapter focuses on describing each camera model by a set of calibrating parameters. First, the intrinsic parameters dealing with the internal geometry and optic characteristics of the image sensor are described. Second, the extrinsic parameters dealing with the position and orientation of the camera with respect to a reference system are presented. Both parameters are thoroughly discussed, explaining how the camera model is obtained. Once the camera model is known, the value of every parameter is estimated, i.e. the camera must be calibrated. The different camera calibration methods are amply explained.

Chapter 3 relates to the study of the fundamental matrix estimation problem.

First, the epipolar geometry is presented. Then, several methods of estimating the fundamental matrix are described and classified into linear, iterative and robust methods. Linear methods require a low computing time but their geometry is rather inaccurate in the presence of noise. Iterative methods improve the accuracy of linear methods but can not cope with outliers. Finally, robust methods can cope with both noise and outliers, obtaining better results but they require a considerable computing time.

Some authors have proposed techniques to estimate the camera motion at a high image rate based on the computation of the linear and angular velocities instead of the orientation and translation between consecutive images. These techniques are commonly based on the optical flow and the differential epipolar constraint. Chapter 4 of this thesis presents an exhaustive survey covering the last twenty years of research in this field. The techniques surveyed here are classified and the mathematical notation is unified. The robustness of each technique with respect to both synthetic and real images is detailed. Nevertheless, all these techniques generate poor results unless they are adapted to the special case of a mobile robot moving on a plane. The adaptation is based on constraining the movement from six degrees of freedom to only two degrees, which are: a) the spin rotation of the mobile robot, and b) the forward motion.

Chapter 5 describes a system for localizing and map-building in robotics applications by means of stereo vision. This system is structured in three steps. First, points of interest are found by using image processing tools. Then the epipolar geometry is computed from the camera calibration in order to simplify the correspondence problem. Third, we proceed to reconstruct the geometry of the scene for each pair of stereo images by means of triangulation. Finally, the trajectory carried out by the robot is estimated and the absolute mapping of the environment built while the robot is moving.

Chapter 6 presents the experimental results obtained from the implementation of the surveyed methods and described in the previous chapters. This chapter is composed of four parts: camera calibration results, fundamental matrix estimation results, motion recovery results and the application related to vehicle localization and map building by means of stereovision. Synthetic and real images have been used for testing the camera calibration methods, fundamental matrix estimation methods and egomotion estimation methods. The results obtained are compared in terms of accuracy and computing time.

Finally, chapter 7 ends this document with conclusions and further work. A list of the related publications is included.

Two appendixes are attached to this document. Appendix A explains the mathematical notation used in this thesis while Appendix B describes some details related to the software implementation.

Chapter 2

Camera Modelling and Calibration

Camera calibrating is a crucial problem for further metric scene measurement. Many techniques and some studies concerning calibration have been presented in the last few years. However, it is still difficult to go into details of a determined calibrating technique and compare its accuracy with respect to other methods. Principally, this problem emerges from the lack of a standardized notation and the existence of various methods of accuracy evaluation to choose from. This chapter presents a detailed review of some of the most used calibrating techniques with the principal idea being to present them all with the same notation.

2.1 Introduction

Camera calibration is the first step towards computational computer vision. Although some information concerning the measuring of scenes can be obtained by using uncalibrated cameras [Hartley 93], calibration is essential when metric information is required. The use of precisely calibrated cameras makes the measurement of distances in a real world from their projections on the image plane possible [Faugeras 93, Haralick 92a]. Some applications of this capability include:

1. *Dense reconstruction:* Each image point determines an optical ray passing through the focal point of the camera towards the scene. The use of more than a single view of a motionless scene (taken from a stereoscopic system, a single moving camera, or even a structured light emitter) permits cross-

ing both optical rays to get the metric position of the 3D points [Ahlers 89, Batlle 98, Jarvis 83]. Obviously, the correspondence problem has to be previously solved [Zhang 93].

2. *Visual inspection:* Once a dense reconstruction of a measuring object is obtained, the reconstructed object can be compared with a stored model in order to detect any manufacturing imperfections such as bumps, dents or cracks. One potential application is visual inspection for quality control. Computerized visual inspection allows automatic and exhaustive examination of products, as opposed to the slower human inspection which usually implies a statistical approach [Newman 95].

3. *Object localization:* When considering various image points from different objects, the relative position between these objects can be easily determined. This has many possible applications such as in industrial part assembly and obstacle avoidance in robot navigation [Broggi 98, Casals 89, Charbonnier 95], among others.

4. *Camera localization:* When a camera is placed in the hand of a robot arm or on a mobile robot, the position and orientation of the camera can be computed by locating some known landmarks in the scene. If these measurements are stored, a temporal analysis allows the handler to determine the trajectory of the robot. This information can be used in robot control and path planning [Khadraoui 96, Lenz 89, Li 94].

Camera calibration is divided into two steps. The first step, called camera modelling, deals with the mathematical approximation of the physical and optical behavior of the sensor by using a set of parameters. The second step deals with the use of direct or iterative methods to estimate the values of these parameters which is again called calibration. There are two kinds of parameters in the model which have to be considered. On the one hand, there is the intrinsic parameter set which models the internal geometry and optical characteristics of the image sensor. Basically, intrinsic parameters determine how light is projected through the lens onto the image plane of the sensor. The other set of parameters are extrinsic. The extrinsic parameters measure the position and orientation of the camera with respect to a world coordinate system which, in turn, provides metric information with respect to a user-fixed coordinate system instead of the camera coordinate system.

Camera calibration techniques can be classified according to several different criteria: 1) linear versus nonlinear camera calibration (usually differentiated depending on the modelling of lens distortion) [Ito 91] and 2) intrinsic versus extrinsic camera calibration. Intrinsic calibration is concerned only with obtaining the physical and optical parameters of the camera [Lenz 88, Penna 91]. Extrinsic calibration concerns the measurement of the position and orientation of the camera in the scene [Liu 90, Wang 92]. 3) Implicit [Hall 82] versus explicit [Batista 99] calibration. Implicit calibration is the process of calibrating a camera without explicitly computing its physical parameters. Although the results can be used for 3D measurement and the generation of image coordinates, they are useless for camera modelling as the obtained parameters do not correspond to the physical ones [Wei 94]. Finally, 4) the methods which use known 3D points as a calibrating pattern [Faugeras 86, Tsai 87] or even a reduced set of 3D points [Hong 93, Kamata 92], with respect to others which use geometrical properties in the scene such as vanishing lines [Wang 91] or other line features [Chen 90, Echigo 90].

These various approaches can also be classified as regards the calibration method used to estimate the parameters of the camera model:

1. *Non-linear optimization techniques.* A calibrating technique becomes nonlinear when any kind of lens imperfection is included in the camera model. In that case, the camera parameters are usually obtained through iteration with the constraint of minimizing a determined function. The minimizing function is usually the distance between the imaged points and the modelled projections obtained by iterating. The advantage of these iterating techniques is that almost any model can be calibrated and accuracy usually increases by increasing the number of iterations up to convergence. However, these techniques require a good initial guess in order to guarantee convergence. Some examples are described in classic photogrammetry [Salvi 97, Slama 80].

2. *Linear techniques which compute the transformation matrix.* These techniques use the least squares method to obtain a transformation matrix which relates 3D points with their 2D projections. The advantage here is the simplicity of the model which consists of a simple and rapid calibration. One drawback is that linear techniques are useless for lens distortion modelling, entailing a rough accuracy of the system. Moreover, it is sometimes difficult to extract

the parameters from the matrix due to the implicit calibration used. Some references related to linear calibration can be found in Hall [Hall 82], Toscani-Faugeras [Faugeras 86, Toscani 87] and Ito [Ito 91].

3. *Two-step techniques.* These techniques use a linear optimization to compute some of the parameters and, as a second step, the remaining parameters are computed iteratively. These techniques permit a rapid calibration which reduces the number of iterations considerably. Moreover, the convergence is nearly guaranteed due to the linear guess obtained in the first step. *Two-step techniques* make use of the advantages of the previously described methods. Some references are Tsai [Tsai 87], Weng [Weng 92b] and Wei [Wei 94].

This chapter is a detailed survey of some of the most frequently used calibrating techniques (see camera calibration surveys [Armangué 00] and [Salvi 02]). The first technique was proposed by Hall in 1982 and is based on an implicit linear camera calibration by computing the 3x4 transformation matrix which relates 3D object points with their 2D image projections [Hall 82]. The latter work of Faugeras, proposed in 1986, was based on extracting the physical parameters of the camera from such a transformation technique and is explained as the second technique [Faugeras 86, Toscani 87]. The following methods are based on non-linear explicit camera calibration, including the modelling of lens distortion. Hence, the first one is a simple adaptation of the Faugeras linear method with the aim of including radial lens distortion [Salvi 97, Salvi 98]. The widely used method proposed by Tsai, based on a *two-step technique* modelling only radial lens distortion, is also detailed [Tsai 87]. Finally, the complete model of Weng, proposed in 1992, including three different types of lens distortion, is explained as the last technique [Weng 92b]. Note that one of the principal problems in understanding a calibrating technique in detail is the lack of notation standardization in mathematical equations and the use of different sets of coordinate systems. Both limitations complicate the comparing of techniques, thus a great deal of effort has been made to present the survey using the same notation. All five techniques are explained herein and their 2D and 3D accuracy shown and discussed. A brief overview of camera accuracy evaluation [Lai 93] is included with the aim of using the same tools to compare the different calibrating techniques implemented.

This chapter is structured as follows. Section 2.2 deals with camera modelling and how the camera model is gradually obtained by a sequence of geometrical transformations. Section 2.3 describes the five different techniques of camera calibration which estimate the parameters of the camera model. Then, a few methods for accuracy evaluation of camera calibrating techniques are explained in section 2.4. Finally, a summary of the chapter is presented.

2.2 Camera Model

A model is a mathematical formulation which approximates the behavior of any physical device by using a set of mathematical equations. Camera modelling is based on approximating the internal geometry along with the position and orientation of the camera in the scene. There are several camera models to choose from depending on the desired accuracy [Ito 91]. The simplest are based on linear transformations without modelling the lens distortion. However, there are also some non-linear models which accurately model the lens. These are useful for some applications where greater precision is required.

The simplest model is the one proposed by Hall [Hall 82]. The goal is to find a linear relationship among the 3D points of the scene with their 2D projecting points on the image plane. This relationship is approximated by means of a transformation matrix[1], as shown in equation (2.1).

$$\begin{pmatrix} s^I X_d \\ s^I Y_d \\ s \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \end{pmatrix} \begin{pmatrix} {}^W X_w \\ {}^W Y_w \\ {}^W Z_w \\ 1 \end{pmatrix} \tag{2.1}$$

Then, given a 3D point P_w, expressed with respect to the metric world coordinate system (i.e. ${}^W P_w$), and applying the transformation matrix proposed by Hall, the 2D point P_d in pixels with respect to the image coordinate system is obtained, i.e. ${}^I P_d = ({}^I X_d, {}^I Y_d)$.

However, camera modelling is usually broken down into 4 steps, as is hereafter detailed (see also Figure 2.1).

---

[1] The appendix A at the end of this document details the used nomenclature.

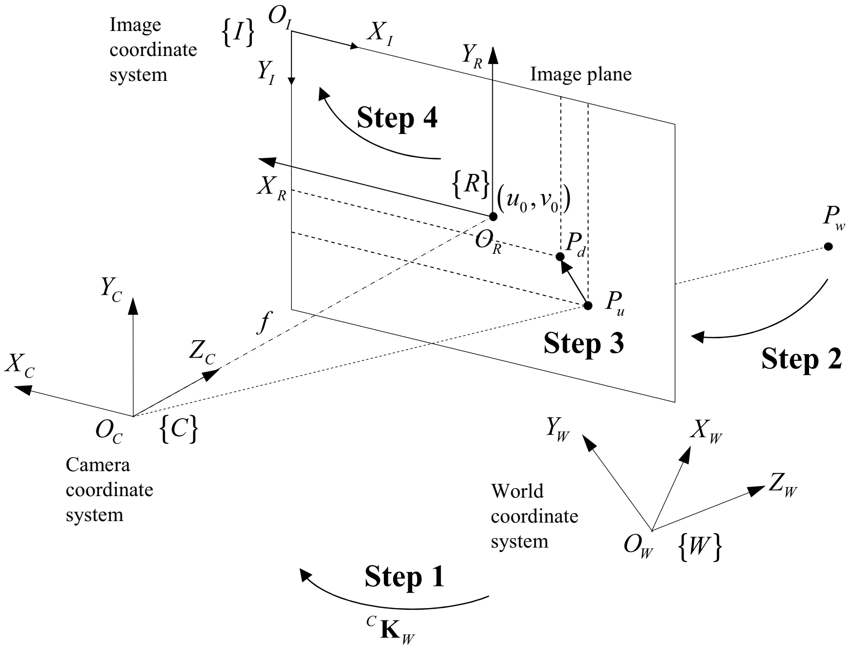


Figure 2.1: The geometric relation between a 3D object point and its 2D image projection.

1. The first step consists of relating a point $^W P_w$ from the world coordinate system to the camera coordinate system, obtaining $^C P_w$. This transformation is performed by using a rotation matrix and a translation vector.
2. Next, it is necessary to carry out the projection of point $^C P_w$ on the image plane obtaining point $^C P_u$, by using a projective transformation.
3. The third step models the lens distortion, based on a disparity with the real projection. Then, point $^C P_u$ is transformed to the real projection of $^C P_d$ (which should coincide with the points captured by the camera).
4. Finally, the last step consists of carrying out another coordinate system transformation in order to change from the metric coordinate system of the camera to the image coordinate system of the computer in pixels, obtaining $^I P_d$.

In the following, the different camera models of Faugeras-Toscani [Toscani 87], Faugeras-Toscani with distortion [Salvi 98], Tsai [Tsai 87] and Weng [Weng 92b] are explained in detail with attention on how they carry out the above four steps.

2.2.1 Camera Position and Orientation

Changing the world coordinate system to the camera coordinate system is carried out in the same way in all the surveyed models. This transformation is modelled using a translation vector and a rotation matrix, as shown in equation (2.2).

$$\begin{pmatrix} {}^{C}X_w \\ {}^{C}Y_w \\ {}^{C}Z_w \end{pmatrix} = {}^{C}\mathbf{R}_W \begin{pmatrix} {}^{W}X_w \\ {}^{W}Y_w \\ {}^{W}Z_w \end{pmatrix} + {}^{C}T_W \tag{2.2}$$

Then, given a point ${}^{W}P_w$ related to the world coordinate system, and applying equation (2.2), the point ${}^{C}P_w$ in relation to the camera coordinate system is obtained. Note that ${}^{C}\mathbf{R}_W$ expresses the orientation of the world coordinate system $\{W\}$ with respect to the axis of the camera coordinate system $\{C\}$, and that ${}^{C}T_W$ expresses the position of the origin of the world coordinate system measured with respect to $\{C\}$.

2.2.2 Perspective Projection

Consider that any optical sensor can be modelled as a *pinhole camera* [Faugeras 93]. That is, the image plane is located at a distance f from the optical center O_C, and is parallel to the plane defined by the coordinate axis X_C and Y_C. Moreover, given an object point $({}^{C}P_w)$ related to the camera coordinate system, if it is projected through the focal point (O_C), the optical ray intercepts the image plane at the 2D image point $({}^{C}P_u)$. This relation is shown in equation (2.3).

$${}^{C}X_u = f\frac{{}^{C}X_w}{{}^{C}Z_w} \qquad {}^{C}Y_u = f\frac{{}^{C}Y_w}{{}^{C}Z_w} \tag{2.3}$$

All the various models reviewed solved the projective transformation by using the same equation (2.3).

2.2.3 Lens Distortion

The third step is based on modelling the distortion of the lenses. Equations (2.4) transform the undistorted point ${}^{C}P_u$ to the distorted point ${}^{C}P_d$, where δ_x and δ_y

represent the distortion involved.

$$ {}^{C}X_u = {}^{C}X_d + \delta_x \qquad {}^{C}Y_u = {}^{C}Y_d + \delta_y \tag{2.4}$$

The camera model proposed by Faugeras and Toscani [Toscani 87] does not model the lens distortion, therefore, ${}^{C}P_u$ and ${}^{C}P_d$ are the same point. In this case δ_x and δ_y are zero, as shown in equation (2.5).

$$\delta_x = 0 \qquad \delta_y = 0 \tag{2.5}$$

The Faugeras-Toscani model, can be improved by modelling the radial lens distortion [Salvi 98], though. Besides, Tsai [Tsai 87] has modelled the lens distortion considering in the same way only radial distortion. As shown in equations (2.6), δ_x and δ_y represent the radial distortion [Slama 80]. This type of distortion is mainly caused by flawed radial curvature of the lens, shown in Figure 2.2 and 2.3. See also [Weng 92b].

$$\delta_x = \delta_{xr} \qquad \delta_y = \delta_{yr} \tag{2.6}$$

The displacement given by the radial distortion dr can be modelled by equations (2.7), which consider only k_1 the first term of the radial distortion series. It has been proven that the first term of this series is sufficient to model the radial distortion in most of the applications [Tsai 87].

$$\delta_{xr} = k_1 {}^{C}X_d \left({}^{C}X_d{}^2 + {}^{C}Y_d{}^2 \right) \qquad \delta_{yr} = k_1 {}^{C}Y_d \left({}^{C}X_d{}^2 + {}^{C}Y_d{}^2 \right) \tag{2.7}$$

The model of Weng [Weng 92b] considers three types of distortion: radial distortion, decentering distortion and thin prism distortion. The total distortion will be the sum of these three distortions.

$$\delta_x = \delta_{xr} + \delta_{xd} + \delta_{xp} \qquad \delta_y = \delta_{yr} + \delta_{yd} + \delta_{yp} \tag{2.8}$$

However, Weng proposed to model the lens distortion from the undistorted image point (${}^{C}X_u$,${}^{C}Y_u$) instead of the distorted one (${}^{C}X_d$,${}^{C}Y_d$). Although both approaches can be considered, it also has to be taken into account that the calibrating

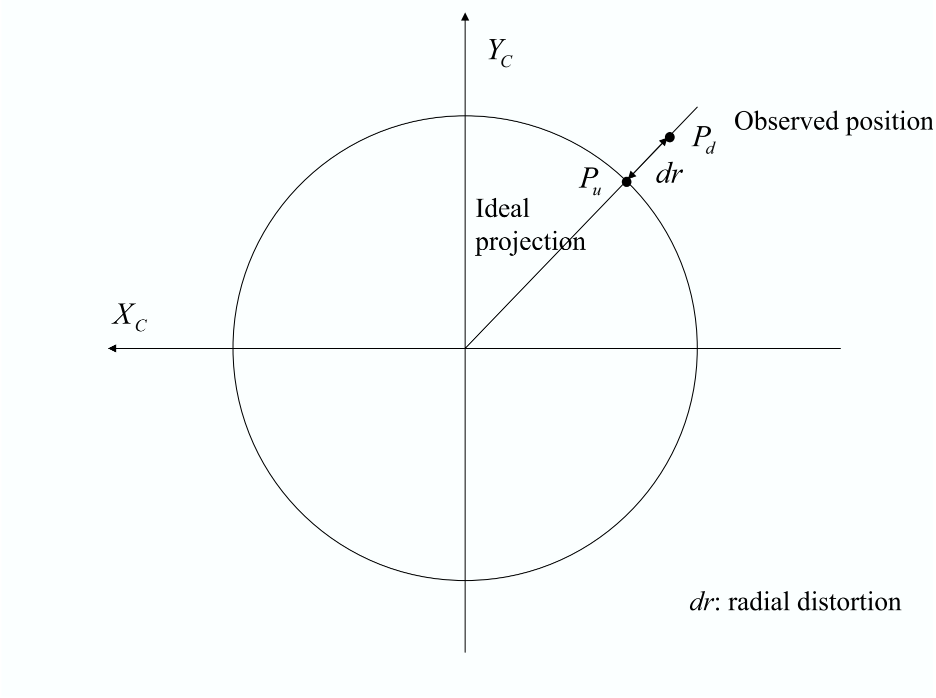


Figure 2.2: Radial distortion.

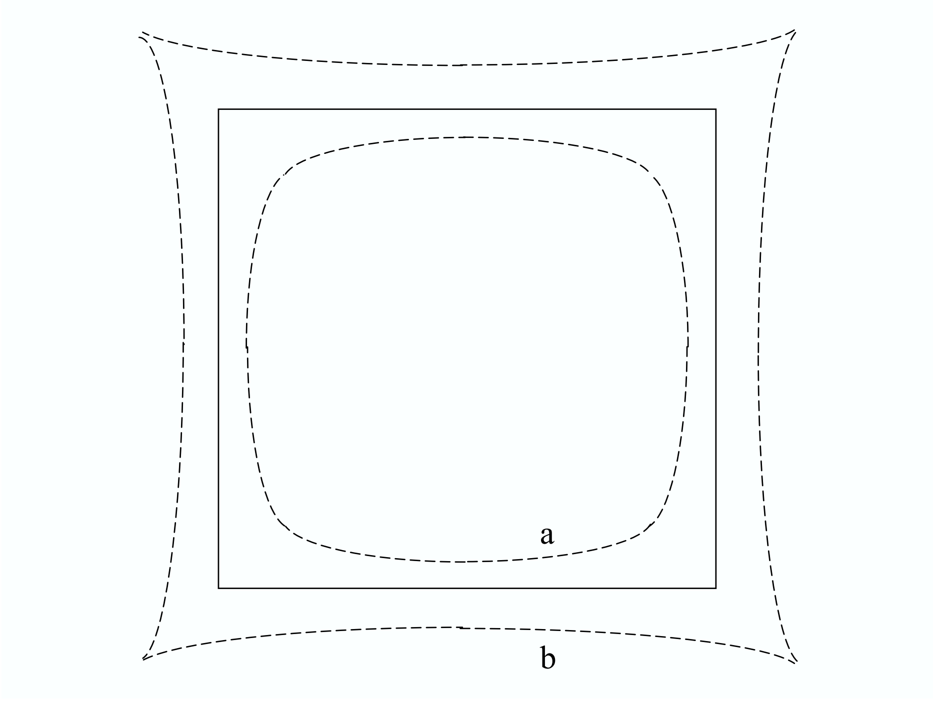


Figure 2.3: Radial distortion effect (a: negative, b: positive).

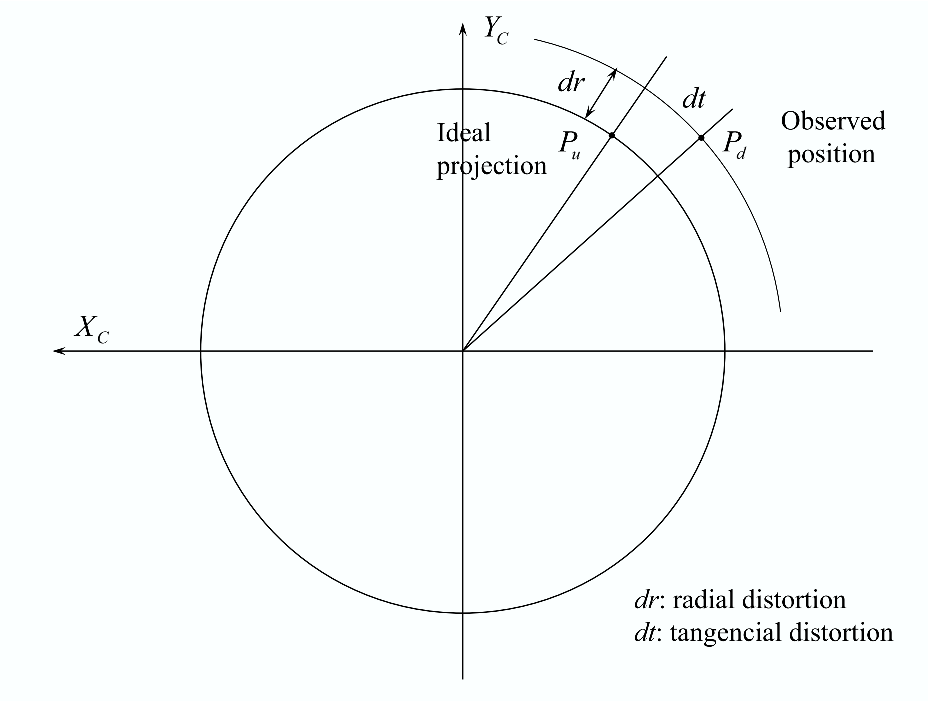


Figure 2.4: Radial and tangential distortion.

parameters will be different. Hence, equations (2.4) have to be substituted by equations (2.9).

$$^{C}X_d = {}^{C}X_u + \delta_x \qquad {}^{C}Y_d = {}^{C}Y_u + \delta_y \tag{2.9}$$

The *radial distortion* is modelled in the same manner Tsai proposed, except that Weng used the undistorted points.

$$\delta_{xr} = k_1 {}^{C}X_u \left({}^{C}X_u^{\,2} + {}^{C}Y_u^{\,2} \right) \qquad \delta_{yr} = k_1 {}^{C}Y_u \left({}^{C}X_u^{\,2} + {}^{C}Y_u^{\,2} \right) \tag{2.10}$$

The *decentering distortion* is due to the fact that the optical center of the lens is not correctly aligned with the center of the camera [Weng 92b]. This type of distortion introduces a radial and tangential distortion [Slama 80] (see Figure 2.4), which can be described by the following equations,

$$\begin{aligned} \delta_{xd} &= p_1 \left(3\,{}^{C}X_u^{\,2} + {}^{C}Y_u^{\,2} \right) + 2p_2 {}^{C}X_u {}^{C}Y_u \\ \delta_{yd} &= 2p_1 {}^{C}X_u {}^{C}Y_u + p_2 \left({}^{C}X_u^{\,2} + 3\,{}^{C}Y_u^{\,2} \right) \end{aligned} \tag{2.11}$$

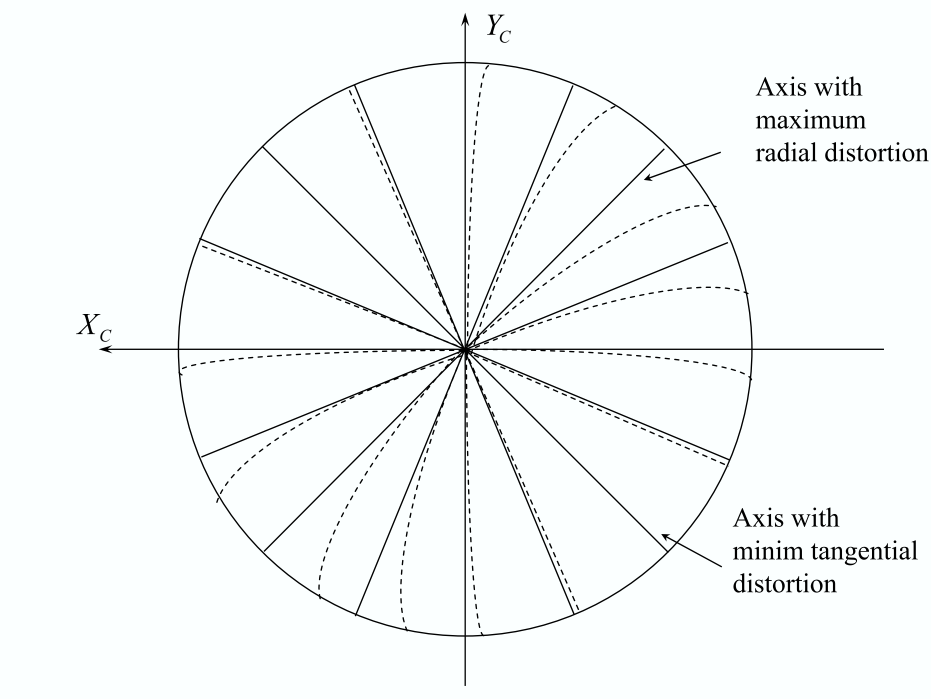


Figure 2.5: Radial and tangential distortion effect.

The *thin prism distortion* arises from imperfection in lens design and manufacturing as well as camera assembly. This type of distortion can be modelled by adding a thin prism to the optic system, causing radial and tangential distortions [Weng 92b]. This distortion is modelled by,

$$\delta_{xp} = s_1 \left({}^C X_u{}^2 + {}^C Y_u{}^2 \right) \qquad \delta_{yp} = s_2 \left({}^C X_u{}^2 + {}^C Y_u{}^2 \right) \tag{2.12}$$

By adding the three equations (2.7), (2.11) and (2.12), and carrying out the following variable replacement: $g_1 = s_1 + p_1$, $g_2 = s_2 + p_2$, $g_3 = 2p_1$ and $g_4 = 2p_2$, equations (2.13) are obtained,

$$\begin{aligned} \delta_x &= (g_1 + g_3)\, {}^C X_u{}^2 + g_4 {}^C X_u {}^C Y_u + g_1 {}^C Y_u{}^2 + k_1 {}^C X_u \left({}^C X_u{}^2 + {}^C Y_u{}^2 \right) \\ \delta_y &= g_2 {}^C X_u{}^2 + g_3 {}^C X_u {}^C Y_u + (g_2 + g_4)\, {}^C Y_u{}^2 + k_1 {}^C Y_u \left({}^C X_u{}^2 + {}^C Y_u{}^2 \right) \end{aligned} \tag{2.13}$$

2.2.4 Computer Image Frame

This final step deals with expressing the ${}^{C}P_d$ point with respect to the computer image plane in pixels $\{I\}$. This change of coordinates can be made in two different ways according to the camera models surveyed.

The camera models proposed by Faugeras-Toscani, Faugeras-Toscani with distortion and by Weng use the following equations to carry out such a transformation:

$$ {}^{I}X_d = -k_u {}^{C}X_d + u_0 \qquad {}^{I}Y_d = -k_v {}^{C}Y_d + v_0 \tag{2.14} $$

where: (k_u,k_v) are the parameters that transform from metric measures with respect to the camera coordinate system to pixels with respect to the computer image coordinate system; and $(u_0,\ v_0)$ are the components that define the projection of the focal point in the plane image in pixels, i.e. the principal point. They are used to determine the translation between both coordinate systems.

The camera model of Tsai proposed other equations to carry out the same transformation. These equations are the following,

$$ {}^{I}X_d = -s_x {d'_x}^{-1} {}^{C}X_d + u_0 \qquad {}^{I}Y_d = -{d_y}^{-1} {}^{C}Y_d + v_0 \tag{2.15} $$

where: $(u_0,\ v_0)$ are the components of the principal point in pixels; s_x is the image scale factor; $d'_x = d_x \frac{N_{cx}}{N_{fx}}$; d_x is the center to center distance between adjacent sensor elements in the X direction; d_y is the center to center distance between adjacent sensor elements in the Y direction; N_{cx} is the number of sensor elements in the X direction; and N_{fx} is the number of pixels in an image row as sampled by the computer.

2.3 Calibrating Methods

The calibrating method depends on the model used to approximate the behavior of the camera. The linear models, i.e. Hall and Faugeras-Toscani, use a least-squares technique to obtain the parameters of the model. However, non-linear calibrating methods, i.e. Faugeras-Toscani with distortion, Tsai and Weng, use a two-stage technique. As a first stage, they carry out a linear approximation with the aim of obtaining an initial guess and then a further iterative algorithm is used to optimize the parameters. In this section, each calibrating method is explained detailing the equations and the algorithm used to calibrate the camera parameters.

2.3.1 The Method of Hall

The method used to calibrate the model of Hall is based on expressing equation (2.1) in the following form,

$$\begin{aligned} {}^{I}X_u &= \frac{A_{11}{}^{W}X_w + A_{12}{}^{W}Y_w + A_{13}{}^{W}Z_w + A_{14}}{A_{31}{}^{W}X_w + A_{32}{}^{W}Y_w + A_{33}{}^{W}Z_w + A_{34}} \\ {}^{I}Y_u &= \frac{A_{21}{}^{W}X_w + A_{22}{}^{W}Y_w + A_{23}{}^{W}Z_w + A_{24}}{A_{31}{}^{W}X_w + A_{32}{}^{W}Y_w + A_{33}{}^{W}Z_w + A_{34}} \end{aligned} \tag{2.16}$$

By arranging the variables, the following expressions are obtained,

$$\begin{aligned} 0 &= A_{11}{}^{W}X_w - A_{31}{}^{I}X_u{}^{W}X_w + A_{12}{}^{W}Y_w \\ &\quad -A_{32}{}^{I}X_u{}^{W}Y_w + A_{13}{}^{W}Z_w - A_{33}{}^{I}X_u{}^{W}Z_w + A_{14} - A_{34}{}^{I}X_u \\ 0 &= A_{21}{}^{W}X_w - A_{31}{}^{I}Y_u{}^{W}X_w + A_{22}{}^{W}Y_w \\ &\quad -A_{32}{}^{I}Y_u{}^{W}Y_w + A_{23}{}^{W}Z_w - A_{33}{}^{I}Y_u{}^{W}Z_w + A_{24} - A_{34}{}^{I}Y_u \end{aligned} \tag{2.17}$$

Finally, the unknowns A_{ij} are arranged in a 12-parameter vector (A), obtaining the following equation:

$$\mathbf{Q}\,A = 0 \tag{2.18}$$

where A is the vector of 12 unknowns of the equation (2.19). Q is a matrix of $2n \times 12$ where n is the number of pair points used to calibrate the camera. A pair of points is formed by a 3D point expressed with respect to the world coordinate system $\{W\}$ and its 2D projection expressed in pixels with respect to coordinate system $\{I\}$.

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} & A_{14} & A_{21} & A_{22} & A_{23} & A_{24} & A_{31} & A_{32} & A_{33} & A_{34} \end{pmatrix}^{\mathrm{T}} \tag{2.19}$$

Every pair of points adds to the Q matrix the two following rows,

$$Q_{2i-1} = \begin{pmatrix} {}^{W}X_{ui} \\ {}^{W}Y_{ui} \\ {}^{W}Z_{ui} \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ -{}^{I}X_{ui}{}^{W}X_{wi} \\ -{}^{I}X_{ui}{}^{W}Y_{wi} \\ -{}^{I}X_{ui}{}^{W}Z_{wi} \\ -{}^{I}X_{ui} \end{pmatrix}^{\mathrm{T}} \qquad Q_{2i} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ {}^{W}X_{ui} \\ {}^{W}Y_{ui} \\ {}^{W}Z_{ui} \\ 1 \\ -{}^{I}Y_{ui}{}^{W}X_{wi} \\ -{}^{I}Y_{ui}{}^{W}Y_{wi} \\ -{}^{I}Y_{ui}{}^{W}Z_{wi} \\ -{}^{I}Y_{ui} \end{pmatrix}^{\mathrm{T}} \tag{2.20}$$

Consider then that the 3D position of a set of n calibrating points and their corresponding 2D projection in the image are known (n should be bigger or equal to 6). Moreover, consider without loss of generality that $A_{34} = 1$. This approximation can be assumed since the transformation matrix is defined up to a scale factor [Faugeras 93]. Then, all the elements of the A vector can be obtained by using a linear least-squares technique as the pseudo-inverse [Hall 82]. With the aim of applying the pseudo-inverse, it becomes necessary to modify equation (2.18) considering that $A_{34} = 1$, obtaining:

$$\mathbf{Q}' A' = B' \tag{2.21}$$

where,

$$A' = \begin{pmatrix} A_{11} & A_{12} & A_{13} & A_{14} & A_{21} & A_{22} & A_{23} & A_{24} & A_{31} & A_{32} & A_{33} \end{pmatrix}^{\mathrm{T}} \tag{2.22}$$

and,

$$Q'_{2i-1} = \begin{pmatrix} {}^{W}X_{ui} \\ {}^{W}Y_{ui} \\ {}^{W}Z_{ui} \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ -{}^{I}X_{ui}{}^{W}X_{wi} \\ -{}^{I}X_{ui}{}^{W}Y_{wi} \\ -{}^{I}X_{ui}{}^{W}Z_{wi} \end{pmatrix}^{\mathrm{T}} \qquad Q'_{2i} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ {}^{W}X_{ui} \\ {}^{W}Y_{ui} \\ {}^{W}Z_{ui} \\ 1 \\ -{}^{I}Y_{ui}{}^{W}X_{wi} \\ -{}^{I}Y_{ui}{}^{W}Y_{wi} \\ -{}^{I}Y_{ui}{}^{W}Z_{wi} \end{pmatrix}^{\mathrm{T}} \tag{2.23}$$

$$B'_{2i-1} = \left({}^{I}X_{ui}\right) \qquad B'_{2i} = \left({}^{I}Y_{ui}\right) \tag{2.24}$$

Finally, the vector of unknowns (A) is computed by applying the pseudo-inverse shown in the following equation (2.25).

$$A' = \left(\mathbf{Q}'^{\mathrm{T}}\mathbf{Q}'\right)^{-1}\mathbf{Q}'^{\mathrm{T}}B' \tag{2.25}$$

2.3.2 The Method of Faugeras

In order to calibrate the complete model of the camera proposed by Faugeras and Toscani, it is necessary to combine equations (2.2), (2.3), (2.4), (2.5) and (2.14), obtaining (2.26).

$$\begin{aligned} {}^{I}X_u &= -k_u f \frac{r_{11}{}^{W}X_w + r_{12}{}^{W}Y_w + r_{13}{}^{W}Z_w + t_x}{r_{31}{}^{W}X_w + r_{32}{}^{W}Y_w + r_{33}{}^{W}Z_w + t_z} + u_0 \\ {}^{I}Y_u &= -k_v f \frac{r_{21}{}^{W}X_w + r_{22}{}^{W}Y_w + r_{23}{}^{W}Z_w + t_y}{r_{31}{}^{W}X_w + r_{32}{}^{W}Y_w + r_{33}{}^{W}Z_w + t_z} + v_0 \end{aligned} \tag{2.26}$$

Note that equations (2.26) can be expressed in a matricial form in the following manner,

$$\begin{pmatrix} s^I X_d \\ s^I Y_d \\ s \end{pmatrix} = \begin{pmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} {}^W X_w \\ {}^W Y_w \\ {}^W Z_w \\ 1 \end{pmatrix} \tag{2.27}$$

where $\alpha_u = -f k_u$ and $\alpha_v = -f k_v$. Then, by computing the product of both matrices, the transformation matrix A is obtained.

$$\begin{pmatrix} s^I X_d \\ s^I Y_d \\ s \end{pmatrix} = \mathbf{A} \begin{pmatrix} {}^W X_w \\ {}^W Y_w \\ {}^W Z_w \\ 1 \end{pmatrix} \tag{2.28}$$

$$\mathbf{A} = \begin{pmatrix} \alpha_u r_1 + u_0 r_3 & \alpha_u t_x + u_0 t_z \\ \alpha_v r_2 + v_0 r_3 & \alpha_v t_y + v_0 t_z \\ r_3 & t_z \end{pmatrix} \tag{2.29}$$

The camera parameters can be extracted from the symbolic matrix (A) by equalling it to the numeric matrix obtained by calibrating the camera with the technique of Hall. Note that the orientation of the vectors r_i must be orthogonal and that it is also known that the dot product between two vectors follows the following criteria:

$$\begin{aligned} r_i {r_j}^{\mathrm{T}} &= 0 \quad i \neq j \\ r_i {r_j}^{\mathrm{T}} &= 1 \quad i = j \end{aligned} \tag{2.30}$$

Using these relationships, the four intrinsic parameters (α_u, α_v, u_0, v_0) and the six extrinsic ones (r_1, r_2, r_3, t_x, t_y, t_z) can be extracted from equation (2.29) in the following manner,

$$\begin{aligned} u_0 &= A_1 {A_3}^{\mathrm{T}} & v_0 &= A_2 {A_3}^{\mathrm{T}} \\ \alpha_u &= -\left(A_1 {A_1}^{\mathrm{T}} - u_0^2\right)^{1/2} & \alpha_v &= -\left(A_2 {A_2}^{\mathrm{T}} - v_0^2\right)^{1/2} \\ r_1 &= \tfrac{1}{\alpha_u}\left(A_1 - u_0 A_3\right) & t_x &= \tfrac{1}{\alpha_u}\left(A_{14} - u_0 A_{34}\right) \\ r_2 &= \tfrac{1}{\alpha_v}\left(A_2 - v_0 A_3\right) & t_y &= \tfrac{1}{\alpha_v}\left(A_{24} - v_0 A_{34}\right) \\ r_3 &= A_3 & t_z &= A_{34} \end{aligned} \tag{2.31}$$

where the numerical matrix **A** is:

$$\mathbf{A} = \begin{pmatrix} A_1 & A_{14} \\ A_2 & A_{24} \\ A_3 & A_{34} \end{pmatrix} \tag{2.32}$$

However, before estimating the camera parameters, the A matrix has to be calculated. Faugeras proposed a slightly different method of estimating A from the one proposed by Hall. Hence, the terms of equation (2.1) have been rearranged in the following way,

$$\begin{array}{rcl} A_1{}^W P_w + A_{14} - {}^I X_u \left(A_3{}^W P_w + A_{34}\right) & = & 0 \\ A_2{}^W P_w + A_{24} - {}^I Y_u \left(A_3{}^W P_w + A_{34}\right) & = & 0 \end{array} \tag{2.33}$$

Both equations are then factorized with respect to the unknowns, obtaining,

$$\begin{array}{rcl} {}^I X_u & = & \frac{A_1}{A_{34}}{}^W P_w + \frac{A_{14}}{A_{34}} - \frac{A_3}{A_{34}}{}^W P_w{}^I X_u \\ \\ {}^I Y_u & = & \frac{A_2}{A_{34}}{}^W P_w + \frac{A_{24}}{A_{34}} - \frac{A_3}{A_{34}}{}^W P_w{}^I Y_u \end{array} \tag{2.34}$$

At this point, a set of 5 parameters is considered $X = (T_1, T_2, T_3, C_1, C_2)^{\mathrm{T}}$, which are $T_1 = \frac{A_1}{A_{34}}$, $T_2 = \frac{A_3}{A_{34}}$, $T_3 = \frac{A_2}{A_{34}}$, $C_1 = \frac{A_{14}}{A_{34}}$ and $C_2 = \frac{A_{24}}{A_{34}}$.

$$\begin{array}{rcl} {}^I X_u & = & T_1{}^W P_w + C_1 - T_2{}^W P_w{}^I X_u \\ {}^I Y_u & = & T_3{}^W P_w + C_2 - T_2{}^W P_w{}^I Y_u \end{array} \tag{2.35}$$

Then, the value of the vector X is obtained by using a linear least-squares technique.

$$B = \mathbf{Q}\, X \tag{2.36}$$

where,

$$\mathbf{Q} = \begin{pmatrix} & & \cdots & & \\ {}^W P_{wi}^{\mathrm{T}} & -{}^I X_{ui}{}^W P_{wi}^{\mathrm{T}} & 0_{1x3} & 1 & 0 \\ 0_{1x3} & -{}^I Y_{ui}{}^W P_{wi}^{\mathrm{T}} & {}^W P_{wi}^{\mathrm{T}} & 0 & 1 \\ & & \cdots & & \end{pmatrix} \qquad B = \begin{pmatrix} \cdots \\ {}^I X_{ui} \\ {}^I Y_{ui} \\ \cdots \end{pmatrix} \tag{2.37}$$

Hence, vector X is computed using equation (2.36).

$$X = \left(\mathbf{Q}^{\mathrm{T}}\mathbf{Q}\right)^{-1}\mathbf{Q}^{\mathrm{T}}B \tag{2.38}$$

Finally, the camera parameters are extracted from X by using equation (2.28).

$$\begin{array}{lllllll} T_1 & = & \frac{r_3}{t_z}u_0 + \frac{r_1}{t_z}\alpha_u & & C_1 & = & u_0 + \frac{t_x}{t_z}\alpha_u \\ T_2 & = & \frac{r_3}{t_z} & & & & \\ T_3 & = & \frac{r_3}{t_z}v_0 + \frac{r_2}{t_z}\alpha_v & & C_2 & = & v_0 + \frac{t_y}{t_z}\alpha_v \end{array} \tag{2.39}$$

At this point, it has to be considered that the norm of the three orientation vectors r_i is equal to unity by definition. By using equations (2.39), the parameter t_z can then be computed. Hence, considering $r_3 = 1$,

$$t_z = \frac{1}{\|T_2\|} \tag{2.40}$$

The rest of the parameters can be obtained using the properties of the dot product and the cross product between vectors, which are,

$$v_1 v_2 \;=\; \|v_1\|\,\|v_2\|\cos\alpha \qquad\qquad v_1 \wedge v_2 \;=\; \|v_1\|\,\|v_2\|\sin\alpha \tag{2.41}$$

so that,

$$\begin{array}{lllllllllllll} r_i {r_j}^{\mathrm{T}} & = & 0 & i & \neq & j & \qquad & r_i \wedge r_j & = & 1 & i & \neq & j \\ r_i {r_j}^{\mathrm{T}} & = & 1 & i & = & j & & r_i \wedge r_j & = & 0 & i & = & j \end{array} \tag{2.42}$$

The intrinsic parameters can then be obtained in the following way,

$$\begin{array}{llllll} u_0 & = & \frac{T_1 {T_2}^{\mathrm{T}}}{\|T_2\|^2} & v_0 & = & \frac{T_1 {T_3}^{\mathrm{T}}}{\|T_2\|^2} \\ \alpha_u & = & -\frac{\left\|{T_1}^{\mathrm{T}} \wedge {T_2}^{\mathrm{T}}\right\|}{\|T_2\|^2} & \alpha_v & = & -\frac{\left\|{T_2}^{\mathrm{T}} \wedge {T_3}^{\mathrm{T}}\right\|}{\|T_2\|^2} \end{array} \tag{2.43}$$

Moreover, the extrinsic parameters which model the orientation are the following,

$$\begin{aligned} r_1 &= -\frac{\|T_2\|}{\|{T_1}^{\mathrm{T}} \wedge {T_2}^{\mathrm{T}}\|}\left(T_1 - \frac{T_1{T_2}^{\mathrm{T}}}{\|T_2\|^2}T_2\right) \\ r_2 &= -\frac{\|T_2\|}{\|{T_2}^{\mathrm{T}} \wedge {T_3}^{\mathrm{T}}\|}\left(T_3 - \frac{T_2{T_3}^{\mathrm{T}}}{\|T_2\|^2}T_2\right) \\ r_3 &= \frac{T_2}{\|T_2\|} \end{aligned} \tag{2.44}$$

Finally, the extrinsic parameters that model the translation are also obtained from (2.39).

$$\begin{aligned} t_x &= -\frac{\|T_2\|}{\|{T_1}^{\mathrm{T}} \wedge {T_2}^{\mathrm{T}}\|}\left(C_1 - \frac{T_1{T_2}^{\mathrm{T}}}{\|T_2\|^2}\right) \\ t_y &= -\frac{\|T_2\|}{\|{T_2}^{\mathrm{T}} \wedge {T_3}^{\mathrm{T}}\|}\left(C_2 - \frac{T_2{T_3}^{\mathrm{T}}}{\|T_2\|^2}\right) \\ t_z &= \frac{1}{\|T_2\|} \end{aligned} \tag{2.45}$$

By using the r_i vectors in equations (2.44), the rotation matrix ${}^{C}R_W$ is directly obtained. The three angles α, β and γ can then be computed by equalling the symbolic rotation matrix to the numeric matrix obtained by calibration. At this point, all the parameters of the linear model of Faugeras are obtained. These parameters determine the relationship between the 3D object points with their 2D projections, as shown in equation (2.28). However, the model of Faugeras can be more accurate if radial lens distortion is included.

2.3.3 The Method of Faugeras with Radial Distortion

When a bright accuracy is necessary, the linear method of Faugeras becomes useless. However, it can be easily modified by including the radial lens distortion as it has been shown in section 2.2.3. Then, the equations become non-linear, and the linear least-squares technique has to be replaced by an iterative algorithm.

Note that by combining equations (2.2), (2.3), (2.4), (2.6) and (2.7), the equations (2.46) are obtained.

$$\begin{aligned}
{}^{C}X_d + {}^{C}X_d k_1 r^2 &= f\frac{r_{11}{}^{W}X_w + r_{12}{}^{W}Y_w + r_{13}{}^{W}Z_w + t_x}{r_{31}{}^{W}X_w + r_{32}{}^{W}Y_w + r_{33}{}^{W}Z_w + t_z} \\
{}^{C}Y_d + {}^{C}Y_d k_1 r^2 &= f\frac{r_{21}{}^{W}X_w + r_{22}{}^{W}Y_w + r_{23}{}^{W}Z_w + t_y}{r_{31}{}^{W}X_w + r_{32}{}^{W}Y_w + r_{33}{}^{W}Z_w + t_z} \\
r &= \sqrt{{}^{C}X_d{}^2 + {}^{C}Y_d{}^2}
\end{aligned} \tag{2.46}$$

Moreover, equations (2.14) have to be used to transform from metric coordinates to pixels. Then, equation (2.47) defines the vector of unknowns which can be computed by using an iterative method as, for instance, the method of Newton-Raphson or Levenberg-Marquardt, among others [Stoer 80].

$$x = (\alpha, \beta, \gamma, t_x, t_y, t_z, k_u, k_v, u_0, v_0, k_1)^{\mathrm{T}} \tag{2.47}$$

For example, the general method of Newton-Raphson minimizes the following equation,

$$G(x_k) \approx G(x_{k-1}) + \mathbf{J}\Delta x_k \tag{2.48}$$

where x is the vector of unknowns, $G(x)$ is the minimization function, $G(x_k)$ is a value close to the solution, and $\mathbf{J}$ represents the jacobian matrix of the function $G(x)$. With the aim of finding a solution of Δx_k, it is necessary to equal $G(x_k)$ to zero.

$$G(x_k) = 0 \tag{2.49}$$

Note that one of the problems of convergence in iterative algorithms is the initial guess. However, an initial guess can be obtained by calibrating the linear method of Faugeras-Toscani without including lens distortion, and assuming $k_1 = 0$. Moreover, the difference between the initial value and the estimated parameters will be the error of the function. For each iteration it is necessary to compute the value of Δx_k to obtain the new value of x.

$$\mathbf{J}\Delta x_k = -G(x_{k-1}) \tag{2.50}$$

Then, arranging equations (2.46) and (2.14), the functions $U(x)$ and $V(x)$ are given.

$$\begin{aligned} U\left(x\right) &= f\frac{r_{11}{}^{W}X_w + r_{12}{}^{W}Y_w + r_{13}{}^{W}Z_w + t_x}{r_{31}{}^{W}X_w + r_{32}{}^{W}Y_w + r_{33}{}^{W}Z_w + t_z} - \frac{\left({}^{I}X_d - u_0\right)}{-k_u} \\ &\quad -k_1\left(\left(\frac{\left({}^{I}X_d - u_0\right)}{-k_u}\right)^2 + \left(\frac{\left({}^{I}Y_d - v_0\right)}{-k_v}\right)^2\right)\frac{\left({}^{I}X_d - u_0\right)}{-k_u} \\ V\left(x\right) &= f\frac{r_{21}{}^{W}X_w + r_{22}{}^{W}Y_w + r_{23}{}^{W}Z_w + t_y}{r_{31}{}^{W}X_w + r_{32}{}^{W}Y_w + r_{33}{}^{W}Z_w + t_z} - \frac{\left({}^{I}Y_d - v_0\right)}{-k_v} \\ &\quad -k_1\left(\left(\frac{\left({}^{I}X_d - u_0\right)}{-k_u}\right)^2 + \left(\frac{\left({}^{I}Y_d - v_0\right)}{-k_v}\right)^2\right)\frac{\left({}^{I}Y_d - v_0\right)}{-k_v} \end{aligned} \tag{2.51}$$

Next, with the aim of solving the system, it is necessary to apply equations (2.51) to the n calibrating points. However, in order to apply equation (2.50), it is necessary to get the symbolic function $G(x)$ and its partial derivative matrix $\mathbf{J}$, as it is shown in the following equations,

$$G(x_{k-1}) = \begin{pmatrix} U_1\left(x_{k-1}\right) \\ V_1\left(x_{k-1}\right) \\ \vdots \\ V_n\left(x_{k-1}\right) \end{pmatrix} \tag{2.52}$$

$$\mathbf{J} = \begin{pmatrix} \frac{\partial U_1(x_{k-1})}{\partial \alpha} & \frac{\partial U_1(x_{k-1})}{\partial \beta} & \cdots & \frac{\partial U_1(x_{k-1})}{\partial k_1} \\ \frac{\partial V_1(x_{k-1})}{\partial \alpha} & \frac{\partial V_1(x_{k-1})}{\partial \beta} & \cdots & \frac{\partial V_1(x_{k-1})}{\partial k_1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial V_n(x_{k-1})}{\partial \alpha} & \frac{\partial V_n(x_{k-1})}{\partial \beta} & \cdots & \frac{\partial V_n(x_{k-1})}{\partial k_1} \end{pmatrix} \tag{2.53}$$

Finally, the parameters of the model are obtained by applying the pseudo-inverse of equations (2.54) in each iteration. The more iterations done, the higher the accuracy obtained until convergence is achieved.

$$\begin{aligned} \Delta x_k &= -\left(\mathbf{J}^{\mathrm{T}}\mathbf{J}\right)^{-1}\mathbf{J}^{\mathrm{T}}G\left(x_{k-1}\right) \\ x_k &= x_{k-1} + \Delta x_k \end{aligned} \tag{2.54}$$

2.3.4 The Method of Tsai

The non-linear method of Faugeras was based on fixing the initial guess without considering lens distortion. Moreover, a large number of iterations are usually necessary to obtain an accurate value of the camera parameters. Besides, the method of Tsai [Tsai 87] also models the radial lens distortion but assumes that there are some parameters of the camera which are provided by manufacturers. This fact reduces the number of calibrating parameters in the first step where an initial guess is estimated. Moreover, although all the parameters are iteratively optimized in the last step, the number of iterations is considerably reduced by using the calibrating algorithm proposed by Tsai.

Firstly, by combining equations (2.2), (2.3), (2.4), (2.6), and (2.7), the equations (2.46) are obtained. Note that at this point model of Tsai is equivalent to the previous model of Faugeras with distortion (2.46). Once ${}^{C}X'_d$ and ${}^{C}Y'_d$ are obtained in metric coordinates by using equation (2.15), they can be expressed in pixels (${}^{I}X_d$ and ${}^{I}Y_d$) and the following equations are obtained.

$$ {}^{C}X'_{di} = -\left({}^{I}X_{di} - u_0\right) d'_x \qquad {}^{C}Y'_{di} = -\left({}^{I}Y_{di} - v_0\right) d_y \tag{2.55} $$

where,

$$ {}^{C}X'_{di} = {}^{C}X_{di} s_x \qquad {}^{C}Y'_{di} = {}^{C}Y_{di} \tag{2.56} $$

It is necessary therefore to find a relationship between the image point P_d (in metric coordinates) with respect to the object point P_w. Figure 2.6 shows how the radial distortion affects the camera model. It can be observed that the segment $\overline{O_R P_d}$ is parallel to the segment $\overline{P_{oz} P_w}$. Considering this constraint, the following relationship is established,

$$ \overline{O_R P_d} // \overline{P_{oz} P_w} \Rightarrow \overline{O_R P_d} \times \overline{P_{oz} P_w} = 0 \tag{2.57} $$

By using equation (2.57), the following equations are obtained.

$$ \begin{aligned} \overline{O_R P_d} \times \overline{P_{oz} P_w} &= 0 && (2.58) \\ \left({}^{C}X_d, {}^{C}Y_d\right) \times \left({}^{C}X_w, {}^{C}Y_w\right) &= 0 && (2.59) \\ {}^{C}X_d {}^{C}Y_w - {}^{C}Y_d {}^{C}X_w &= 0 && (2.60) \end{aligned} $$

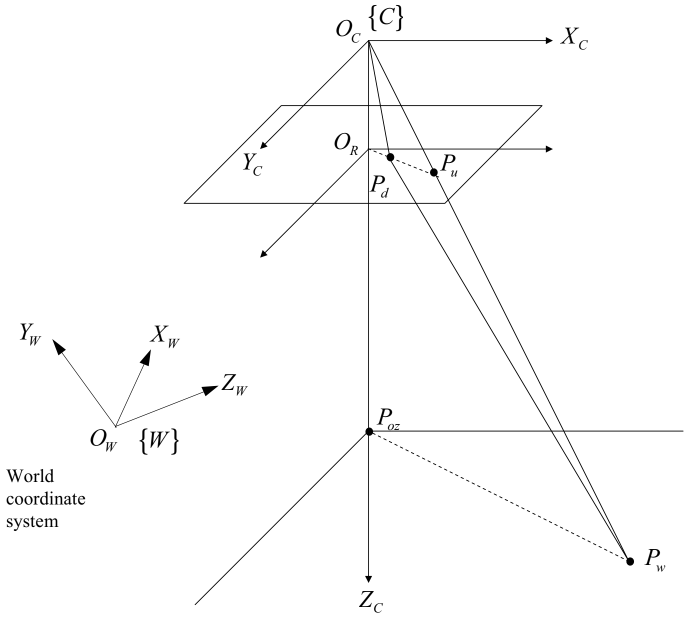


Figure 2.6: Illustration of the radial alignment constraint [Tsai 87].

Equation (2.60) can be arranged expressing the object point P_w with respect to the world coordinate system, instead of expressing it with respect to the camera coordinate system.

$$\begin{aligned} &{}^{C}X_d \left(r_{21}{}^{W}X_w + r_{22}{}^{W}Y_w + r_{23}{}^{W}Z_w + t_y \right) = \\ &\quad {}^{C}Y_d \left(r_{11}{}^{W}X_w + r_{12}{}^{W}Y_w + r_{13}{}^{W}Z_w + t_x \right) \end{aligned} \qquad (2.61)$$

Operating equation (2.61) and arranging the terms,

$$\begin{aligned} {}^{C}X_d &= {}^{C}Y_d{}^{W}X_w\frac{r_{11}}{t_y} + {}^{C}Y_d{}^{W}Y_w\frac{r_{12}}{t_y} + {}^{C}Y_d{}^{W}Z_w\frac{r_{13}}{t_y} + {}^{C}Y_d\frac{t_x}{t_y} \\ &\quad -{}^{C}X_d{}^{W}X_w\frac{r_{21}}{t_y} - {}^{C}X_d{}^{W}Y_w\frac{r_{22}}{t_y} + {}^{C}X_d{}^{W}Z_w\frac{r_{23}}{t_y} \end{aligned} \qquad (2.62)$$

In order to compute equation (2.62) for the n points obtained from equations (2.55), it is necessary to combine equation (2.62) with the equations (2.56), obtaining,

$$\begin{aligned} {}^{C}X'_{di} &= {}^{C}Y'_{di}\,{}^{W}X_{wi}\frac{s_x r_{11}}{t_y} + {}^{C}Y'_{di}\,{}^{W}Y_{wi}\frac{s_x r_{12}}{t_y} + {}^{C}Y'_{di}\,{}^{W}Z_{wi}\frac{s_x r_{13}}{t_y} + {}^{C}Y'_{di}\frac{s_x t_x}{t_y} \\ &\quad -{}^{C}X'_{di}\,{}^{W}X_{wi}\frac{r_{21}}{t_y} - {}^{C}X'_{di}\,{}^{W}Y_{wi}\frac{r_{22}}{t_y} + {}^{C}X'_{di}\,{}^{W}Z_{wi}\frac{r_{23}}{t_y} \end{aligned} \tag{2.63}$$

At this point, a system with n equations and 7 unknowns is obtained, which can be expressed in the following form,

$$\begin{pmatrix} {}^{C}Y'_{di}\,{}^{W}X_{wi} \\ {}^{C}Y'_{di}\,{}^{W}Y_{wi} \\ {}^{C}Y'_{di}\,{}^{W}Z_{wi} \\ {}^{C}Y'_{di} \\ -{}^{C}X'_{di}\,{}^{W}X_{wi} \\ -{}^{C}X'_{di}\,{}^{W}Y_{wi} \\ -{}^{C}X'_{di}\,{}^{W}Z_{wi} \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} t_y^{-1}s_x r_{11} \\ t_y^{-1}s_x r_{12} \\ t_y^{-1}s_x r_{13} \\ t_y^{-1}s_x t_x \\ t_y^{-1}s_x r_{21} \\ t_y^{-1}s_x r_{22} \\ t_y^{-1}s_x r_{23} \end{pmatrix} = {}^{C}X'_{di} \tag{2.64}$$

In order to simplify the notation, the 7 unknown components of the vector renamed.

$$\begin{gathered} a_1 = t_y^{-1}s_x r_{11} \quad a_5 = t_y^{-1}r_{21} \\ a_2 = t_y^{-1}s_x r_{12} \quad a_6 = t_y^{-1}r_{22} \\ a_3 = t_y^{-1}s_x r_{13} \quad a_7 = t_y^{-1}r_{23} \\ a_4 = t_y^{-1}s_x t_x \end{gathered} \tag{2.65}$$

Note that the a_i components can be easily computed by using a least-squares technique. Therefore, the point of interest is to extract the calibrating parameters of the camera from these a_i components. First t_y can be obtained by using equations (2.65) in the following manner,

$$t_y = \frac{\|r_2\|}{\|a_{5,6,7}\|} \tag{2.66}$$

and equation (2.66) is simplified because the norm of the vector r_2 is equal to the unity, obtaining the parameter t_y.

$$|t_y| = \frac{1}{\sqrt{{a_5}^2 + {a_6}^2 + {a_7}^2}} \tag{2.67}$$

However, equation (2.67) is insufficient since it does not provide the sign of the t_y component. In order to determine this sign, a point $({}^{I}X_d, {}^{I}Y_d)$ located at the periphery of the image, far from the center, is taken from the set of test points (its corresponding 3D point is also kept). It is then supposed that the t_y sign is positive, and the following equations are computed.

$$\begin{array}{cc} r_{11} = a_1 t_y / s_x & r_{21} = a_5 t_y \\ r_{12} = a_2 t_y / s_x & r_{22} = a_6 t_y \\ r_{13} = a_3 t_y / s_x & r_{23} = a_7 t_y \\ \multicolumn{2}{c}{t_x = a_4 t_y} \end{array} \tag{2.68}$$

By using the corresponding 3D point $({}^{W}X_w, {}^{W}Y_w, {}^{W}Z_w)$, the linear projection of this 3D point on the image plane (without considering lens distortion) can be computed by using equations (2.69).

$$\begin{aligned} {}^{C}X_u &= r_{11}{}^{W}X_w + r_{12}{}^{W}Y_w + r_{13}{}^{W}Z_w + t_x \\ {}^{C}Y_u &= r_{21}{}^{W}X_w + r_{22}{}^{W}Y_w + r_{23}{}^{W}Z_w + t_y \end{aligned} \tag{2.69}$$

At this point the t_y sign can be verified. If both components of the point $({}^{C}X_u, {}^{C}Y_u)$ have a sign equal to the components of the point $({}^{I}X_d, {}^{I}Y_d)$, it means that the t_y sign was correctly chosen as positive. Otherwise, it has to be considered negative.

The second parameter to be extracted is the scale factor (s_x). Note that by arranging equations (2.65), the following equation is obtained,

$$s_x = \frac{\|a_{1,2,3}\|\, t_y}{\|r_1\|} \tag{2.70}$$

where it is known that the norm of r_1 is the unity and the scale factor is always positive. Then, s_x is obtained by using equation (2.71).

$$s_x = \sqrt{{a_1}^2 + {a_2}^2 + {a_3}^2}\, |t_y| \tag{2.71}$$

Furthermore, the 2D points, with respect to the camera coordinate system $({}^{C}X_d, {}^{C}Y_d)$, can be computed from the same point with respect to the image coordinate system, that is $({}^{I}X_d, {}^{I}Y_d)$, by using equations (2.56). Moreover, by using equations (2.68) the r_1 and r_2 vectors of the rotation matrix ${}^{C}\mathbf{R}_W$, and the first element of the trans-

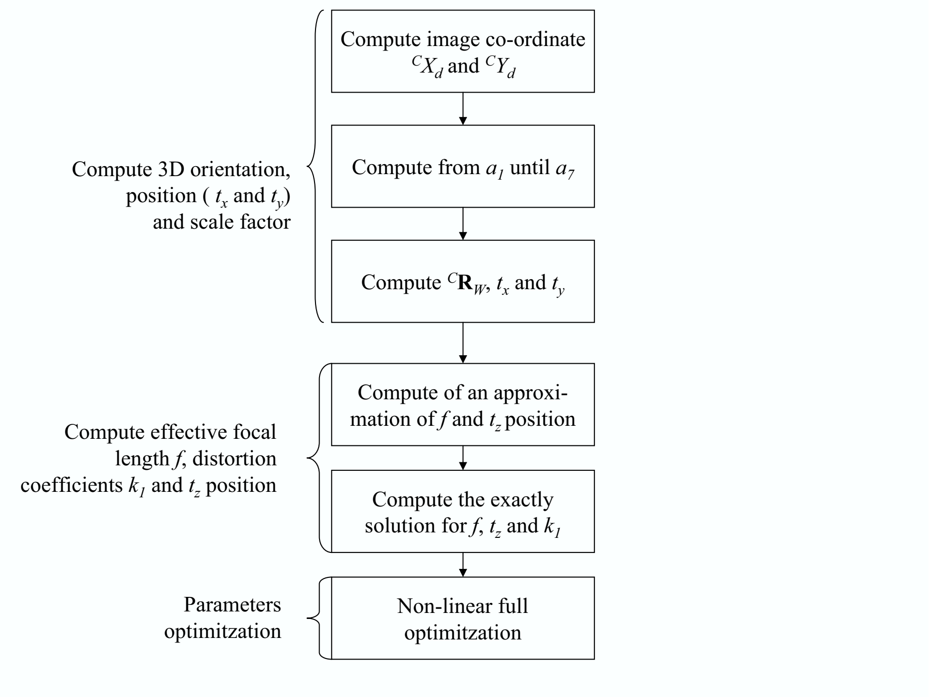


Figure 2.7: Flowchart of the method of Tsai.

lation vector $^{C}T_W$, i.e. t_x, can be calculated. Finally, the third orientation vector (r_3) can be computed by a cross product between r_1 and r_2 because of the property of orthogonality, (note also that the determinant of any rotation matrix is the unity, i.e. $|^{C}R_W| = 1$). At this point, the first three steps of the method of Tsai are completed, see Figure 2.7.

However, the following parameters are still unknown: the focal distance (f), the radial lens distortion coefficient (k_1), and the translation of the camera with respect to the Z axis (t_z). In order to compute these three parameters, a linear approximation is first used without considering the k_1 parameter. The linear approximation is shown in equation (2.72), which was obtained from equations (2.46).

$$\begin{pmatrix} r_{21}{}^{W}X_{wi} + r_{22}{}^{W}Y_{wi} + r_{23}{}^{W}Z_{wi} + t_y & -{}^{C}Y_d \end{pmatrix} \begin{pmatrix} f \\ t_z \end{pmatrix} = \left(r_{31}{}^{W}X_{wi} + r_{32}{}^{W}Y_{wi} + r_{33}{}^{W}Z_{wi}\right) {}^{C}Y_d \tag{2.72}$$

Equation (2.72) has now been applied to the whole set of test points, obtaining a system of n equations and two unknowns. The linear approximation of both unknowns, f and t_z, is obtained by using a pseudo-inverse. However, in order to calculate a better approximation including the k_1 parameter, it is necessary to iterate equations (2.46) by using an optimization method considering the linear method with $k_1 = 0$ as an initial solution.

Finally, all the parameters are optimized iteratively with the aim of obtaining an accurate solution. The entire process is explained in Figure 2.7.

2.3.5 The Method of Weng

The method of Tsai is based on modelling radial lens distortion. The accuracy obtained by Tsai is sufficient for most applications. However, in some cases where the camera lens needs to be accurately modelled, a simple radial approximation is not sufficient. In such situations, Weng [Weng 92b] modifies the model proposed by Faugeras-Toscani [Toscani 87] including up to three types of lens distortion, which has been explained in section 2.2.3. This fact increases the number of steps needed to calibrate the camera. A flowchart of the entire process is detailed in Figure 2.8.

The first step is to obtain the complete model of Weng. Then, Weng proposed to simplify the equations by introducing a variable substitution. Hence, equalling equations (2.9) and (2.14), equations (2.73) are obtained.

$$\begin{aligned} {}^C X_u + \delta_x\left({}^C X_u, {}^C Y_u\right) &= \left({}^I X_d - u_0\right)/-k_u \\ {}^C Y_u + \delta_y\left({}^C X_u, {}^C Y_u\right) &= \left({}^I Y_d - v_0\right)/-k_v \end{aligned}$$

At this point, two new unknowns are introduced, in the following manner,

$${}^C\hat{X}_d = \left({}^I X_d - u_0\right)/\alpha_u \qquad {}^C\hat{Y}_d = \left({}^I X_d - v_0\right)/\alpha_v \tag{2.73}$$

A substitution is then applied to simplify equations (2.73), obtaining equations (2.74).

$$\frac{{}^C X_u}{f} = {}^C\hat{X}_d - \frac{\delta_x\left({}^C X_u, {}^C Y_u\right)}{f} \qquad \frac{{}^C Y_u}{f} = {}^C\hat{Y}_d - \frac{\delta_y\left({}^C X_u, {}^C Y_u\right)}{f} \tag{2.74}$$

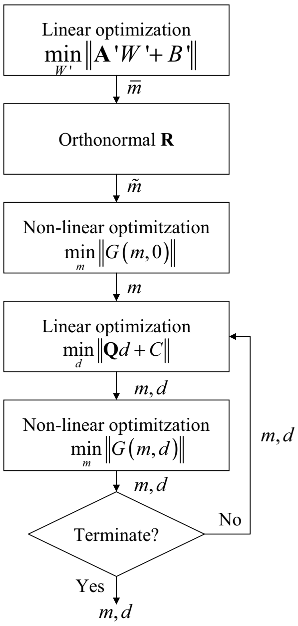


Figure 2.8: Flowchart of the method of Weng [Weng 92b].

This replacement of unknowns is necessary because the value of (${}^{C}X_u$, ${}^{C}Y_u$) cannot be obtained by observation. This fact makes it necessary to compute the distortion from the observed points after representing them with respect to the camera coordinate system, that is from (${}^{C}\hat{X}_d$, ${}^{C}\hat{Y}_d$) [Slama 80, Weng 92b]. This replacement is reasonable because the distortion on the image plane suffered by the point (${}^{C}X_u$,${}^{C}Y_u$) is approximately equal to the distortion suffered by the point(${}^{C}\hat{X}_d$, ${}^{C}\hat{Y}_d$). Therefore, the distortion coefficients in δ'_x and δ'_y will be estimated from (${}^{C}\hat{X}_d$,${}^{C}\hat{Y}_d$), instead of δ_x and δ_y, which was estimated from (${}^{C}X_u$,${}^{C}Y_u$). As a result, the equations which relate distorted to undistorted points are the following,

$$\frac{{}^{C}X_u}{f} = {}^{C}\hat{X}_d + \delta'_x\left({}^{C}\hat{X}_d, {}^{C}\hat{Y}_d\right) \qquad \frac{{}^{C}Y_u}{f} = {}^{C}\hat{Y}_d + \delta'_y\left({}^{C}\hat{X}_d, {}^{C}\hat{Y}_d\right) \tag{2.75}$$

Finally, redefining the coefficients k_1 and g_1 up to g_4, and combining equations (2.2), (2.3) and (2.75) the complete camera model is obtained,

$$\begin{aligned}
&\frac{r_{11}{}^{W}X_w + r_{12}{}^{W}Y_w + r_{13}{}^{W}Z_w + t_x}{r_{31}{}^{W}X_w + r_{32}{}^{W}Y_w + r_{33}{}^{W}Z_w + t_z} = {}^{C}\hat{X}_d \\
&\quad + (g_1 + g_3)\,{}^{C}\hat{X}_d^{\,2} + g_4{}^{C}\hat{X}_d{}^{C}\hat{Y}_d + g_1{}^{C}\hat{Y}_d^{\,2} + k_1{}^{C}\hat{X}_d\left({}^{C}\hat{X}_d^{\,2} + {}^{C}\hat{Y}_d^{\,2}\right) \\
&\frac{r_{21}{}^{W}X_w + r_{22}{}^{W}Y_w + r_{23}{}^{W}Z_w + t_y}{r_{31}{}^{W}X_w + r_{32}{}^{W}Y_w + r_{33}{}^{W}Z_w + t_z} = {}^{C}\hat{Y}_d \\
&\quad + g_2{}^{C}\hat{X}_d^{\,2} + g_3{}^{C}\hat{X}_d{}^{C}\hat{Y}_d + (g_2 + g_4)\,{}^{C}\hat{Y}_d^{\,2} + k_1{}^{C}\hat{Y}_d\left({}^{C}\hat{X}_d^{\,2} + {}^{C}\hat{Y}_d^{\,2}\right)
\end{aligned} \tag{2.76}$$

In order to be able to calibrate all the parameters of the model, Weng proposed to obtain a first approximation of the linear parameters, i.e. the extrinsic and intrinsic parameters without distortion. The m vector is now defined containing these linear parameters.

$$m = (u_0, v_0, \alpha_u, \alpha_v, t_x, t_y, t_z, \alpha, \beta, \gamma)^{\mathrm{T}} \tag{2.77}$$

Furthermore, the non-linear parameters which model the lens define a new vector d.

$$d = (k_1, g_1, g_2, g_3, g_4)^{\mathrm{T}} \tag{2.78}$$

Moreover, the calibration is based on the 3D test points and their projections. Let us call F the camera model, Ω the set of 3D points, and ω the set of their projections. Then, the calibration problem is the same as optimizing the parameters (m^*,d^*) which minimize the equation F by using both sets of test points.

$$F\left(\Omega, \omega, m^*, d^*\right) = \underset{m,d}{min}\, F\left(\Omega, \omega, m, d\right) \tag{2.79}$$

This problem of optimization can be solved by using a non-linear method, in the following manner:

1. Fix $d = 0$.
2. Calculate m, which minimizes F by fixing d, that is: $\underset{m}{min}\, F\left(\Omega, \omega, m, d\right)$
3. Calculate d, which minimizes F by fixing m, that is: $\underset{d}{min}\, F\left(\Omega, \omega, m, d\right)$
4. Return to step 2 until the minimization error is sufficiently tolerable.

This method of optimization is used to solve diverse problems. First, the vector d can be coupled with m making the minimization of F false. Second, the intrinsic parameters can not be optimized until a sufficient approximation of the extrinsic parameters is achieved. Third, since m corresponds to an approximation of the linear parameters, it cannot be the best solution if a significant distortion is presented.

With the aim of obtaining a good estimation of m with a non-linear optimization method, it is necessary to obtain an initial guess before iterating. Therefore, the initial guess is calculated supposing $d = 0$. Then, the model of Weng removing distortion, see equations (2.74), is applied to the n calibrating points, obtaining $2n$ equations of the form:

$$\begin{aligned}
&\left({}^{I}X_{ui} - u_0\right) {}^{W}X_{wi} r_{31} + \left({}^{I}X_{ui} - u_0\right) {}^{W}Y_{wi} r_{32} \\
&\quad + \left({}^{I}X_{ui} - u_0\right) {}^{W}Z_{wi} r_{33} + \left({}^{I}X_{ui} - u_0\right) t_z \\
&\quad - \alpha_u {}^{W}X_{wi} r_{11} - \alpha_u {}^{W}Y_{wi} r_{12} - \alpha_u {}^{W}Z_{wi} r_{13} - \alpha_u t_x = 0 \\
&\left({}^{I}Y_{ui} - v_0\right) {}^{W}X_{wi} r_{31} + \left({}^{I}Y_{ui} - v_0\right) {}^{W}Y_{wi} r_{32} \\
&\quad + \left({}^{I}Y_{ui} - v_0\right) {}^{W}Z_{wi} r_{33} + \left({}^{I}Y_{ui} - v_0\right) t_z \\
&\quad - \alpha_v {}^{W}X_{wi} r_{21} - \alpha_v {}^{W}Y_{wi} r_{22} - \alpha_v {}^{W}Z_{wi} r_{23} - \alpha_v t_y = 0
\end{aligned} \tag{2.80}$$

By using equations (2.80), all the m parameters can be calculated. As the m vector has 10 unknowns, it is necessary to use at least 5 test points. Nevertheless, a large number of points is used in order to obtain a more accurate solution. The following parameters are then defined,

$$\begin{array}{rclcrcl}
W_1 &=& \alpha_u r_1 + u_0 r_3 & \quad & w_4 &=& \alpha_u t_x + u_0 t_z \\
W_2 &=& \alpha_v r_2 + v_0 r_3 & \quad & w_5 &=& \alpha_v t_y + v_0 t_z \\
W_3 &=& r_3 & \quad & w_6 &=& t_z
\end{array} \tag{2.81}$$

where the vectors r_1, r_2 and r_3 correspond to each row of the matrix ${}^{C}\mathbf{R}_{W}$, respectively. Moreover, the set of equations (2.80) is expressed in matricial form as

$$\mathbf{A}\,W = 0 \tag{2.82}$$

where $\mathbf{A}$ is a matrix with $2n$ rows and 12 columns.

$$\mathbf{A} = \begin{pmatrix} -{}^{W}P_{w1}{}^{\mathrm{T}} & 0_{1x3} & {}^{I}X_{u1}{}^{W}P_{w1}{}^{\mathrm{T}} & -1 & 0 & {}^{I}X_{u1} \\ 0_{1x3} & -{}^{W}P_{w1}{}^{\mathrm{T}} & {}^{I}Y_{u1}{}^{W}P_{w1}{}^{\mathrm{T}} & 0 & -1 & {}^{I}Y_{u1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -{}^{W}P_{wn}{}^{\mathrm{T}} & 0_{1x3} & {}^{I}X_{un}{}^{W}P_{wn}{}^{\mathrm{T}} & -1 & 0 & {}^{I}X_{un} \\ 0_{1x3} & -{}^{W}P_{wn}{}^{\mathrm{T}} & {}^{I}Y_{un}{}^{W}P_{wn}{}^{\mathrm{T}} & 0 & -1 & {}^{I}Y_{un} \end{pmatrix} \tag{2.83}$$

However, the vector $W = (W_1, W_2, W_3, w_4, w_5, w_6)^{\mathrm{T}}$ cannot be directly calculated because of the homogeneity of the system, which deals with multiple solutions. Then, only one of these potential solutions satisfies the following conditions: a) The norm of the W_3 vector has to be the unity because it is the third row of the rotation matrix; b) The w_6 sign has to coincide with the position of the optical center with respect to the image plane: Positive if the z-axis intersects the image plane, otherwise has to be considered negative.

With the aim of avoiding the homogeneity of the system of equation (2.82), it is necessary to impose the following temporary restriction,

$$w_6 = t_z = 1 \tag{2.84}$$

Hence, equation (2.82) is modified, obtaining

$$\mathbf{A}'\,W' + B' = 0 \tag{2.85}$$

where $\mathbf{A}'$ is the first 11 columns of the $\mathbf{A}$ matrix, B' is the last column of A and W' is a vector of the 11 unknowns, i.e. $W' = (W_1, W_2, W_3, w_4, w_5)$. Then, W' is computed by using the pseudo-inverse,

$$W' = \left(\mathbf{A}'^{\mathrm{T}}\mathbf{A}'\right)^{-1}\mathbf{A}'^{\mathrm{T}}\,(-B') \tag{2.86}$$

At this point, W' is the solution of the system shown in equation (2.85). However, in order to be a solution of equation (2.82) as well, it has to accomplish the two constraints. Therefore, the solution is divided by $\|W_3\|$, which forces the norm of

W_3 to be the unity, and replaces the w_6 sign if necessary. See equation (2.87).

$$S = \begin{pmatrix} S_1 \\ S_2 \\ S_3 \\ s_4 \\ s_5 \\ s_6 \end{pmatrix} = \pm \frac{1}{\|W_3\|} \begin{pmatrix} W_1 \\ W_2 \\ W_3 \\ w_4 \\ w_5 \\ w_6 \end{pmatrix} \tag{2.87}$$

Moreover, knowing that the vectors r_1, r_2 and r_3 are orthogonal, equations (2.81) can be applied to obtain a first approximation of the m vector.

$$\begin{array}{rclrcl} \bar{u}_0 & = & S_1^{\mathrm{T}} S_3 & \bar{v}_0 & = & S_2^{\mathrm{T}} S_3 \\ \bar{\alpha}_u & = & -\left\| S_1 - \bar{u}_0 S_3 \right\| & \bar{\alpha}_v & = & -\left\| S_2 - \bar{v}_0 S_3 \right\| \\ \bar{t}_x & = & (s_4 - \bar{u}_0 s_6)/\bar{\alpha}_u & \bar{r}_1 & = & (S_1 - \bar{u}_0 S_3)/\bar{\alpha}_u \\ \bar{t}_y & = & (s_5 - \bar{v}_0 s_6)/\bar{\alpha}_v & \bar{r}_2 & = & (S_2 - \bar{v}_0 S_3)/\bar{\alpha}_v \\ \bar{t}_z & = & s_6 & \bar{r}_3 & = & S_3 \end{array} \tag{2.88}$$

However, this first approximation does not imply that the matrix ${}^C\bar{\mathbf{R}}_W$ is orthonormal. The next step consists of calculating the orthonormal matrix ${}^C\tilde{\mathbf{R}}_W$. The first step is to verify,

$$\left\| {}^C\tilde{\mathbf{R}}_W - {}^C\bar{\mathbf{R}}_W \right\| = \min_{{}^C\mathbf{R}_W} \left\| {}^C\bar{\mathbf{R}}_W - {}^C\mathbf{R}_W \right\| \tag{2.89}$$

With the aim of solving equation (2.89), it is rewritten including a 3×3 identity matrix $\mathbf{I}$.

$$\left\| {}^C\tilde{\mathbf{R}}_W I - {}^C\bar{\mathbf{R}}_W \right\| = \min_{{}^C\mathbf{R}_W} \left\| {}^C\bar{\mathbf{R}}_W - {}^C\mathbf{R}_W \right\| \tag{2.90}$$

It is then defined a 4×4 matrix $\mathbf{B}$,

$$\mathbf{B} = \sum_{i=1}^{3} \mathbf{B}_i^{\mathrm{T}} \mathbf{B}_i \tag{2.91}$$

where,

$$\mathbf{B}_i = \begin{pmatrix} 0 & (i_i - \bar{r}_i)^{\mathrm{T}} \\ \bar{r}_i - i_i & (\bar{r}_i + i_i)_{\times} \end{pmatrix} \tag{2.92}$$

and where $\mathbf{I} = (i_1, i_2, i_3)^{\mathrm{T}}$, and $(x,y,z)_{\times}$ is the antisymmetric matrix of the vector

(x,y,z), that is:

$$(x,y,z)_\times = \begin{pmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{pmatrix} \tag{2.93}$$

The vector $q = (q_0, q_1, q_2, q_3)^\mathrm{T}$ is then obtained by calculating the eigenvalues associated with matrix $\mathbf{B}$, where q_i is an eigenvalue and $q_i \leq q_{i+1}$. Finally, the solution of the matrix ${}^C\tilde{\mathbf{R}}_W$ is shown in the following equation,

$${}^C\tilde{\mathbf{R}}_W = \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2\left(q_1q_2 - q_0q_3\right) & 2\left(q_1q_3 + q_0q_2\right) \\ 2\left(q_2q_1 + q_0q_3\right) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2\left(q_2q_3 - q_0q_1\right) \\ 2\left(q_3q_1 - q_0q_2\right) & 2\left(q_3q_2 + q_0q_1\right) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix} \tag{2.94}$$

With the orthornomal rotation matrix, the rest of the parameters are recalculated once again, obtaining:

$$\begin{array}{rclrcl} \tilde{u}_0 & = & S_1^\mathrm{T}\tilde{r}_3 & \tilde{v}_0 & = & S_2^\mathrm{T}\tilde{r}_3 \\ \tilde{\alpha}_u & = & -\left\|S_1 - \tilde{u}_0\tilde{r}_3\right\| & \tilde{\alpha}_v & = & -\left\|S_2 - \tilde{v}_0\tilde{r}_3\right\| \\ & & \tilde{t}_x & = & (s_4 - \tilde{u}_0 s_6)/\tilde{\alpha}_u & \\ & & \tilde{t}_y & = & (s_5 - \tilde{v}_0 s_6)/\tilde{\alpha}_v & \\ & & \tilde{t}_z & = & \bar{t}_z & \end{array} \tag{2.95}$$

An iterative method is then used to recalculate, for the third time, the values of m, assuming zero distortion. Finally, a two-stage iterative method is used. In the first stage, the parameters of d are linearly obtained by using least-squares. The second stage computes the values of m iteratively. These stages are repeated as many times as needed depending on the desired accuracy.

Stage of non-linear optimization of m by fixing d.

The camera model of Weng is expressed in equation (2.96), see also (2.76).

$$\begin{aligned}
U(x) &= \frac{r_{11}{}^W X_w + r_{12}{}^W Y_w + r_{13}{}^W Z_w + t_x}{r_{31}{}^W X_w + r_{32}{}^W Y_w + r_{33}{}^W Z_w + t_z} - {}^C\hat{X}_d \\
&\quad - (g_1 + g_3)\,{{}^C\hat{X}_d}^2 - g_4{}^C\hat{X}_d{}^C\hat{Y}_d - g_1{{}^C\hat{Y}_d}^2 - k_1{}^C\hat{X}_d\left({{}^C\hat{X}_d}^2 + {{}^C\hat{Y}_d}^2\right) \qquad (2.96) \\
V(x) &= \frac{r_{21}{}^W X_w + r_{22}{}^W Y_w + r_{23}{}^W Z_w + t_y}{r_{31}{}^W X_w + r_{32}{}^W Y_w + r_{33}{}^W Z_w + t_z} - {}^C\hat{Y}_d \\
&\quad - g_2{{}^C\hat{X}_d}^2 - g_3{}^C\hat{X}_d{}^C\hat{Y}_d - (g_2 + g_4)\,{{}^C\hat{Y}_d}^2 - k_1{}^C\hat{Y}_d\left({{}^C\hat{X}_d}^2 + {{}^C\hat{Y}_d}^2\right)
\end{aligned}$$

Equation (2.97) shows the function of minimization that has to be used in optimization.

$$\sum_{i=1}^{n}\left\{\left({}^I X_{di} - {}^I X_{di}(m,d)\right)^2 + \left({}^I Y_{di} - {}^I Y_{di}(m,d)\right)^2\right\} \qquad (2.97)$$

At this point any optimization algorithm such as Newton-Raphson or Levenberg-Marquardt can be used to optimize equations (2.96).

Stage of linear optimization of d by fixing m.

Note that by arranging equations (2.14) and (2.76), the equations which have to be optimized become linear. Therefore, they can be optimized by using the pseudo-inverse technique. The linear equations obtained are the following,

$$\begin{aligned}
&{}^I X_d(m,d) - {}^I X_d = u_0 + \alpha_u{}^C\hat{X}_d - {}^I X_d \\
&= \quad u_0 + \alpha_u\left(\frac{r_{11}{}^W X_w + r_{12}{}^W Y_w + r_{13}{}^W Z_w + t_x}{r_{31}{}^W X_w + r_{32}{}^W Y_w + r_{33}{}^W Z_w + t_z}\right. \\
&\qquad \left. - (g_1 + g_3)\,{{}^C\hat{X}_d}^2 - g_4{}^C\hat{X}_d{}^C\hat{Y}_d - g_1{}^C\hat{Y}_d^2 - k_1{}^C\hat{X}_d\left({{}^C\hat{X}_d}^2 + {{}^C\hat{Y}_d}^2\right)\right) - {}^I X_d \qquad (2.98) \\
&{}^I Y_d(m,d) - {}^I Y_d = v_0 + \alpha_v{}^C\hat{Y}_d - {}^I Y_d \\
&= \quad v_0 + \alpha_v\left(\frac{r_{21}{}^W X_w + r_{22}{}^W Y_w + r_{23}{}^W Z_w + t_y}{r_{31}{}^W X_w + r_{32}{}^W Y_w + r_{33}{}^W Z_w + t_z}\right. \\
&\qquad \left. - g_2{{}^C\hat{X}_d}^2 - g_3{}^C\hat{X}_d{}^C\hat{Y}_d - (g_2 + g_4)\,{}^C\hat{Y}_d^2 - k_1{}^C\hat{Y}_d\left({{}^C\hat{X}_d}^2 + {{}^C\hat{Y}_d}^2\right)\right) - {}^I Y_d
\end{aligned}$$

where the function to minimize is expressed in equation (2.99):

$$\min_{d} \|\mathbf{Q}\, d + C\| \tag{2.99}$$

where,

$$C = \begin{pmatrix} u_0 + \alpha_u \left(\frac{r_{11}{}^{W}X_{w1} + r_{12}{}^{W}Y_{w1} + r_{13}{}^{W}Z_{w1} + t_x}{r_{31}{}^{W}X_{w1} + r_{32}{}^{W}Y_{w1} + r_{33}{}^{W}Z_{w1} + t_z} \right) - {}^{I}X_{d1} \\ v_0 + \alpha_v \left(\frac{r_{21}{}^{W}X_{w1} + r_{22}{}^{W}Y_{w1} + r_{23}{}^{W}Z_{w1} + t_y}{r_{31}{}^{W}X_{w1} + r_{32}{}^{W}Y_{w1} + r_{33}{}^{W}Z_{w1} + t_z} \right) - {}^{I}Y_{d1} \\ \vdots \\ u_0 + \alpha_u \left(\frac{r_{11}{}^{W}X_{wn} + r_{12}{}^{W}Y_{wn} + r_{13}{}^{W}Z_{wn} + t_x}{r_{31}{}^{W}X_{wn} + r_{32}{}^{W}Y_{wn} + r_{33}{}^{W}Z_{wn} + t_z} \right) - {}^{I}X_{dn} \\ v_0 + \alpha_v \left(\frac{r_{21}{}^{W}X_{wn} + r_{22}{}^{W}Y_{wn} + r_{23}{}^{W}Z_{wn} + t_y}{r_{31}{}^{W}X_{wn} + r_{32}{}^{W}Y_{wn} + r_{33}{}^{W}Z_{wn} + t_z} \right) - {}^{I}Y_{dn} \end{pmatrix} \tag{2.100}$$

$$\mathbf{Q} = \begin{pmatrix} -\alpha_u{}^{C}\hat{X}_{d1}\left({{}^{C}\hat{X}_{d1}}^{2} + {{}^{C}\hat{Y}_{d1}}^{2}\right) & -\alpha_u\left({{}^{C}\hat{X}_{d1}}^{2} + {{}^{C}\hat{Y}_{d1}}^{2}\right) & 0 & -\alpha_u{}^{C}\hat{X}_{d1} & -\alpha_u{}^{C}\hat{X}_{d1}{}^{C}\hat{Y}_{d1} \\ -\alpha_v{}^{C}\hat{Y}_{d1}\left({{}^{C}\hat{X}_{d1}}^{2} + {{}^{C}\hat{Y}_{d1}}^{2}\right) & 0 & -\alpha_v\left({{}^{C}\hat{X}_{d1}}^{2} + {{}^{C}\hat{Y}_{d1}}^{2}\right) & -\alpha_v{}^{C}\hat{X}_{d1}{}^{C}\hat{Y}_{d1} & -\alpha_v{}^{C}\hat{Y}_{d1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -\alpha_u{}^{C}\hat{X}_{dn}\left({{}^{C}\hat{X}_{dn}}^{2} + {{}^{C}\hat{Y}_{dn}}^{2}\right) & -\alpha_u\left({{}^{C}\hat{X}_{dn}}^{2} + {{}^{C}\hat{Y}_{dn}}^{2}\right) & 0 & -\alpha_u{}^{C}\hat{X}_{dn} & -\alpha_u{}^{C}\hat{X}_{dn}{}^{C}\hat{Y}_{dn} \\ -\alpha_v{}^{C}\hat{Y}_{dn}\left({{}^{C}\hat{X}_{dn}}^{2} + {{}^{C}\hat{Y}_{dn}}^{2}\right) & 0 & -\alpha_v\left({{}^{C}\hat{X}_{dn}}^{2} + {{}^{C}\hat{Y}_{dn}}^{2}\right) & -\alpha_v{}^{C}\hat{X}_{dn}{}^{C}\hat{Y}_{dn} & -\alpha_v{}^{C}\hat{Y}_{dn} \end{pmatrix} \tag{2.101}$$

The solution for d can now be obtained by using the pseudo-inverse in the following way,

$$d = -\left(\mathbf{Q}^{\mathrm{T}}\mathbf{Q}\right)^{-1}\mathbf{Q}^{\mathrm{T}}C \tag{2.102}$$

2.4 Accuracy Evaluation

The systems used to evaluate the accuracy of camera calibration can be classified in two groups. The first group is based on analyzing the discrepancy between the real

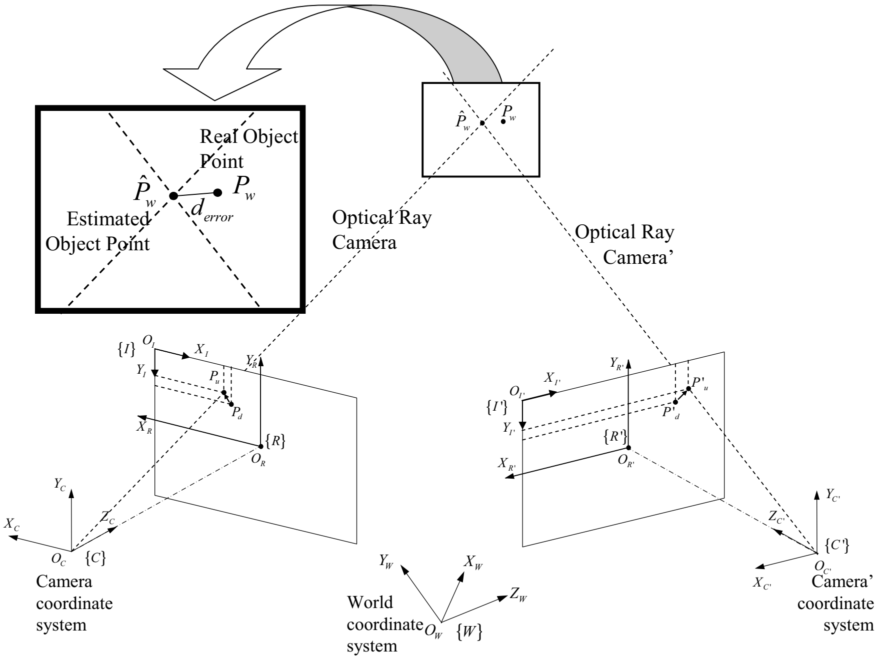


Figure 2.9: Accuracy obtained from stereo triangulation.

position of the 3D object point with respect to the 3D position estimated from its 2D projection. The second group compares the real position in pixels of a 2D image point with respect to the calculated projection of the 3D object point on the image plane.

2.4.1 3D Measurement

In the following text, some of the most frequently used methods of accuracy evaluation in 3D coordinates system are described.

1. *3D position obtained from stereo triangulation.* In the first step, two images are acquired from a set of 3D test points whose 3D coordinates are known. In the second, the estimated 3D coordinates of the same points are computed from their projections using the calibrated parameters. Finally, the discrepancy between real and estimated positions is compared. In this case the accuracy depends on calibration of both cameras, see Figure 2.9. The stereo triangulation principle is detailed in the next chapter of stereo vision.

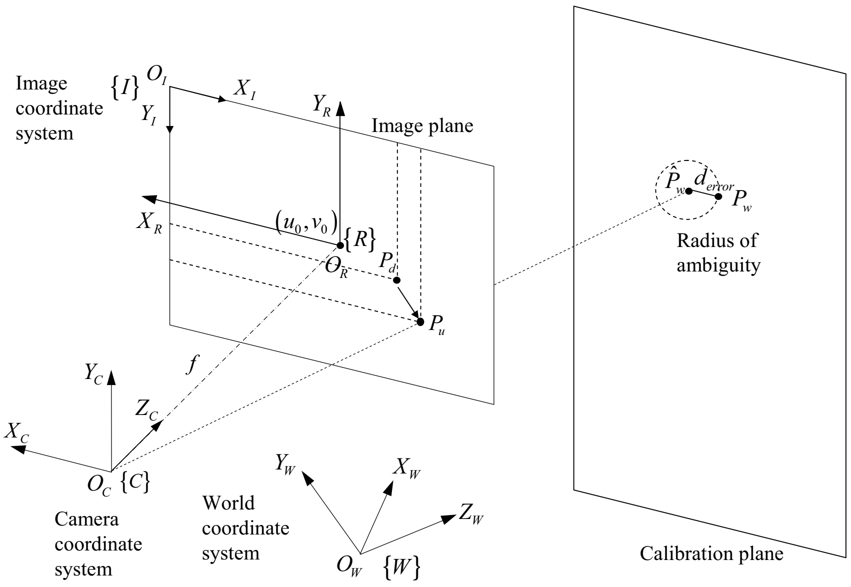


Figure 2.10: Accuracy obtained from radius of ambiguity in the calibrating plane.

2. *Radius of ambiguity in the calibrating plane.* First, a set of 3D test points, which lay on the test plane and whose coordinates in the world coordinate system are known, is acquired. Second, for each image point, the calibrated model is used to project the optical ray back from the focal point through the 2D projection. The transverse of the optical ray with the test plane determines the intersection point. The distance from the 3D test point to this intersection point defines a radius of ambiguity around the 3D point (see Figure 2.10).

3. *Distance with respect to the optical ray.* This method is a generalization of the previous one. In this case, the discrepancy to be measured is the distance between the 3D test points and the optical ray generated from their projections (see Figure 2.11).

4. *Normalized Stereo Calibration Error* [Weng 92b]. The array of pixels in an image is projected back to the scene so that each back-projected pixel covers a certain area of the object surface (see Figure 2.12). This area indicates the uncertainty of the basic resolution at this distance. The orientation of the surface has been fitted to a plane which is orthogonal to the optical axis.

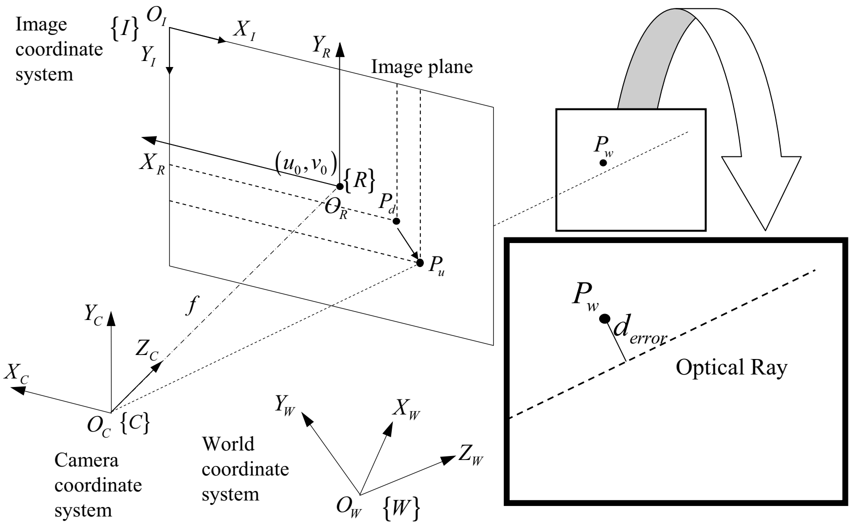


Figure 2.11: Accuracy obtained from distance between 3D object point and camera optical ray.

Let the depth of this plane be equal to ${}^{C}Z_{w}$, and the row and column focal lengths be α_u and α_v. The back projection of the pixel on this plane is a rectangle of $a \times b$ size. Let the real coordinates of the i_{th} 3D object points $({}^{C}X_{wi}, {}^{C}Y_{wi}, {}^{C}Z_{wi})$ be represented in the camera coordinate system, and let its coordinates obtained by back-projecting the pixel and intersecting it with the surface plane $({}^{C}\hat{X}_{wi}, {}^{C}\hat{Y}_{wi}, {}^{C}\hat{Z}_{wi})$ be also represented in the camera coordinate system. The Normalized Stereo Calibration Error (NSCE) is defined as,

$$\mathrm{NSCE} = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{\left({}^{C}\hat{X}_{wi} - {}^{C}X_{wi}\right)^2 + \left({}^{C}\hat{Y}_{wi} - {}^{C}Y_{wi}\right)^2}{{}^{C}\hat{Z}_{wi}^{\;2}\left(\alpha_u^{-2} + \alpha_v^{-2}\right)\Big/12}\right]^{1/2} \tag{2.103}$$

2.4.2 2D Measurement

The following techniques describe two ways to compute the accuracy in the image plane in pixels.

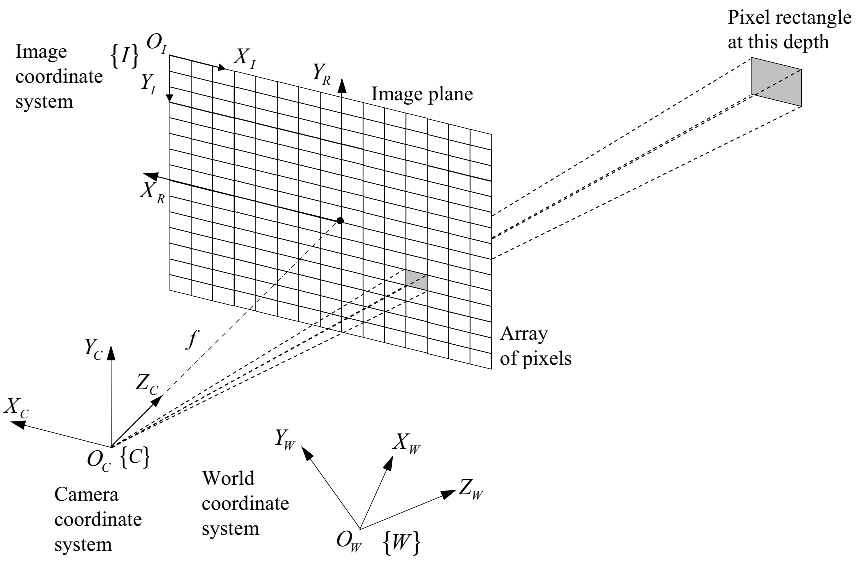


Figure 2.12: Backward projection of a pixel to 3D space.

1. *Accuracy of distorted image coordinates.* First, take an image of a set of 3D test points. Then, calculate the 2D position of each 3D point on the image plane, taking into account lens distortion. Accuracy is obtained by measuring the discrepancy between the real 2D points (obtained from image segmentation) and the estimated ones (obtained by using the camera model), shown in Figure 2.13.

2. *Accuracy of undistorted image coordinates.* First, take an image of a set of 3D test points. Calculate the linear projection of the 3D points on the image plane, without taking lens distortion into account. Continue by determining the real 2D points through image segmentation and remove the lens distortion by using the camera model to obtain a set of undistorted points. Finally, accuracy is obtained by measuring the discrepancy between the linear projections and the undistorted points, shown in Figure 2.14.

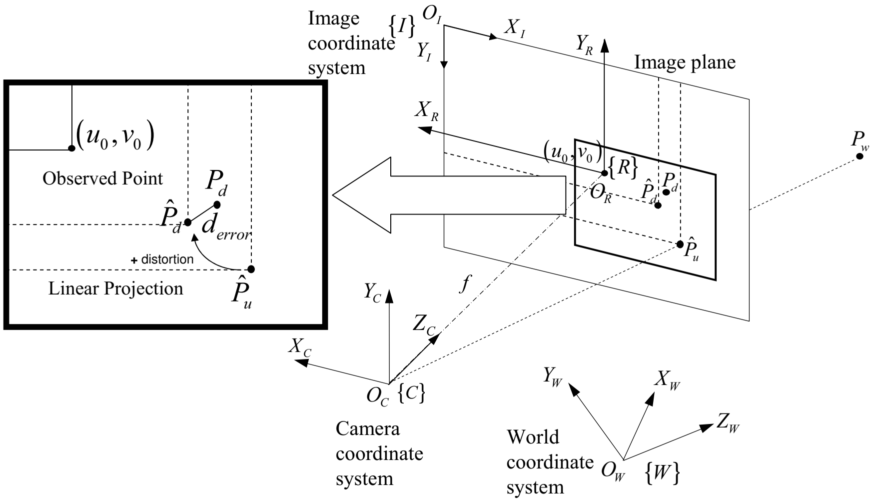


Figure 2.13: Accuracy of distorted image coordinates.

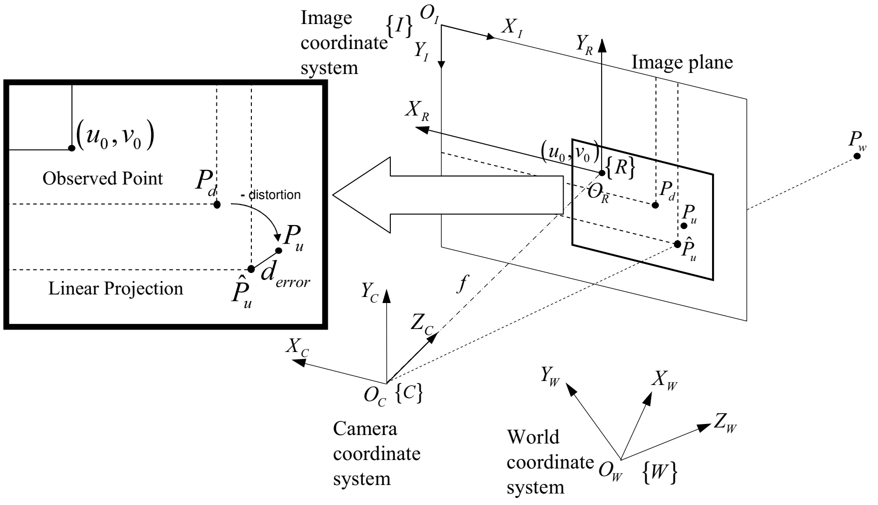


Figure 2.14: Accuracy of undistorted image coordinates.

2.5 Summary

In this chapter, we have presented a comparative study of the most commonly used camera calibrating methods of the last few decades. These techniques cover a wide range of the classical hard calibration of image sensors which begin from a previous knowledge of a set of 3D points and their corresponding 2D projections on an image plane in order to estimate the camera parameters. Hence, this study has described a total of 5 different camera calibrating techniques which include implicit vs. explicit calibration and linear vs. non-linear calibration.

A great deal of attention has been paid to using the same nomenclature and a standardized notation in the presentation of all the techniques. Actually, this is one of the greatest difficulties when going into the details of any calibrating technique. This problem usually arises because each method defines a different set of coordinate systems and camera parameters. Therefore, all the techniques have been re-arranged so as to allow a comparative presentation. The reader has been introduced to calibration with the implicit linear technique of the pseudo-inverse presented by Hall, with the explicit linear calibration of Faugeras-Toscani presented afterwards. Furthermore, this chapter has described an easy modification of the Faugeras method in order to include radial lens distortion, the well-known method of Tsai, and finally the complete method of Weng which models up to three different kinds of lens distortion has been discussed. In order to compare the accuracy provided by each technique surveyed, a brief description of accuracy evaluation has been presented.

The reader can take the equations directly from this chapter and easily use them in the desired calibrating algorithm[2]. There are numerous advantages thanks to an accurate calibration. For instance, dense reconstruction of 3D objects and surfaces has applications in visual inspection and medical imaging, such as quality control in industrial manufacturing and reconstruction of human spinal cords and skulls for the detection of deformations or for surgical purposes. Another problem is the 3D position estimation of an object in a scene, which has many applications such as obstacle avoidance, landmark detection and industrial part assembly, among others.

---

[2] The implementation of the camera calibration algorithms surveyed are detailed in appendix B.

Chapter 3

Stereo Vision and the Epipolar Geometry

Stereo Vision is based on acquiring 3D information from multiple views obtained by a single moving camera or a fixed structure composed of at least two cameras. The relationship between each set of two viewspoints is determined by the epipolar geometry and the fundamental matrix estimation is the only way to compute it. This chapter is a fresh look at the subject and overviews classic and newer presented methods of fundamental matrix estimation which have been classified into linear methods, iterative methods and robust methods.

3.1 Introduction

Nowadays, there are many methods to obtain 3D information from an unknown scene by means of computer vision and every year new proposals and techniques are aggregated. These methods are classified in what Woodham [Woodham 78] refers to as *direct* methods and *indirect* methods. Direct methods are those which try to measure distance ranges directly for example as in pulsed laser based systems, where the depth information is the only information available. Indirect methods are those which attempt to determine distance by measuring parameters calculated from images of the illuminated object. Several direct and indirect methods commonly refer to these subjects as *Shape from* X, where X is one of a number of options considering the spread of such technologies in the last few years. Shape from X can be itemized at the same time into the following groups:

1. *Techniques based on modifying the intrinsic camera parameters, i.e. Depth from Focus/Defocus and Depth from Zooming.* Considering the Depth from Focus/Defocus, images are obtained by changing the camera parameters (typically the focal setting or the image plane axial position), and taken from the same point of view. The difference between depth from focus and depth from defocus is that, in the first case, it is possible to dynamically change the camera parameters during the surface estimation process while in the second, this is not possible [Favaro 02]. Besides, Depth-from-Zoom considers the use of multiple images taken with a single camera coupled with a motorized zoom.

2. *Techniques based on considering an additional source of light projected onto the scene, i.e. Shape from Photometric Stereo and Shape from Structured Light.* Photometric Stereo is based on considering various radiance maps of the measuring surface captured by using a single camera and a set of known light sources. The use of at least three radiance maps determines a single position and orientation for every imaged point [Solomon 96]. The structured light technique is based on the projection of a known pattern of light onto the measuring surface, such as points, lines, stripes or grids. 3D information of the scene is obtained by analyzing the deformations of the projected pattern when it is imaged by the camera [Salvi 97].

3. *Techniques based on considering additional surface information, i.e. Shape from Shading, Shape from Texture and Shape from Geometric Constraints.* Shape from Shading uses the pattern of shading in a single image to infer the shape of the surface. Often, the parameter of the reflectance map is unknown. In this case we have to estimate the albedo and illuminant direction. From the reflection map and by assuming local surface smoothness, we can estimate local surface normal, which can be integrated to give local surface shape [Gibbins 94]. The basic principle behind shape from texture is the distortion of the individual texels. Their variation across the image gives an estimate of the shape of the observed surface. The shape reconstruction exploits perspective distortion which makes objects farther from the camera appear smaller, and foreshortening distortion which makes objects not parallel to the image plane seem shorter. Assuming that the normals are dense enough and

the surface is smooth, these distortions can be used to reconstruct the surface shape [Chantler 94]. Finally, Shape from Geometric Constraints consider the problem of obtaining 3D reconstruction from 2D points localized in a single image. Planarity, colinearity, known angles and other geometric properties provided by the "user" are used to remove ambiguities from the scene and, if possible, obtain a single reconstruction [Grossmann 02].

4. *Techniques merely based on multiple views, such as Shape from Stereo and Shape from Motion.* Shape from Stereo is based on solving the correspondence problem between two or more views of a given surface taken from different locations. Each image point determines an optical ray which intersects with the others in space with the aim of computing the 3D surface point. Shape from Motion exploits the relative motion between camera and scene. Similar to the stereo technique, this process can be divided into the subprocesses of finding correspondences from consecutive frames and finally to the reconstruction of the scene. The differences between consecutive frames are, on average, much smaller than those of typical stereo pairs because the image sequences are sampled at higher rates. Motion computation can be obtained from optical flow and differential epipolar constraint which is the main subject of chapter 4.

This chapter focuses on stereo vision which is basically achieved by means of two different approaches.

The first approach is based on a previous camera calibration so that the imaging sensor model which relates 3D object points to their 2D projections on the image plane is known. Camera modelling and calibration, presented in the previous chapter, have been widely studied during the last decades (see camera calibration surveys [Armangué 00] and [Salvi 02]). Once the system has been calibrated, the camera model can be used either to estimate the 2D projection of an object point or to compute the 3D optical ray passing through a given 2D image projection. Therefore, at least two optical rays are needed to compute the 3D position of the object point by means of triangulation (section 3.2). The set of optical rays can be obtained by a single moving camera coupled with a robotic arm, a binocular system consisting of two cameras located at fixed positions, or a multiple-view configuration such as a trinocular configuration. Figure 3.1 shows an example of a stereo vision system composed of two pinhole cameras, a 3D object and its projection on both

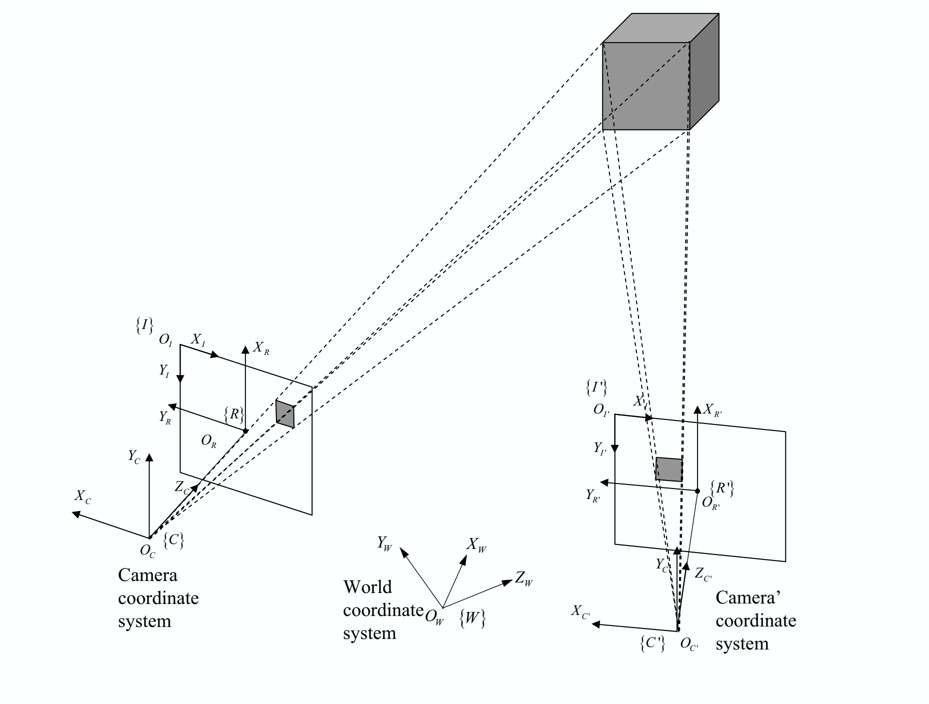


Figure 3.1: A stereo vision system composed by two pinhole cameras.

image planes.

Calibration can not be used in active systems due to its lack of flexibility. Note that in active systems, the optical and geometrical characteristics of the cameras might change dynamically depending on the imaging scene and camera motion. The second approach is based on establishing some geometric constraints between the imaging sensors [Faugeras 93]. Thus, the geometry between two cameras related to a set of 3D surface points is called the epipolar geometry, which was first published by Longuet-Higgins in 1981 [Longuet-Higgins 81]. Since that time, a great deal of effort has been expended to increase the knowledge in this field [Faugeras 93, Huang 89]. Many articles have been presented on self-calibrated and uncalibrated systems as a result of the boom in the 1990's. For instance, in 1992 Faugeras published a brief survey on self-calibration and the derived Kruppa equations which are used to estimate the camera parameters from the epipolar geometry [Faugeras 92a]. Basically, intrinsic parameters of both cameras and the position and orientation of one camera related to the other can be extracted by using Kruppa equations [Hartley 97]. In the same year, Faugeras also gave an answer to the question *"What can be seen in three dimensions with an uncalibrated stereo rig?"* [Faugeras 92b]. Hartley also did a lot of work with geometry and how it is contained in the essential and the fundamental matrix [Hartley 92a] as well as the estimation of the camera position [Hartley 92b]. Two years later, Deriche et al. presented a robust method for recovering epipolar geometry based on a matching by correlation and detecting the outliers [Deriche 94]. As a result, Hartley studied the geometry involved in a rotating camera [Hartley 94] while Li studied the geometry of a head-eye system [Li 94] and Luong et al. introduced a Canonic representation [Luong 94b]. Also, in 1994, Luong and Faugeras published an interesting article on analyzing the stability of the fundamental matrix due to uncertainty in the epipole computation, noise in the image point localization, camera motion, and so on [Luong 94a].

Some applications of epipolar geometry are the simplification of the image matching in stereoscopic systems [Brooks 96], the estimation of camera motion [Jang 96] and scene reconstruction [Zhang 96b]. It is important, therefore, to develop accurate techniques to compute it. Classic linear methods are mainly based on least-squares minimization [Zhang 98] and eigen values minimization [Torr 97]. Other methods are based on optimizing linear methods by means of iteration [Li 96]. Robust methods are based on computing a more accurate geometry detection and removing false matchings [Rousseeuw 87, Zhang 98]. Robust computation is still a subject for wide research focusing mainly on proposing new estimators to improve the accuracy of the fundamental matrix and on reducing computation expenses [Bober 98, Stewart 95, Torr 00].

This chapter surveys nineteen of the most widely used techniques in computing the fundamental matrix such as the 7-point, the least-squares and the eigen analysis linear techniques among others and robust techniques such as M-estimators, LMedS, RANSAC and so on [Armangué 03b, Salvi 01]. All these techniques have been implemented and their accuracy analyzed in synthetic and real scenarios. This chapter is organized as follows. First, section 3.2 describes the principle to compute a 3D surface point from at least two optical rays given by the image projections of such a surface point. Second, a brief introduction to epipolar geometry and fundamental matrix is presented in section 3.3. Then, all the surveyed methods are described in section 3.4, analyzing their advantages and drawbacks with respect to the previous methods and presenting an overview of every surveyed technique from the algorithmic point of view. Finally, the chapter concludes with a summary.

3.2 The Triangulation Principle

The triangulation principle is based on computing the 3D position of a surface point from the intersection of a set of optical rays determined by at least two views of the same 3D surface point. In practice, a camera does not function in the same way as a pinhole camera due to lens distortion. Also, there is uncertainty around the image point due to camera digitalization and image segmentation. This uncertainty is caused by the fact that both optical rays do not cross at any single point. The triangulation process must find, then, the 3D surface point closest to both optical rays.

Consider P_d an image point in the first image plane $\{I\}$ and P'_d its corresponding point in the second image plane $\{I'\}$. Both image points are projections of a 3D object point P_w (see Figure 3.2). The intrinsic camera parameters of both cameras permit computation of P_u and P'_u with respect to the camera coordinate system (see chapter 2). The 3D point P_u, lying on the image plane, and the origin of the first camera coordinate system O_C define the first optical ray given by the vector. Also, P'_u and the origin of the second camera coordinate system $O_{C'}$ define the vector. Consider P_r the point lying on the optical ray defined by $\vec{u}$ as being the closest to the optical ray defined by $\vec{v}$. Again, consider P_s the point lying on the optical ray defined by $\vec{v}$ as being the closest point to the optical ray defined by $\vec{u}$,

$$\begin{aligned} P_r &= P_d + \alpha\vec{u} \qquad (3.1) \\ P_s &= P'_d + \beta\vec{v} \qquad (3.2) \end{aligned}$$

where $\alpha, \beta \in \mathbb{R}$ and $\overline{P_rP_s}$ is orthogonal to $\vec{u}$ and $\vec{v}$. Then, the following equations are solved by using $\overline{P_rP_s} = \overline{P_uP'_u} - \alpha\vec{u} + \beta\vec{v}$.

$$\left\{ \begin{array}{lllll} 0 & = & \overline{P_rP_s}\,\vec{u} & = & \overline{P_uP'_u} - \alpha\|\vec{u}\|^2 + \beta(\vec{u}\,\vec{v}) \\ 0 & = & \overline{P_rP_s}\,\vec{v} & = & \overline{P_uP'_u} - \alpha(\vec{u}\,\vec{v}) + \beta\|\vec{v}\|^2 \end{array} \right. \qquad (3.3)$$

where α and β are unknown. Then, P_r and P_s are obtained by substituting α and β in equations (3.1) and (3.2). Finally, P_w is easily determined by using the following equation.

$$P_w = \frac{P_r - P_s}{2} \qquad (3.4)$$

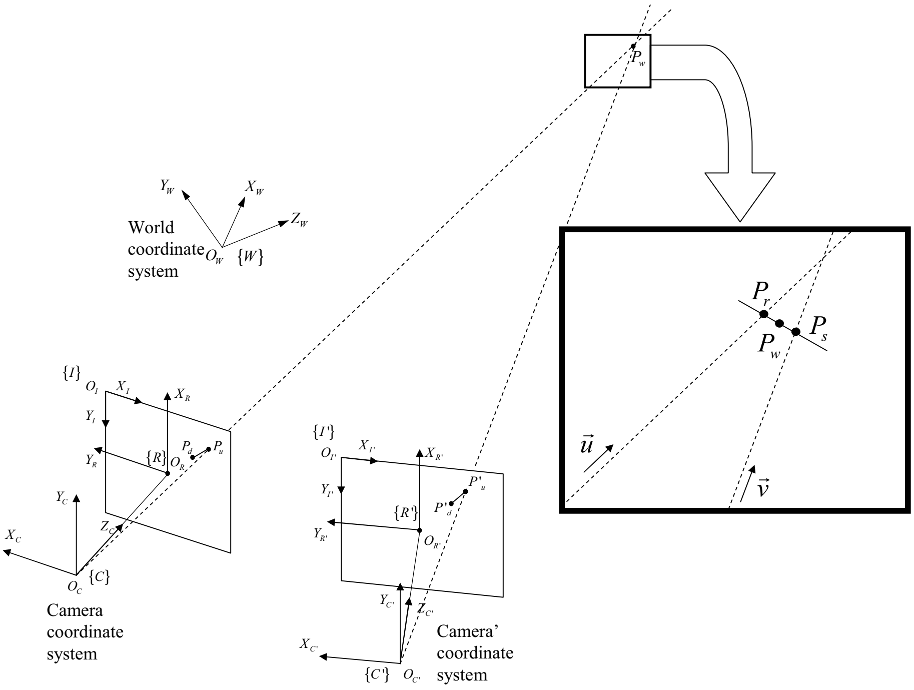


Figure 3.2: Stereo triangulation to obtain 3D object point.

3.3 Epipolar Geometry

Given a 3D object point $M = ({}^{W}X,\, {}^{W}Y,\, {}^{W}Z,\, 1)^{\mathrm{T}}$ expressed with respect to a world coordinate system $\{W\}$, and its 2D projection on the image plane in pixels $m = ({}^{I}X,\, {}^{I}Y,\, 1)^{\mathrm{T}}$, both points are related to a projective transformation matrix as shown in equation (3.5),

$$s\,m = {}^{I}\mathbf{P}_{W}\, M \tag{3.5}$$

in which s is a scale factor and ${}^{I}\mathbf{P}_{W}$ is a 3×4 matrix, which can be decomposed as

$${}^{I}\mathbf{P}_{W} = {}^{I}\mathbf{A}_{C}\, {}^{C}\mathbf{K}_{W} \tag{3.6}$$

in which ${}^{I}\mathbf{A}_{C}$ is a 3×4 matrix relating the metric camera coordinate system located at the focal point O_C to the image coordinate system located at the top-left corner of the image plane in pixels, that is the optical and internal geometry of the camera. Moreover, ${}^{C}\mathbf{K}_{W}$ is a 4×4 matrix which relates the camera coordinate system $\{C\}$ to the world coordinate system $\{W\}$, that is the position and orientation of the

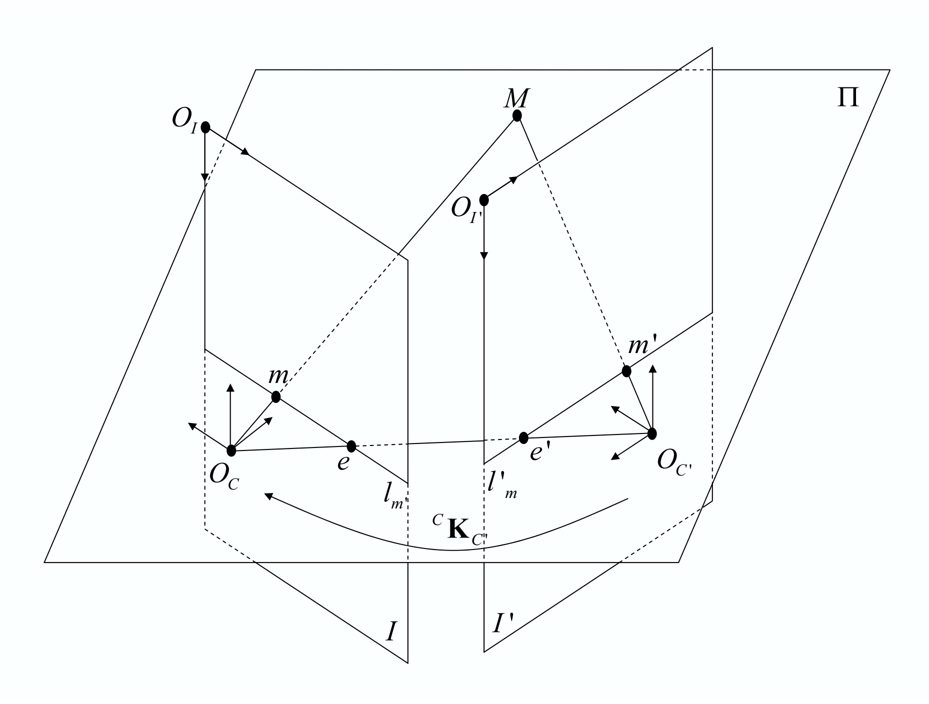


Figure 3.3: The geometric relation between two pinhole cameras.

camera in the scene (see chapter 2 for further details).

$$^C\mathbf{K}_W = \begin{pmatrix} ^C\mathbf{R}_W & ^Ct_W \\ 0 & 1 \end{pmatrix} \tag{3.7}$$

Epipolar geometry defines the geometry between two cameras creating a stereoscopic system or the geometry between two different positions of a unique moving camera. Consider an object point M and its 2D projections m and m' on both image planes, the 3 points define a plane Π, which intersects both image planes at the epipolar lines $l_{m'}$ and l'_m respectively, as shown in Figure 3.3. Note that the same plane Π can be computed using both focal points O_C and $O_{C'}$ and a single 2D projection, which is the principle to reduce the correspondence problem to a single scanning along the epipolar line. Moreover, the intersection of all the epipolar lines defines an epipole on both image planes, which can also be obtained by intersecting the line defined by both focal points O_C and $O_{C'}$ with both image planes.

All the epipolar geometry is contained in the so called fundamental matrix as shown in equation (3.8).

$$m^{\mathrm{T}}\mathbf{F}m' = 0 \tag{3.8}$$

The fundamental matrix **F** contains the intrinsic parameters of both cameras and the rigid transformation of one camera related to the other, which depends on which camera has been considered as the origin of the world coordinate system. In equation (3.9), the origin of the world coordinate system coincides with the coordinate system of the second camera, located at $O_{C'}$.

$$\mathbf{F} = {}^{I}\mathbf{A}_{C}{}^{-\mathrm{T}}[{}^{C}t_{C'}]_{\times}{}^{C}\mathbf{R}_{C'}\,{}^{I'}\mathbf{A}_{C'}{}^{-1} \tag{3.9}$$

A particular case of the fundamental matrix is the essential matrix. When the intrinsic camera parameters are known, it is possible to simplify equations (3.8) and (3.9) obtaining

$$q^{\mathrm{T}}\mathbf{E}\,q' = 0 \tag{3.10}$$

where,

$$q = {}^{I}\mathbf{A}_{C}{}^{-1}m, \qquad \mathbf{E} = [{}^{C}t_{C'}]_{\times}{}^{C}\mathbf{R}_{C'}, \qquad q' = {}^{I'}\mathbf{A}_{C'}{}^{-1}m' \tag{3.11}$$

The matrix **E** is called essential [Huang 89].

3.4 Estimating the Fundamental Matrix

In the last few years, several methods to estimate the fundamental matrix have been proposed, which can be classified into linear, iterative and robust methods. Linear and iterative methods can cope with bad point localization in the image plane due to noise in image segmentation. Robust methods can cope with both image noise and outliers, i.e. wrong matching between point correspondences in both image planes. All of these methods are based on solving a homogeneous system of equations which can be deduced from equation (3.8) rewriting it in the following way:

$$\mathbf{U}f = 0 \tag{3.12}$$

where,

$$f = (F_{11}, F_{12}, F_{13}, F_{21}, F_{22}, F_{23}, F_{31}, F_{32}, F_{33})^{\mathrm{T}} \tag{3.13}$$

$$\mathbf{U} = \begin{pmatrix} {}^IX_1\,{}^IX'_1 & {}^IX_1\,{}^IY'_1 & {}^IX_1 & {}^IY_1\,{}^IX'_1 & {}^IY_1\,{}^IX'_1 & {}^IY_1 & {}^IX'_1 & {}^IY'_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ {}^IX_n\,{}^IX'_n & {}^IX_n\,{}^IY'_n & {}^IX_n & {}^IY_n\,{}^IX'_n & {}^IY_n\,{}^IX'_n & {}^IY_n & {}^IX'_n & {}^IY'_n & 1 \end{pmatrix} \tag{3.14}$$

It is important to note that there are only 7 independent parameters and 9 unknowns. The 7 independent parameters are given by two independent columns and the scale factor forcing the fundamental matrix to be rank-2 [Zhang 98].

3.4.1 Linear Methods

The linear method of the *seven points* is based on computing the fundamental matrix by using only seven point correspondences [Zhang 98]. Due to the homogeneity of the equations, the solution is a set of matrices of the form

$$\mathbf{F} = \alpha \mathbf{F}_1 + (1-\alpha)\mathbf{F}_2 \tag{3.15}$$

By forcing the rank of the matrix to be equal to 2 and using the expression $\det(\mathbf{F})$, a cubic polynomial is obtained which has to be solved to obtain α and then $\mathbf{F}$. The main advantage of this method is that a fundamental matrix can be estimated by using only seven points. However this fact becomes a drawback when some points are poorly located. Moreover, the 7-points method cannot be applied in the presence of redundancy. Hence, it can not be applied using n points where $n > 7$.

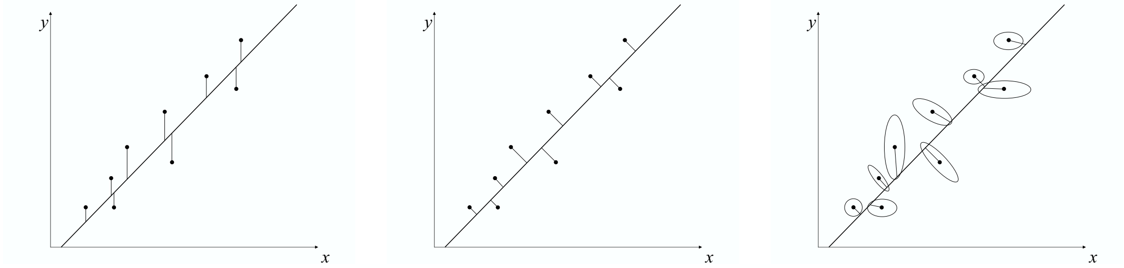
Another interesting method is the *8-points method*, in which the redundancy of points permits the minimization of the error in estimating $\mathbf{F}$. The equation to minimize in the 8-points method is the residual of equation (3.8), that is:

$$\min_{\mathbf{F}} \sum_i \left(m_i^{\mathrm{T}} \mathbf{F} m'_i \right)^2 \tag{3.16}$$

The classical method to solve such an equation is the *least-squares technique* of forcing one of the components of $\mathbf{F}$ to be the unity [Luong 96]. This simplification can be assumed because $\mathbf{F}$ is always defined up to a scale factor. Then, the equation to solve is

$$f' = \left(\mathbf{U}'^{\mathrm{T}} \mathbf{U}' \right)^{-1} \mathbf{U}'^{\mathrm{T}} c_9 \tag{3.17}$$

in which $\mathbf{U}'$ is a matrix containing the first eight columns of $\mathbf{U}$, c_9 is the last column of $\mathbf{U}$ (see also equation (3.14)) and f' is a vector containing the first eight elements



(a) (b) (c)

Figure 3.4: Cost functions: (a) Algebraic least squares minimization; (b) Orthogonal least squares; and (c) Approximate maximum likelihood.

of f. Note that the last element of f is 1.

A variant of the 8-points method can be applied if equation 3.16 is solved by using *eigen analysis*, also called *orthogonal least-squares technique* [Torr 97]. In this case $\mathbf{F}$ can be determined from the eigen vector corresponding to the smallest eigen value of $\mathbf{U}^{\mathrm{T}}\mathbf{U}$. The difference between this method and the classical least-squares resides in the form of calculating the error between correspondences and epipolar lines so that an orthogonal minimization is much more realistic (see Figure 3.4a and Figure 3.4b).

The last linear method we surveyed is the *analytic method with rank-2 constraint* which imposes the rank-2 constraint in minimization [Zhang 98]. Then, the matrix $\mathbf{U}'$ is defined as the composition of the first seven columns of $\mathbf{U}$ and c_8 and c_9 are defined as the eighth and ninth columns of $\mathbf{U}$ respectively, so that $\mathbf{F}$ can be computed as

$$f' = -f_8 \left(\mathbf{U}'^{\mathrm{T}}\mathbf{U}'\right)^{-1} \mathbf{U}'^{\mathrm{T}} c_8 - f_9 \left(\mathbf{U}'^{\mathrm{T}}\mathbf{U}'\right)^{-1} \mathbf{U}'^{\mathrm{T}} c_9 \tag{3.18}$$

in which f' is the vector containing the first seven elements of f, and f_8 and f_9 are the eighth and ninth elements of f. In order to obtain the values of f_8 and f_9, $\mathbf{F}$ is computed by using the seven points algorithm. Then, f is computed by selecting from any choice of pairs of $\mathbf{F}$, the one which minimizes $\|f\| = 1$. Although the *analytic method with rank-2 constraint* obtains a rank-2 matrix, it does not greatly improve the results of the previously explained methods.

3.4.2 Iterative Methods

Iterative methods can be classified into two groups: those that minimize the distances between points and epipolar lines and those that are based on the gradient.

In the first classification, the iterative methods minimizing the distances between points and epipolar lines are based on solving the following equation

$$\min_{\mathbf{F}} \sum_i \left(d^2(m_i, \mathbf{F}m'_i) + d^2(m'_i, \mathbf{F}m_i) \right) \tag{3.19}$$

A first approach consists of directly applying an iterative method as *Newton-Raphson* or *Levenberg-Marquardt* in the equation (3.19) [Salvi 97]. Another possibility consists of applying an *iterative linear method* as was proposed by Luong and Faugeras (1996) [Luong 96], in which equation (3.19) has to be rewritten as

$$\min_{\mathbf{F}} \sum_i w_i^2 \left(m_i^{\mathrm{T}} \mathbf{F} m'_i \right)^2 \tag{3.20}$$

where,

$$\begin{aligned} w_i &= \left(\frac{1}{{l_1}^2 + {l_2}^2} + \frac{1}{{l'_1}^2 + {l'_2}^2} \right)^{1/2} && (3.21) \\ \mathbf{F}m'_i &= (l_1, l_2, l_3)^{\mathrm{T}} && (3.22) \\ \mathbf{F}^{\mathrm{T}} m_i &= (l'_1, l'_2, l'_3)^{\mathrm{T}} && (3.23) \end{aligned}$$

The iterative linear method is based on computing the weight value w_i equivalent to the epipolar distances by using the previous $\mathbf{F}$ (in the first iteration $w_i = 1$) and then minimize by using *least-squares* in each iteration. Neither approach imposes the rank-2 constraint. However, the *nonlinear minimization in parameter space* [Zhang 98] can solve this situation. This method is based on parameterizing the fundamental matrix, keeping in mind that it has a rank-2 in the following way,

$$\mathbf{F} = \begin{pmatrix} a & b & -ax_e - by_e \\ c & d & -cx_e - dy_e \\ -ax_{e'} - cy_{e'} & -bx_{e'} - dy_{e'} & (ax_e + by_e)x_{e'} + (cx_e + dy_e)y_{e'} \end{pmatrix} \tag{3.24}$$

in which (x_e, y_e) and $(x_{e'}, y_{e'})$ are the coordinates of the epipole in the first image plane and second image plane, respectively. Equation (3.24) is just one of the multiple parameterizations of **F** which must be computed. Finally, the estimated **F** becomes the parameterization which maximizes the following equation,

$$(ad-bc)^2\sqrt{{x_e}^2+{y_e}^2+1}\sqrt{{x_{e'}}^2+{y_{e'}}^2+1} \tag{3.25}$$

The iteration of this method allows the computation of a better rank-2 **F**.

Besides, the minimization of equation (3.16) is not accurate enough to obtain a good estimation because the variance of points is not analogous and the least-squares technique assumes they are comparable. In order to overcome this drawback, the second group of methods has to be considered.

The second group of methods considers the *gradient-based* [Hartley 00]. In this case, the equation to solve is

$$\min_{\mathbf{F}}\sum_i \left(m_i^{\mathrm{T}}\mathbf{F}m_i'\right)^2/g_i^2 \tag{3.26}$$

where $g_i=\sqrt{{l_1}^2+{l_2}^2+{l_1'}^2+{l_2'}^2}$.

This method has to potential minimizations: least-squares and eigen analysis. Besides, Chojnacki, Brooks, van den Hengel and Gawley [Chojnacki 02] recently proposed two new iterative methods based on an approximate maximum likelihood estimate which can be applied to several computer vision applications. Such methods are called *Fundamental Numerical Scheme* (FNS) and *Constrained Fundamental Numerical Scheme* (CFNS). Both methods are based on Newton-Raphson minimization technique. Then, in order to estimate the fundamental matrix **F** the following minimization has to be overcome,

$$J_{AML}=\sum_i\frac{(m_i^T\mathbf{F}m_i')^2}{m_i^T\mathbf{F}\mathbf{F}^Tm_i+m_i'^T\mathbf{F}\mathbf{F}^Tm_i'} \tag{3.27}$$

so that $\partial_f J_{AML}(f)$ is the row vector of partial derivatives of J_{AML} with respect to f. The minimization forces such vector to zero so that $\partial_f J_{AML}(f)=2\mathbf{X}_f f$. Then, arranging the terms of this equation,

$$\mathbf{X}_f f=0 \tag{3.28}$$

where,

$$\begin{aligned}
\mathbf{X}_f &= \sum_{i=1}^{n} \frac{\mathbf{A}_i}{f^T\mathbf{B}_i f} - \sum_{i=1}^{n} \frac{f^T\mathbf{A}_i f}{(f^T\mathbf{B}_i f)^2}\mathbf{B}_i && (3.29)\\
\mathbf{A}_i &= u_i u_i^T && (3.30)\\
\mathbf{B}_i &= \partial_m u_i \mathbf{\Lambda}_m \partial_m u_i^T && (3.31)
\end{aligned}$$

∂_m represents the partial derivative of the corresponding points, $\mathbf{\Lambda}_m$ is the symmetric covariance matrix that relates the point uncertainty [Brooks 01, Chojnacki 00] (see Figure 3.4c). Then, FNS is based on solving equation (3.28) by means of an initial seed to search for a local minimum.

The CFNS [Chojnacki 02] improves FNS by including in the optimization method an ancillary constraint obtained from the minimization function. In such a case, the $\mathbf{X}_f$ matrix is replaced to a more complex one, i.e. $\mathbf{Z}_f$.

$$\mathbf{Z}_f = \mathbf{P}_f \mathbf{X}_f \mathbf{P}_f \qquad (3.32)$$

where,

$$\begin{aligned}
\mathbf{P}_f &= \mathbf{I} - \|a_f\|^{-2} a_f a_f^T && (3.33)\\
a_f &= \partial_f \phi(f)^T && (3.34)
\end{aligned}$$

and $\phi(f) = 0$ is the ancillary constraint.

3.4.3 Robust Methods

In this section we present five robust methods: *M-Estimators*, *Least-Median-Squares* (LMedS), *Random Sampling* (RANSAC), MLESAC and MAPSAC which can be used both in the presence of outliers and in bad point localization.

M-estimators [Hartley 00] reduces the effect of outliers weighting the residual of each point. Consider r_i the residual of $m_i^{\mathrm{T}}\mathbf{F}m_i'$. Then, *M-estimators* are based on solving the following expression

$$\min_{\mathbf{F}} \sum_i w_i \left(m_i^{\mathrm{T}}\mathbf{F}m_i'\right)^2 \qquad (3.35)$$

in which w_i is a weight function. A lot of different weight functions have been proposed so a new M-estimator is obtained for each one. A common weight function proposed by Huber [Huber 81] is the following

$$w_i = \begin{cases} 1 & |r_i| \leq \sigma \\ \sigma/|r_i| & \sigma < |r_i| \leq 3\sigma \\ 0 & 3\sigma < |r_i| \end{cases} \tag{3.36}$$

Another interesting weight function is proposed by Mosteller and Tukey [Mosteller 77],

$$w_i = \begin{cases} \left(1 - \left(\frac{r_i}{4.6851}\right)^2\right)^2 & |r_i| \leq 4.6851\sigma \\ 0 & \text{otherwise} \end{cases} \tag{3.37}$$

In order to obtain σ, the robust standard deviation can be used (see [Zhang 98]).

$$\sigma = 1.4826\,(1 + 5/(n-7))\,\text{median}_i\,|r_i| \tag{3.38}$$

There are a lot of weight functions and for each one we obtained different results. Overall, the results given by this method are quite good in the presence of gaussian noise in image point localization, but they are rather limited in outlier detection.

LMedS [Zhang 98] and *RANSAC* [Torr 97] techniques are quite similar. Both techniques are based on randomly selecting the set of points used to compute an approximation of **F** by using a linear method. The difference between both techniques is in the way use to determine the chosen **F**. LMedS calculates for each **F** the median distance between the points and epipolar lines, in which the chosen fundamental matrix has to minimize this median. RANSAC calculates for each **F** the number of inliers, in which the chosen **F** is the one that maximizes it. Once the outliers are removed, **F** is recalculated with the aim of obtaining a better approach.

Another difference between both methods is that LMedS is more restrictive than RANSAC, so that LMedS removes more points than RANSAC. However, the principal constraints of both techniques is their lack of repetitivity due to the aleatory way of selecting the points. Although experimental results show that LMedS gives better results in terms of accuracy, it does not always model the epipolar geometry properly.

Recently some other methods based on RANSAC has been proposed. The

MLESAC [Torr 00] (*Maximum LikElihood SAmple Consensus*) is a generalization of RANSAC based on the same point selection strategy and the solution is the one that maximizes a likelihood, so that the shape of a normal distribution instead of the number of inliers. Besides, MAPSAC [Torr 02] (*Maximum A Posteriori SAmple Consensus*) improves MLESAC being more robust against noise and outliers including Bayesian probabilities in minimization.

3.4.4 Algorithmic Overview

This section gives an algorithmic point of view of the surveyed methods to estimate the fundamental matrix described in sections 3.4.1, 3.4.2 and 3.4.3. The main objective of this section is to present an overall schema to make agile the comparison among the different methodologies.

Figure 3.5 summarizes the algorithmic methodology of the linear methods. Besides, Figure 3.6 abstracts the methodology used by the iterative methods to minimize the distance between points and the corresponding epipolar lines. Moreover, Figure 3.7 illustrates the schema of the iterative methods based on minimizing the gradient. The two schemas are two different implementations of the gradient technique, that is linear-squares and eigen analysis. Figure 3.8 shows the methods who minimize the approximate maximum likelihood. The two schemas are based on FNS and CFNS which methodology are quite similar. Note that $\mathbf{Z}_f = 0$ is equivalent to $\mathbf{Z}_f^T \mathbf{Z}_f = 0$ obtaining a symmetric matrix which is replaced to $\mathbf{X}_f$ in the algorithm. Then, Figure 3.9 deals with robust methods based in M-Estimators. The schemas are three different implementations of the M-Estimator technique, that is the minimization of equation (3.35) by using least-squares and eigen analysis, and the implementation proposed by Torr [Torr 97] in which the fundamental matrix is forced to be rank-2 in every iteration (explained in detail in section 3.4.5). Finally, the methods shown in Figure 3.10 correspond to LMedS and RANSAC, respectively.

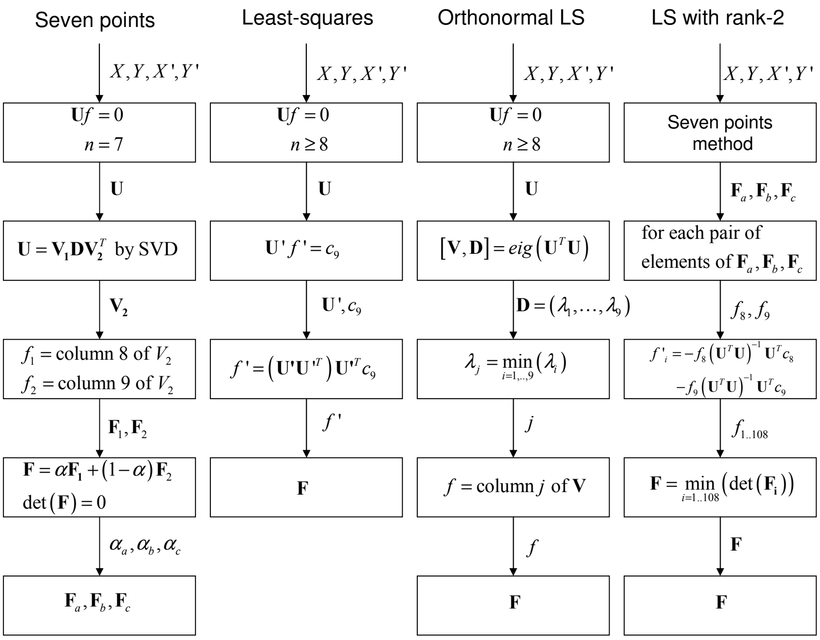


Figure 3.5: Linear methods flow schemes.

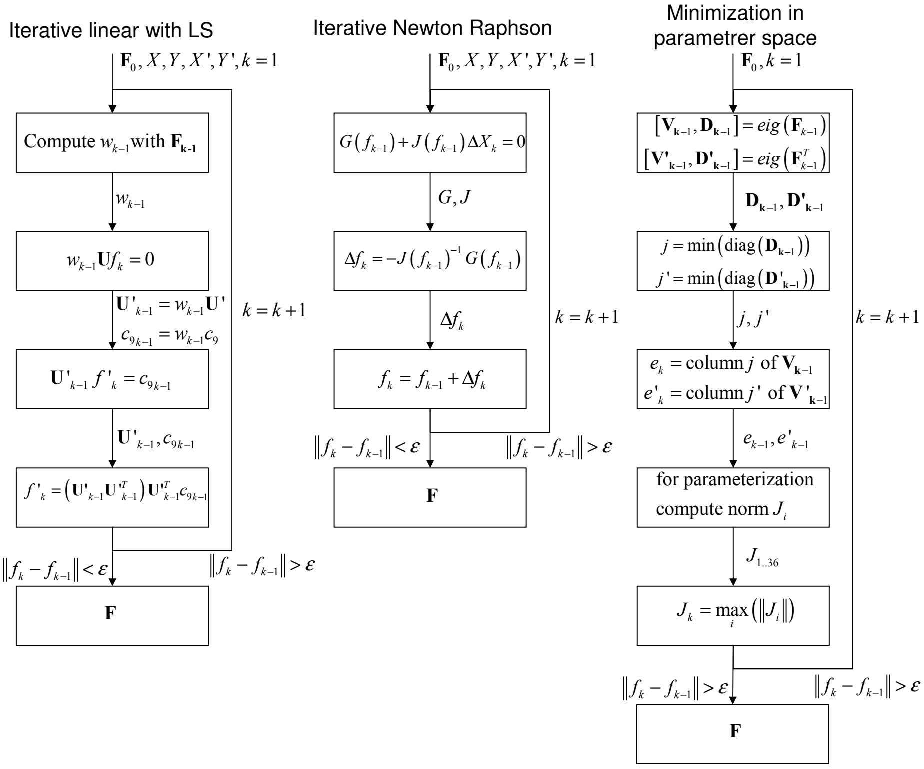


Figure 3.6: Flow schemes of the iterative methods minimizing the distances between points and epipolar lines.

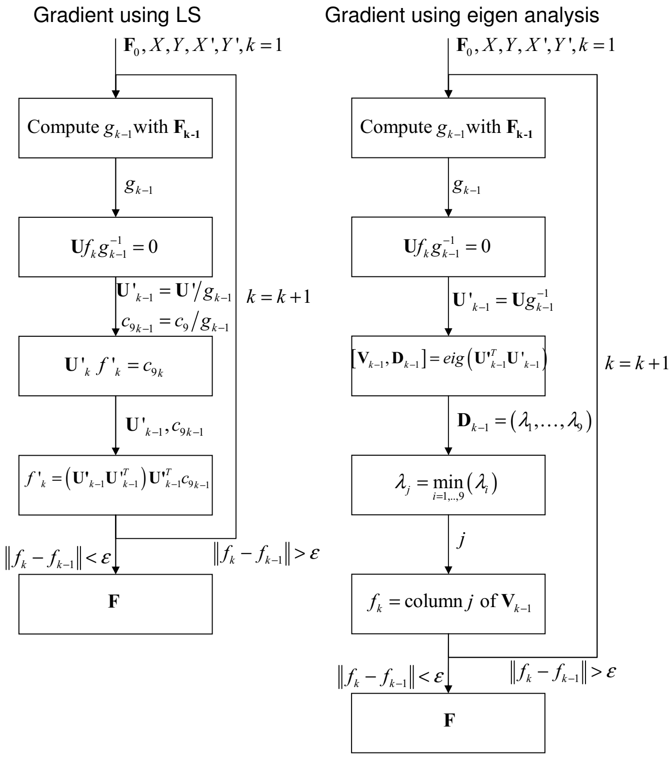


Figure 3.7: Flow schemes of the iterative methods minimizing the gradient.

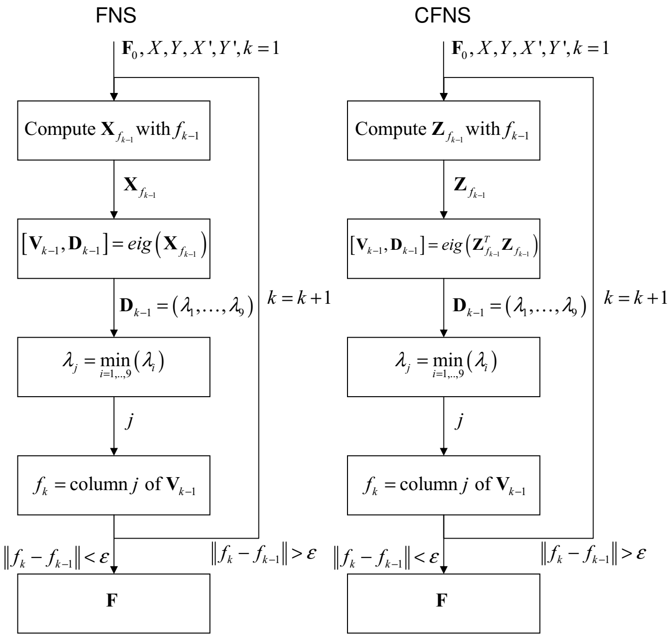


Figure 3.8: Flow schemes of the iterative methods FNS and CFNS.

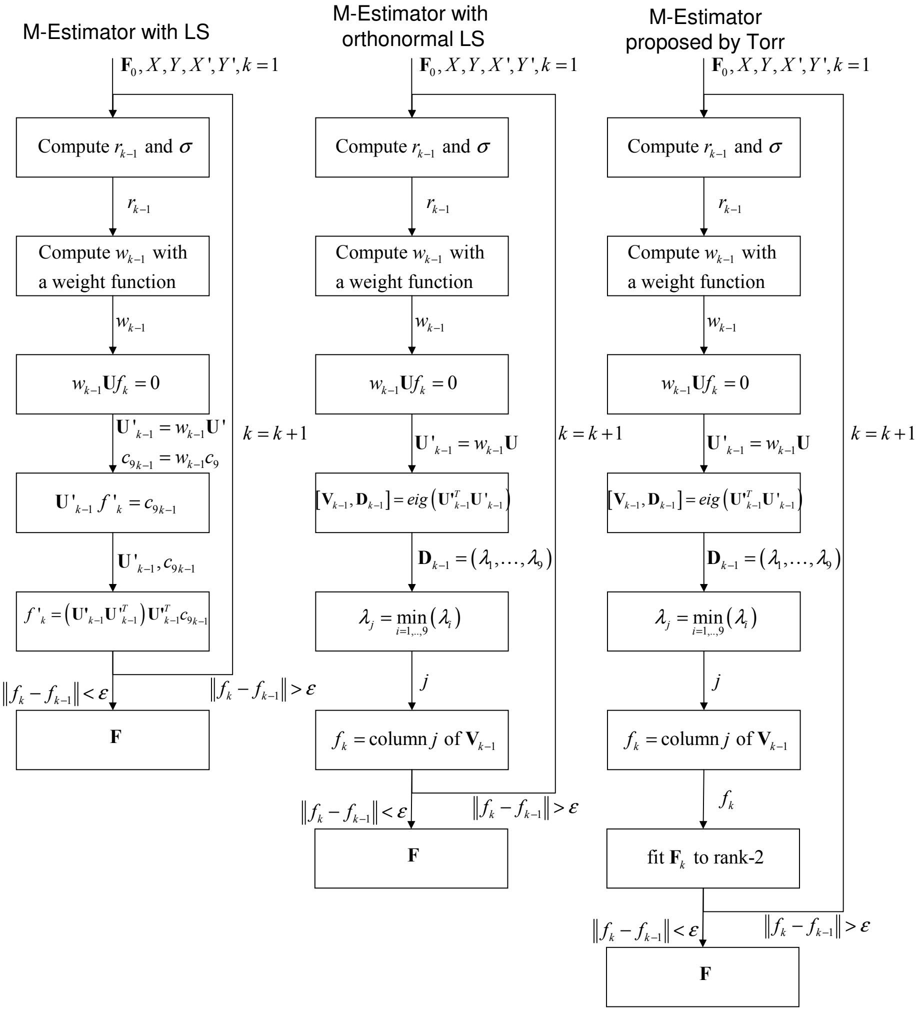


Figure 3.9: Flow schemes of M-Estimators methods.

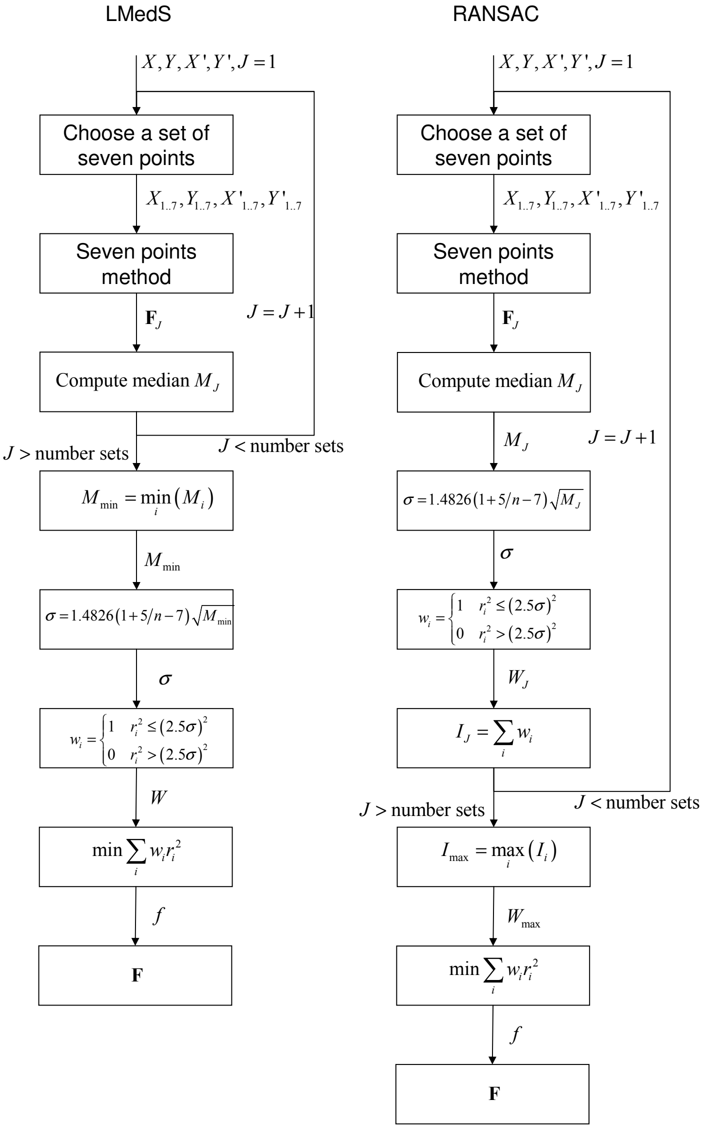


Figure 3.10: Flow schemes of LMedS and RANSAC methods.

3.4.5 Considerations in F Estimation

This section present to considerations to keep in mind when using fundamental matrix estimators. On one hand the data normalization and on the other hand the rank-2 constraint.

Normalizing data

Data normalization is a key point in fundamental matrix estimation. It has been proved that the computation should not be applied directly to raw data in pixels due to potential uncertainties given by huge numbers. The process of normalization consists of scaling and translating the data so that points m_i and m'_i are transformed to ($\hat{m}_i = \mathbf{T}m_i$ and $\hat{m}'_i = \mathbf{T}'m'_i$) by using two transformation matrices $\mathbf{T}$ and $\mathbf{T}'$ respectively. Then, the $\hat{\mathbf{F}}$ matrix is estimated from the normalized points and, finally, it has to be restored to obtain $\mathbf{F}$ using the following equation

$$\mathbf{F} = \mathbf{T}^{\mathrm{T}}\hat{\mathbf{F}}\mathbf{T}' \tag{3.39}$$

Basically there are two different methods of data normalization. The first method proposed [Zhang 98] normalizes the data between [-1,1]. The second was proposed by Hartley [Hartley 95] and is based on two transformations. First, the points are translated so that their centroid is placed at the origin. Then, the points are scaled so that the mean of the distances of the points to the origin is $\sqrt{2}$. It has been proved that Hartley's method gives more accurate results compared to normalizing between [-1,1].

Rank-2 constraint

In most circumstances, the estimated $\mathbf{F}$ should be a rank-2 matrix in order to model the epipolar geometry with all the epipolar lines intersecting in a unique epipole. Although the rank-2 constraint is not imposed in most of the surveyed methods, there is a mathematical method which transforms a rank-n square matrix to the closest rank-$(n-1)$ matrix [Hartley 00]. The $\mathbf{F}$ is decomposed in

$$\mathbf{F} = \mathbf{U}\mathbf{S}\mathbf{V}^{\mathrm{T}} \tag{3.40}$$

by using singular value decomposition, where $\hat{\mathbf{S}} = \mathrm{diag}(\sqrt{\lambda_1}, \sqrt{\lambda_2}, \sqrt{\lambda_3})$. The component with the smallest weight is removed obtaining
$\hat{\mathbf{S}} = \mathrm{diag}(\sqrt{\lambda_1}, \sqrt{\lambda_2}, 0)$. Then, $\mathbf{F}$ is recalculated in the following way:

$$\hat{\mathbf{F}} = \mathbf{U}\hat{\mathbf{S}}\mathbf{V}^{\mathrm{T}} \tag{3.41}$$

However, transforming the obtained **F** to a rank-2 matrix gives worse results then keeping a rank-2 **F** in the minimization process. Besides, a rank-3 matrix gives an additional degree of freedom that usually reduces the distance between points and epipolar lines.

Overall, we suggest the use of any method which imposes a rank-2 matrix in the computation of **F** because rank-2 matrices models more accurate the epipolar geometry.

3.5 Summary

Fundamental matrix estimation is a key point in computer vision because an accurate estimation is required in order to compute the epipolar geometry of a stereoscopic system. Some applications of epipolar geometry are: a) the simplification of image matching which is reduced to a search along the epipolar line; b) the estimation of camera motion when mounted either to a robotic arm or a mobile robot with useful applications in grasping and mobile navigation; and c) scene reconstruction with several applications in industrial inspection, prototyping and even mould generation.

In this chapter we present a comparative study of the most commonly used fundamental matrix estimation methods of the last few decades. The techniques cover a wide range of methods from the classical linear estimation such as the 7-points, the least squares and the eigen analysis up to some of the most frequently used robust methods such as the M-Estimators, the least median of squares and the random sampling. Hence, this study is presented describing a total of 15 methods and up to 19 different implementations. The different methods have been programmed[1].

The chapter describes each method surveyed and compares it with the others,

---

[1] Detailed in appendix B.

giving a fresh look which may be useful to any reader entering the field for the first time or who is searching for some sort of method for his application. This chapter presents a scheme of every method surveyed allowing the reader to compare the techniques in terms of algorithm complexity.

Finally, we have also pointed out that data have to be normalized and the best results have been obtained by using the method proposed by Hartley in his recently published book [Hartley 00]. Finally, a rank-2 matrix is preferable because it models the epipolar geometry with a single epipole defined by the intersection of all the epipolar lines.

Chapter 4

Motion from Discrete and Differential Epipolar Geometry

The estimation of camera egomotion is an old problem in computer vision. Since the 1980s, many approaches based on both the discrete and the differential epipolar constraint have been proposed. The discrete case is used mainly in self-calibrated stereoscopic systems, whereas the differential case deals with a single moving camera. This chapter surveys several methods for 3D motion estimation unifying the mathematics convention which are then adapted to the common case of a mobile robot moving on a plane.

4.1 Introduction

The problem of estimating structure and motion from image sequences has been studied by the computer vision community over the past few decades. Most techniques try to decouple the two problems by first estimating the motion and then the structure.

Approaches to motion estimation can be separated into *discrete* and *differential* methods, depending on whether they use a set of image correspondences or optical flow. Probably the best known point of view is the discrete epipolar constraint formulated by Longuet-Higgins [Longuet-Higgins 81], Huang [Huang 89] and Faugeras [Faugeras 93]. In this case the relative 3D displacement between both views is recovered by the epipolar constraint from a set of correspondences in both image planes. This information is contained in the *fundamental matrix* which includes the intrinsic parameters of both cameras and the position and orientation of one camera with respect to the other. The *essential matrix* can be extracted of the intrinsic parameters are previously known. Many papers describe different methods to estimate the fundamental matrix and it has been discussed in depth in the previous chapter of this document as well as in the articles [Armangué 03b, Salvi 01]. Tsai and Huang [Tsai 84] proved that there are only two possible 3D displacements obtainable from the essential constraint. In 1986, Toscani and Faugeras [Toscani 86] proposed a three-step algorithm for recovering the 3D displacement from image correspondences.

Overall, this essential constraint only recovers discrete 3D displacement. The displacement estimation obtained by using epipolar constraint works well when the displacement between the two images is relatively large. However, if the velocity of the moving camera is low or we use a high frame rate, the relative displacement between two consecutive images might become small. In this case, the algorithm become singular due to the short translation and the estimation results are less reliable.

A differential version of the 3D motion estimation is to recover the 3D velocity of the camera from the optical flow. The optical flow is the velocity of pixels in the image plane in an image sequence. This problem has been explored by many authors such as Prazdny [Prazdny 80], Bruss and Horn [Bruss 83], Zhuang et al [Zhuang 84], and Kanatani [Kanatani 93b], for example.

Approaches to motion estimation can be classified into discrete and differential methods depending on whether they use a set of point correspondences or the optical flow [Armangué 03a]. Another possible classification takes into account the estimation techniques used for motion recovery (linear or nonlinear techniques). In table 4.1, the algorithms are summarized and classified in terms of their nature (discrete and differential case), and estimation method (linear and nonlinear technique).

This chapter analyzes several different algorithms for camera motion estimation based on differential image motion. The surveyed methods have also been compared with them. Moreover, this chapter analyzes the adaptation of general methods used in free 3D movement to planar motion which corresponds to the common case of a

Table 4.1: Motion recovery methods

Discrete Case	**Differential Case**
	Linear techniques
Longuet-Higgins [Longuet-Higgins 81]	Zhuang, Huang, Ahuja, Haralick [Zhuang 84, Zhuang 88][1]
Tsai, Huang [Tsai 84]	Heeger, Jepson [Heeger 92, Jepson 91, Jepson 93]
Toscani, Faugeras [Toscani 86]	Kanatani [Kanatani 93b, Kanatani 93a][1]
Tomasi, Kanade [Tomasi 92]	Tomasi, Shi [Tomasi 93, Shi 94]
	Brooks, Chojnacki, Hengel, Baumela [Brooks 98][1]
	• Seven Points
	• Least Squares
	• Iteratively Reweighted Least Squares
	• Modified Iteratively Reweighted Least Squares
	• Least Median Squares
	Ma, Košecká, Sastry [Ma 00, Ma 98a][1]
	Baumela, Agapito, Bustos, Reid [Baumela 00][1]
	Nonlinear techniques
Horn [Horn 90]	Prazdny [Prazdny 80, Prazdny 81]
Weng, Ahuja, Huang [Weng 92a]	Bruss, Horn [Bruss 83]
Taylor, Kriegman [Taylor 95]	Zhang, Tomasi [Zhang 99]
Soatto, Brockett [Soatto 98]	
Ma, Košecká, Sastry [Ma 98b]	

robot moving on a plane with the aim of studying how much accuracy improves by constraining the camera movement [Armangué 02]. Hence, this chapter focuses on linear techniques, as the motion has to be recovered in real-time.

This chapter is structured as follows. Section 4.2 presents similarities and differences between discrete and differential epipolar constraint. Then, section 4.3 deals with motion recovery by means of the discrete epipolar constraint. Section 4.4 describes twelve algorithms for 3D motion estimation based on optical flow. Section 4.5 focuses on the estimation of planar motion by constraining the free movement explained in the previous section. The chapter ends with a summary.

4.2 Discrete vs. Differential Epipolar Constraint

The following sections describe two different approaches for motion recovery. First, we explain the common case of the discrete epipolar geometry formulated by Longuet-Higgins [Longuet-Higgins 81] using the fundamental matrix and a set of corresponding points. Then, differential epipolar constraint is detailed and compared with a

---

[1] These methods are based on the differential epipolar constraint.

discrete approach find similarities and differences. This constraint incorporates two matrices which encode information about the ego-motion of the camera using any sufficiently large subset of an optical flow field.

4.2.1 Discrete Epipolar Constraint

The relative 3D displacement between both camera views is recovered by the epipolar constraint from a set of correspondences in both image planes (see section 3.3). All the epipolar geometry is contained in the so-called fundamental matrix, as shown in equation (4.1).

$$m^{\mathrm{T}}\,\mathbf{F}m' = 0 \tag{4.1}$$

where the fundamental matrix depends on the intrinsic parameters of both cameras and the rigid transformation between them,

$$\mathbf{F} = {}^{I}\mathbf{A}_{C}{}^{-\mathrm{T}}\,{}^{C}\mathbf{R}_{C'}{}^{\mathrm{T}}\,{}^{C}\hat{T}_{C'}\,{}^{I'}\mathbf{A}_{C'}{}^{-1}. \tag{4.2}$$

See section 3.4 for a detailed survey on computing **F**. When the intrinsic camera parameters are known, it is possible to simplify equations (4.1) and (4.2), obtaining

$$q^{T}\mathbf{E}\,q' = 0 \tag{4.3}$$

where

$$q = {}^{I}\mathbf{A}_{C}{}^{-1}m, \qquad \mathbf{E} = {}^{C}\mathbf{R}_{C'}{}^{\mathrm{T}}\,{}^{C}\hat{T}_{C'}, \qquad q' = {}^{I'}\mathbf{A}_{C'}{}^{-1}m' \quad . \tag{4.4}$$

Matrix **E** is called essential [Huang 89] and will be used to extract the motion of a moving camera.

4.2.2 Differential Epipolar Constraint

The differential case is the infinitesimal version of the discrete case, in which both views are always given by a single moving camera. If the velocity of the camera is low enough and the frame rate is very high, the relative displacement between two consecutive images becomes very small. The 2D displacement of image points can then be obtained from an image sequence using the optical flow. In this case,

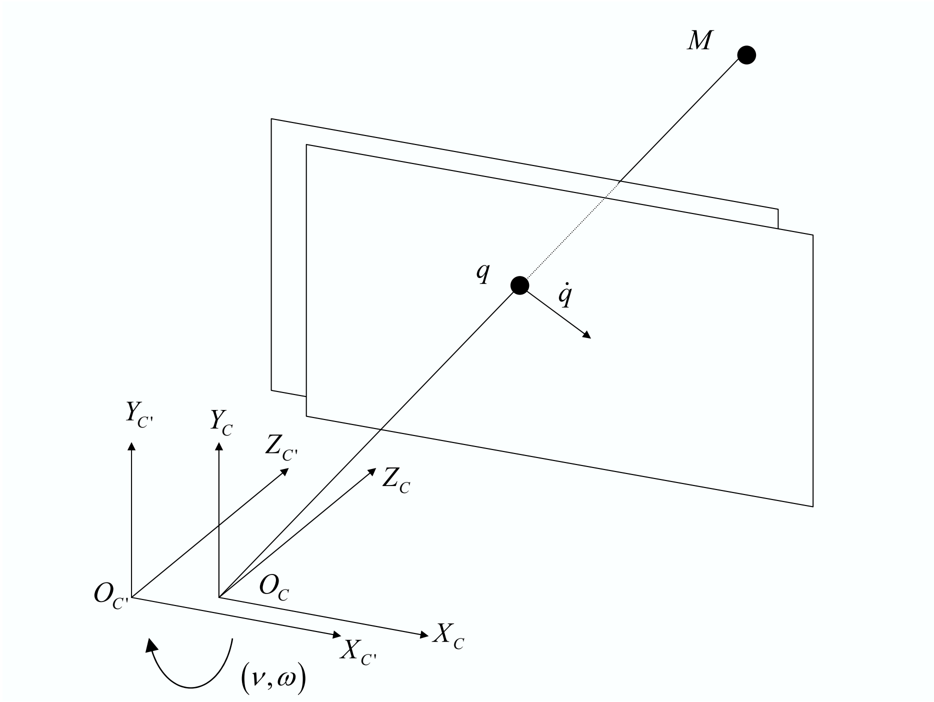


Figure 4.1: Differential epipolar case.

the 3D camera motion is described by a rigid motion using a rotation matrix and a translation vector, as in

$$M(t) = \mathbf{R}(t)\, M(0) + T(t) \tag{4.5}$$

where differentiating

$$\dot{M}(t) = \dot{\mathbf{R}}(t)\, M(0) + \dot{T}(t) \quad . \tag{4.6}$$

Then, replacing the parameter $M(0)$ to $\mathbf{R}^{-1}(t)\,[M(t) - T(t)]$ in equation (4.6), the following equation is obtained,

$$\dot{M}(t) = \dot{\mathbf{R}}(t)\, \mathbf{R}^{-1}(t)\, M(t) + \dot{T}(t) - \dot{\mathbf{R}}(t)\, \mathbf{R}^{-1}(t)\, T(t) \quad , \tag{4.7}$$

which leads to the following differential epipolar constraint,

$$q^{\mathrm{T}}\, \hat{\upsilon}\dot{q} + q^{\mathrm{T}}\, \hat{\omega}\hat{\upsilon}\, q = 0 \tag{4.8}$$

where, $\omega = (\omega_1, \omega_2, \omega_3)^{\mathrm{T}}$ is the angular velocity of the camera and $\upsilon = (\upsilon_1, \upsilon_2, \upsilon_3)^{\mathrm{T}}$ is the linear velocity of the camera. By projecting M and $\dot{M}$ in the image plane, q in the camera coordinates and its corresponding optical flow $\dot{q}$ are obtained.

For a complete demonstration, the reader is directed to Haralick's book, chapter 15 [Haralick 92b], where the movement of a rigid body in relation to a camera is explained. In our case, the demonstration is used to describe the movement of a camera in relation to a static object, in which only the sign of the obtained velocities differs from the previous one. Nevertheless, equation (4.8) can also be demonstrated in different ways as explained by Viéville [Viéville 96] and Brooks [Brooks 97]. Also, another equivalent form of equation (4.8) is shown in equation (4.9). In this case, since matrix **S** is symmetric, the number of unknowns is reduced to six.

$$q^{\mathrm{T}}\, \hat{v}\dot{q} + q^{\mathrm{T}}\, \mathbf{S}\, q = 0 \tag{4.9}$$

where

$$\mathbf{S} = \frac{1}{2}\left(\hat{\omega}\hat{v} + \hat{v}\hat{\omega}\right) \quad . \tag{4.10}$$

The existence of two forms to express the differential epipolar constraint (equations (4.8) and (4.9)) indicates that a redundancy exists in equation (4.8) (for a demonstration see Viéville [Viéville 96], Brooks [Brooks 97] and Ma [Ma 00]). Several books describe optical flow, such as Trucco and Verri [Trucco 98], and the article published by Barron et al. [Barron 92] gives a state-of-the-art in optical flow estimation.

When comparing the discrete and differential methods, the discrete epipolar equation incorporates a single matrix, whereas the differential epipolar equation incorporates two matrices. These matrices encode information about the linear and angular velocities of the camera [Haralick 92b].

4.3 Discrete Motion Estimation

This approach was originally developed by Huang et al. [Huang 89] and is based on motion recovery between two camera views from essential matrix, described in equation (4.4). Huang and Faugeras [Huang 89] established that matrix **E** is an essential matrix if the singular value decomposition (SVD) of $\mathbf{E} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathrm{T}}$ forces

$$\mathbf{D} = \mathrm{diag}\{\lambda, \lambda, 0\} \tag{4.11}$$

for some $\lambda > 0$. This work also demonstrates that there are two pairs of $\mathbf{R}$ and T which satisfy an essential matrix $\mathbf{E}$ such as $\mathbf{R}^{\mathrm{T}}\hat{T} = \mathbf{E}$. The following equations give the two solutions:

$$\begin{array}{rclcrcl} \mathbf{R}_1 & = & \mathbf{U}\mathbf{R}_{\mathbf{Z}}(+\frac{\pi}{2})^{\mathrm{T}}\,\mathbf{V}^{\mathrm{T}}\,, & & \hat{T}_1 & = & \mathbf{V}\mathbf{R}_{\mathbf{Z}}(+\frac{\pi}{2})\mathbf{D}\mathbf{V}^{\mathrm{T}} \\ \mathbf{R}_2 & = & \mathbf{U}\mathbf{R}_{\mathbf{Z}}(-\frac{\pi}{2})^{\mathrm{T}}\,\mathbf{V}^{\mathrm{T}}\,, & & \hat{T}_2 & = & \mathbf{V}\mathbf{R}_{\mathbf{Z}}(-\frac{\pi}{2})\mathbf{D}\mathbf{V}^{\mathrm{T}} \end{array} \tag{4.12}$$

where $\mathbf{R}_{\mathbf{Z}}(\theta)$ is defined as a rotation matrix around the Z axis by an angle θ.

The estimation of the essential matrix can only recover up to an arbitrary scale factor, so many matrices satisfy the epipolar constraint. In particular, both $\mathbf{E}$ and $-\mathbf{E}$ are solutions of the same equation. In general, four solutions of $\mathbf{R}$ and T will be obtained from a set of image correspondences. However, it is possible to reduce this ambiguity by imposing the positive depth constraint to fit the problem into just two solutions. Following this paragraph we present the three-step algorithm proposed by Maybank [Maybank 93] to estimate $\mathbf{R}$ and T from $\mathbf{E}$.

The three step SVD based displacement estimation algorithm:

1. *Estimate the essential matrix*: For a set of image correspondences find the matrix $\mathbf{E}$ which minimizes

$$\min_{\mathbf{E}} \sum_i \left(q_i^{\mathrm{T}}\mathbf{E}q_i'\right)^2 \tag{4.13}$$

   such as $\|\mathbf{E}\| = 1$ (see Chapter 3 for more details about fundamental matrix estimators).

2. *Use a singular value decomposition in the essential matrix*: Find the singular value decomposition of the matrix $\mathbf{E}$.

$$\mathbf{E} = \mathbf{U}\,\mathrm{diag}\{\sigma_1, \sigma_2, \sigma_3\}\mathbf{V}^{\mathrm{T}} \tag{4.14}$$

   where $\sigma_1 \geq \sigma_2 \geq \sigma_3$.

3. *Recover 3D displacement*: We define the diagonal matrix as

$$\mathbf{D}' = \mathrm{diag}\{\sigma_1', \sigma_2', 0\} \tag{4.15}$$

   to project onto the essential manifold where $\sigma_1' = \frac{\sigma_1+\sigma_2}{2}$ and $\sigma_2' = \frac{\sigma_1+\sigma_2}{2}$ to obtain the two solutions of 3D displacement as

$$\begin{array}{rclrcl} \mathbf{R}_1 & = & \mathbf{U}\mathbf{R}_\mathbf{Z}(+\frac{\pi}{2})^\mathrm{T}\,\mathbf{V}^\mathrm{T}\,, & \hat{T}_1 & = & \mathbf{V}\mathbf{R}_\mathbf{Z}(+\frac{\pi}{2})\mathbf{D}'\mathbf{V}^\mathrm{T} \\ \mathbf{R}_2 & = & \mathbf{U}\mathbf{R}_\mathbf{Z}(-\frac{\pi}{2})^\mathrm{T}\,\mathbf{V}^\mathrm{T}\,, & \hat{T}_2 & = & \mathbf{V}\mathbf{R}_\mathbf{Z}(-\frac{\pi}{2})\mathbf{D}'\mathbf{V}^\mathrm{T} \quad . \end{array} \tag{4.16}$$

4.4 Overview of Differential Motion Estimation

In this section, we detail some methods used for the recovery of every 6-DOF[2] motion parameter from optical flow, providing insights into the complexity of the problem. The surveyed methods have been classified considering whether they are based on the Differential Epipolar Constraint or not.

4.4.1 Methods Based on the Differential Epipolar Constraint

The methods based on the Differential Epipolar Constraint deal with the minimization of the following criteria,

$$\min_{\upsilon,\mathbf{S}} \sum_{i=1}^{n} \left(q_i{}^\mathrm{T}\,\hat{\upsilon}\dot{q}_i + q_i{}^\mathrm{T}\,\mathbf{S}\,q_i\right)^2 \quad . \tag{4.17}$$

Rewriting equation (4.17) in matrix form,

$$\min_{\theta} \|\mathbf{U}\,\theta\|^2 \tag{4.18}$$

where,

$$\theta = \Big(\upsilon_1, \upsilon_2, \upsilon_3, s_{11}, s_{12}, s_{13}, s_{22}, s_{23}, s_{33}\Big)^\mathrm{T} \quad , \tag{4.19}$$

$$\mathbf{U} = \Big(u_1, u_2, \dots, u_n\Big)^\mathrm{T} \tag{4.20}$$

in which

$$\begin{array}{rcl} u_i & = & \big(\dot{q}_{i_2}q_{i_3} - \dot{q}_{i_3}q_{i_2}, \dot{q}_{i_3}q_{i_1} - \dot{q}_{i_1}q_{i_3}, \dot{q}_{i_1}q_{i_2} - \dot{q}_{i_2}q_{i_1}, q_{i_1}{}^2, \\ & & 2q_{i_1}q_{i_2}, 2q_{i_1}q_{i_3}, q_{i_2}{}^2, 2q_{i_2}q_{i_3}, q_{i_3}{}^2\big)^\mathrm{T} \quad . \end{array} \tag{4.21}$$

Hereafter, 7 different methods based on such minimization criteria are consecu-

---

[2] Degrees Of Freedom

tively explained in the following paragraphs.

Seven Points Estimator: This method was proposed by Brooks [Brooks 98] but not entirely, since a similar method to estimate the fundamental matrix has been used by other authors [Zhang 98, Torr 97].

Using seven points, through singular value decomposition ($\mathbf{U} = \mathbf{V_1 D V_2}^{\mathrm{T}}$), vectors θ_1 and θ_2 are obtained from the two last columns of $\mathbf{V_2}$. The solution is a linear combination of θ_1 and θ_2, $\theta = \alpha\,\theta_1 + (1-\alpha)\theta_2$ which corresponds to υ and $\mathbf{S}$. Substituting θ into $\upsilon^{\mathrm{T}}\,\mathbf{S}\,\upsilon = 0$ (for details see [Brooks 97]) a cubic equation on α is obtained. This equation could have one or three solutions for υ and $\mathbf{S}$.

The main advantage of this method is that υ and $\mathbf{S}$ can be estimated using only seven points, but this can also be a drawback when some of the points are corrupted by noise. Moreover, the 7-points method cannot be applied in the presence of redundancy. Hence, it can not be applied using n points where $n > 7$.

Least Squares Estimator using Eigen Analysis: When there are n points, where $n \geq 8$, equation (4.17) goes into redundancy, in which $\theta = 0$ is the trivial solution. Then, some constraints have to be established in order to avoid the null solution. A general constraint is to fix $\|\theta\| = 1$. Then, equation (4.18) is rewritten as,

$$\min_{\|\theta\|=1} \theta^{\mathrm{T}}\ \mathbf{U}^{\mathrm{T}}\,\mathbf{U}\,\theta \tag{4.22}$$

Introducing the Lagrange multiplier λ,

$$\mathcal{L}(\theta) \;=\; \theta^{\mathrm{T}}\ \mathbf{U}^{\mathrm{T}}\,\mathbf{U}\,\theta\ -\lambda(\theta^{\mathrm{T}}\,\theta - 1) \tag{4.23}$$

Equation (4.22) is equivalent to minimize,

$$\min_{\|\theta\|=1} \mathbf{U}^{\mathrm{T}}\,\mathbf{U}\,\theta\ -\lambda\theta \tag{4.24}$$

Thus, the solution θ must be the eigenvector of the $9x9$ matrix $\mathbf{X} = \mathbf{U}^{\mathrm{T}}\,\mathbf{U}$ corresponding to the smallest eigenvalue λ_0 (more details in [Trucco 98] Appendix A.6).

Iteratively Reweighted Least Squares Estimator: This method is an evolution of the previous Seven-Point Estimator proposed by Brooks [Brooks 98]. In this case, θ is computed by using an iterative approximation method where all points

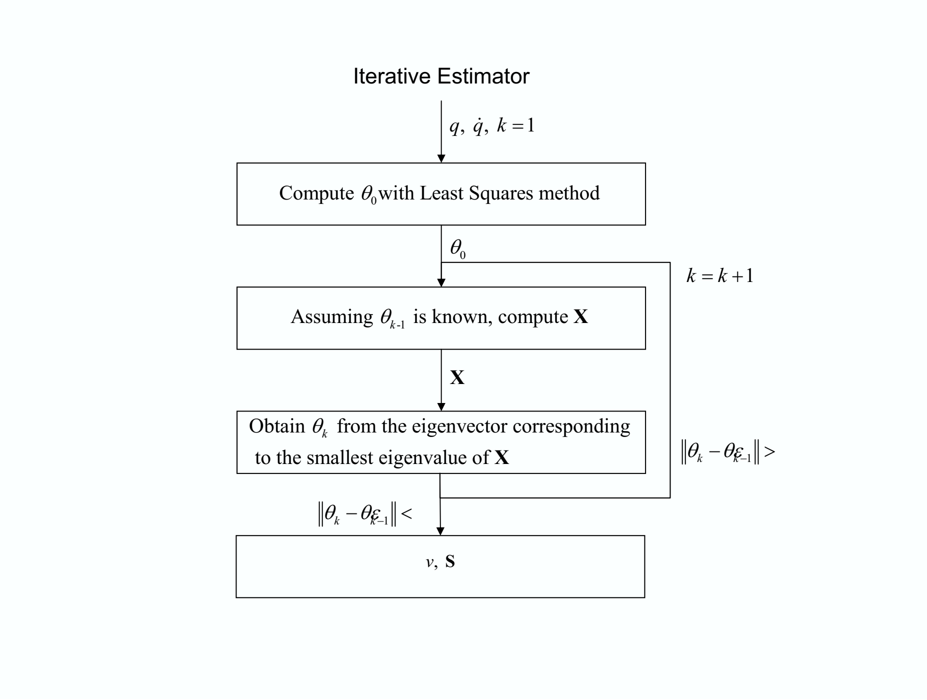


Figure 4.2: Iterative estimator procedure.

are reweighted in each iteration. The equation to minimize is

$$\min_{v,\mathbf{S}} \sum_{i=1}^{n} \left(\frac{|{q_i}^{\mathrm{T}}\, \hat{v}\dot{q}_i + {q_i}^{\mathrm{T}}\, \mathbf{S}\, q_i|^2}{\sqrt{\|2\,\mathbf{S}\, q_i + \hat{v}\dot{q}_i\|^2 + \|\hat{v}q_i\|^2}} \right)^2 \tag{4.25}$$

which can be rewritten in matrix form obtaining,

$$\min_{\|\theta\|=1} \theta^{\mathrm{T}}\, w^{\mathrm{T}}\ \mathbf{U}^{\mathrm{T}}\, \mathbf{U}\, \theta \tag{4.26}$$

$$\min_{\|\theta\|=1} w^{\mathrm{T}}\ \mathbf{U}^{\mathrm{T}}\, \mathbf{U}\, \theta\ - \lambda\theta \tag{4.27}$$

where $w = (w_1, \ldots, w_n)^{\mathrm{T}}$ and $w_i = (\|2\,\mathbf{S}\, q_i + \hat{v}\dot{q}_i\|^2 + \|\hat{v}q_i\|^2)^{-1}$. Employing Lagrange multipliers, as in the previous method, the solution θ is the eigenvector corresponding to the smallest eigenvalue of $\mathbf{X} = w^{\mathrm{T}}\ \mathbf{U}^{\mathrm{T}}\ \mathbf{U}$. Then, the proposed iterative method is shown in Figure 4.2.

Modified Iteratively Reweighted Least Squares Estimator: This technique, surveyed by Brooks [Brooks 98], is similar to the precedent method but is based on a different minimization of equation (4.25). Starting from the reweighted equation w_i, we can express it in matrix form as shown in equation (4.28).

$$w_i^{-1} = \|2\,\mathbf{S}\,q_i + \hat{\upsilon}\dot{q}_i\|^2 + \|\hat{\upsilon}q_i\|^2 = \theta^{\mathrm{T}}\,\mathbf{N}_i\theta \tag{4.28}$$

in which

$$\mathbf{N}_i = 4\sum_{\alpha=1}^{3}\boldsymbol{\rho}_\alpha{}^{\mathrm{T}}\,q_iq_i{}^{\mathrm{T}}\,\boldsymbol{\rho}_\alpha + 4\sum_{\alpha=1}^{3}\boldsymbol{\rho}_\alpha{}^{\mathrm{T}}\,q_i\dot{q}_i{}^{\mathrm{T}}\,\boldsymbol{\sigma}_\alpha - \sum_{\alpha=1}^{3}\boldsymbol{\sigma}_\alpha{}^{\mathrm{T}}\,\dot{q}_i\dot{q}_i{}^{\mathrm{T}}\,\boldsymbol{\sigma}_\alpha - \sum_{\alpha=1}^{3}\boldsymbol{\sigma}_\alpha{}^{\mathrm{T}}\,q_iq_i{}^{\mathrm{T}}\,\boldsymbol{\sigma}_\alpha \quad . \tag{4.29}$$

Where

$$\begin{aligned}
\boldsymbol{\rho}_1 &= \begin{pmatrix}1&0&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0&0\\0&0&1&0&0&0&0&0&0\end{pmatrix}, &\quad \boldsymbol{\sigma}_1 &= \begin{pmatrix}0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&1&0&0\\0&0&0&0&0&0&0&1&0\end{pmatrix},\\
\boldsymbol{\rho}_2 &= \begin{pmatrix}0&1&0&0&0&0&0&0&0\\0&0&0&1&0&0&0&0&0\\0&0&0&0&1&0&0&0&0\end{pmatrix}, &\quad \boldsymbol{\sigma}_2 &= \begin{pmatrix}0&0&0&0&0&0&-1&0&0\\0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&1\end{pmatrix},\\
\boldsymbol{\rho}_3 &= \begin{pmatrix}0&0&1&0&0&0&0&0&0\\0&0&0&0&1&0&0&0&0\\0&0&0&0&0&1&0&0&0\end{pmatrix}, &\quad \boldsymbol{\sigma}_3 &= \begin{pmatrix}0&0&0&0&0&0&0&-1&0\\0&0&0&0&0&0&0&0&-1\\0&0&0&0&0&0&0&0&0\end{pmatrix}.
\end{aligned} \tag{4.30}$$

After that, equation (4.26) is reorganized as,

$$\min_{\|\theta\|=1} 2\,\mathbf{X}\,\theta \tag{4.31}$$

where,

$$\mathbf{X} = \sum_{i=1}^{n}\frac{\mathbf{M}_i}{\theta^{\mathrm{T}}\;\mathbf{N}_i\,\theta} - \sum_{i=1}^{n}\frac{\theta^{\mathrm{T}}\;\mathbf{M}_i\,\theta}{(\theta^{\mathrm{T}}\;\mathbf{N}_i\,\theta)^2}\mathbf{N}_i \tag{4.32}$$

$$\mathbf{M}_i = u_iu_i^{\mathrm{T}} \quad . \tag{4.33}$$

By using eigen analysis, a solution of θ is obtained. The procedure proposed is shown in Figure 4.2.

Least Median Squares Estimator: The previous methods assumes that image points can only present a gaussian noise in its localization in the image plane. LMedS is considered a robust estimator as the method does not use all data because it assumes that a set of points could present a matching error with its correspondent. The points with an erroneous matching are called outliers and the rest are called inliers. LMedS is able to recognize the outliers and keep them out of the computation. LMedS is considered a statistical method which was first proposed by Rousseeuw [Rousseeuw 87] but later used in several other applications, such as Fundamental matrix estimation [Zhang 98].

First, this technique is based on selecting a number of sets made up of seven random points which are used to compute approximations of θ by using the 7-points method. It is important to assure that sets are made up of points evenly spread throughout the image. The number of sets considered depends on the estimated outliers ratio and the probability that at least one set is free of outliers. This probability is given by $P = 1 - (1 - (1 - \epsilon)^s)^p$, where ϵ is the maximum outlier ratio, s is the number of elements in each set sample (seven in our case) and p is the number of set samples. Arranging the terms, we obtain

$$p = \left\lceil \frac{\log(1 - P)}{\log(1 - (1 - \epsilon)^s)} \right\rceil \quad . \tag{4.34}$$

The LMedS method calculates the median of algebraic residual for each θ using all data, where the chosen θ has to minimize this median. This θ is used to identify the outliers. Finally, when the outliers are removed, a non-robust method using all the remaining points is used to estimate the best solution.

The complete algorithm is the following:

1. Choose p sets of seven points evenly spread throughout the image.
2. For each set, obtain an estimation of θ using the 7-Point method. We obtain v_j and $\mathbf{S}_j$, where $j = 1, \ldots, p$.
3. For each estimation, compute the algebraic residual and determine the median as,
$$m_j = \operatorname*{median}_{i=1,\ldots,n} \left(\frac{|{q_i}^{\mathrm{T}}\, \hat{v}_j \dot{q}_i + {q_i}^{\mathrm{T}}\, \mathbf{S}_j\, q_i|^2}{\sqrt{\|2\, \mathbf{S}_j\, q_i + \hat{v}_j \dot{q}_i\|^2 + \|\hat{v}_j q_i\|^2}} \right)^2 \quad . \tag{4.35}$$
4. From every estimation m_j, take the m_k in which the median is the minimum: $m_k = \min_{j=1,\ldots,p} m_j$.
5. Compute the robust standard deviation $\hat{\sigma} = 1.4826 \left(1 + \frac{5}{n-7}\sqrt{m_k}\right)$.

6. Then, q_i is considered an outlier only if $\left(\frac{|{q_i}^{\mathrm{T}}\, \hat{\upsilon}_k \dot{q}_i + {q_i}^{\mathrm{T}}\, \mathbf{S}_k\, q_i|^2}{\sqrt{\|2\,\mathbf{S}_k\, q_i + \hat{\upsilon}_k \dot{q}_i\|^2 + \|\hat{\upsilon}_k q_i\|^2}} \right)^2 > (2.5\hat{\sigma})^2$.
7. Finally, recompute υ and $\mathbf{S}$ considering only the inliers by using one of the previous methods.

Ma, Košecká and Sastry Estimator: The estimator proposed by Ma, Košecká and Sastry [Ma 00, Ma 98a] estimates not only the linear velocity υ and the symmetric matrix $\mathbf{S}$ like the previous methods, but also the angular velocity ω which is computed from $\mathbf{S}$.

The computation of the 3D velocity (υ, ω) is divided into four steps:

1. Estimate υ_0 and $\mathbf{S}_0$. Obtain υ_0 and $\mathbf{S}_0$, minimizing the error function using least squares estimator by means of eigen analysis.

$$\min_{\theta_0} \|\mathbf{U}\theta_0\|^2 \tag{4.36}$$

2. Recover the special symmetric matrix. The symmetric matrix $\mathbf{S}$ obtained in the previous step probably does not have the form $\mathbf{S} = \frac{1}{2}(\hat{\omega}\hat{\upsilon} + \hat{\upsilon}\hat{\omega})$ called *special symmetric matrix* by the authors. Therefore, diagonalize the symmetric matrix $\mathbf{S}_0$ using eigenvalue decomposition:

$$\mathbf{S}_0 = \mathbf{V}_1 \operatorname{diag}\{\lambda_1, \lambda_2, \lambda_3\} {\mathbf{V}_1}^{\mathrm{T}} \tag{4.37}$$

with $\lambda_1 \geq \lambda_2 \geq \lambda_3$. Project the symmetric matrix onto the special symmetric matrix.

$$\mathbf{S} = \mathbf{V}_1 \operatorname{diag}\{\sigma_1, \sigma_2, \sigma_3\} {\mathbf{V}_1}^{\mathrm{T}} \tag{4.38}$$

where,

$$\sigma_1 = \tfrac{2\lambda_1 + \lambda_2 - \lambda_3}{3}, \quad \sigma_2 = \tfrac{\lambda_1 + 2\lambda_2 + \lambda_3}{3} \quad \text{and} \quad \sigma_3 = \tfrac{2\lambda_3 + \lambda_2 - \lambda_1}{3} \quad . \tag{4.39}$$

3. Recover the linear and angular velocities from the special symmetric matrix, obtaining four solutions:

$$\begin{array}{rclrcl} \hat{\omega}_{1,2} & = & \mathbf{U}_2\mathbf{R}_{\mathbf{Z}}(\pm\frac{\pi}{2}) \operatorname{diag}\{\lambda, \lambda, 0\} {\mathbf{U}_2}^{\mathrm{T}}\,, & \hat{\upsilon}_{1,2} & = & \mathbf{V}_2\mathbf{R}_{\mathbf{Z}}(\pm\frac{\pi}{2}) \operatorname{diag}\{1, 1, 0\} {\mathbf{V}_2}^{\mathrm{T}} \\ \hat{\omega}_{3,4} & = & \mathbf{V}_2\mathbf{R}_{\mathbf{Z}}(\pm\frac{\pi}{2}) \operatorname{diag}\{\lambda, \lambda, 0\} {\mathbf{V}_2}^{\mathrm{T}}\,, & \hat{\upsilon}_{3,4} & = & \mathbf{U}_2\mathbf{R}_{\mathbf{Z}}(\pm\frac{\pi}{2}) \operatorname{diag}\{1, 1, 0\} {\mathbf{U}_2}^{\mathrm{T}} \end{array} \tag{4.40}$$

where $\mathbf{U}_2 = -\mathbf{V}_2\mathbf{R_Y}(\psi)$, $\mathbf{V}_2 = \mathbf{V}_1\mathbf{R_Y}^{\mathrm{T}}\,({}^{\psi}\!/_2 - {}^{\pi}\!/_2)$, $\lambda = \sigma_1 - \sigma_3 \geq 0$ and $\psi = \arccos(-\sigma_2/\lambda) \in [0, \pi]$.

4. Recover the linear velocity. It is necessary to choose one velocity from the four available. Choose the one which accomplishes

$$v_k^{\mathrm{T}}\, v_0 = \max_{i=1,\ldots,4} v_i^{\mathrm{T}}\, v_0 \quad . \tag{4.41}$$

Then the 3D velocity estimated is $v = v_0$ and $\omega = \omega_k$.

Baumela, Agapito, Bustos and Reid Estimator: This motion estimator was proposed by Baumela et al. [Baumela 00] and is based on using information from the uncertainty of the optical flow. This algorithm is based on the assumption that the optical flow estimation produce exact values for q and noisy estimations for $\dot{q}$.

By starting from equation (4.9) and knowing that $q = (q_1, q_2, 1)^{\mathrm{T}}$ and $\dot{q} = (\dot{q}_1, \dot{q}_2, 0)^{\mathrm{T}}$, we obtain

$$a\dot{q}_1 + b\dot{q}_2 + c = 0 \tag{4.42}$$

where $a = v_3q_2 - v_2$, $b = v_1 - v_3q_1$ and $c = s_{11}{q_1}^2 + 2s_{12}q_1q_2 + 2s_{13}q_1 + s_{22}{q_2}^2 + 2s_{23}q_2 + s_{33}$. Then the algorithm is based on minimizing the distance between equation (4.42) and the optical flow for all the points

$$\min_{v,\mathbf{S}} \sum_{i=1}^{n} \frac{(a_i\dot{q}_{i_1} + b_i\dot{q}_{i_2} + c_i)^2}{{a_i}^2 + {b_i}^2} \quad . \tag{4.43}$$

The uncertainty of each flow measurement can be expressed by the covariance matrix $\mathbf{\Sigma}_{\dot{q}_i}$, consequently equation (4.43) becomes

$$\min_{\theta} \sum_{i=1}^{n} \frac{\theta^{\mathrm{T}}\, f_i\, {f_i}^{\mathrm{T}}\, \theta}{v^{\mathrm{T}}\, \mathbf{H}_i\mathbf{\Sigma}_{\dot{q}_i}{\mathbf{H}_i}^{\mathrm{T}}\, v} \tag{4.44}$$

where,

$$f_i = \Big(\dot{q}_{i_2}, -\dot{q}_{i_1}, \dot{q}_{i_1}q_{i_2} - \dot{q}_{i_2}q_{i_1}, {q_{i_1}}^2, 2q_{i_1}q_{i_2}, 2q_{i_1}, {q_{i_2}}^2, 2q_{i_2}, 1\Big)^{\mathrm{T}} \tag{4.45}$$

$${\mathbf{H}_i}^{\mathrm{T}} = \begin{pmatrix} 1 & 0 & -q_{i_2} \\ 0 & -1 & q_{i_1} \end{pmatrix} \quad . \tag{4.46}$$

It is necessary to impose the constraint $\upsilon^{\mathrm{T}}\, \mathbf{S}\upsilon = 0$ in the minimization of equation (4.44). The possibility proposed by the authors of the method is to substitute s_{33} to the explicit value and employ an iterative minimization method. In this case, equation (4.44) is rewritten in the following way,

$$\min_{\delta} \sum_{i=1}^{n} \frac{\delta^{\mathrm{T}}\, g_i\, {g_i}^{\mathrm{T}}\, \delta}{\upsilon^{\mathrm{T}}\, \mathbf{H}_i \mathbf{\Sigma}_{\dot{q}_i} {\mathbf{H}_i}^{\mathrm{T}}\, \upsilon} \tag{4.47}$$

where,

$$\begin{aligned} g_i \;=\; & \Bigg(\dot{q}_{i_2}, -\dot{q}_{i_1}, \dot{q}_{i_1} q_{i_2} - \dot{q}_{i_2} q_{i_1}, {q_{i_1}}^2 - \frac{{\upsilon_1}^2}{{\upsilon_3}^2}, \\ & 2(q_{i_1} q_{i_2} - \frac{\upsilon_1 \upsilon_2}{{\upsilon_3}^2}), 2(q_{i_1} - \frac{\upsilon_1}{\upsilon_3}), {q_{i_2}}^2 - \frac{{\upsilon_2}^2}{{\upsilon_3}^2}, 2(q_{i_2} - \frac{\upsilon_2}{\upsilon_3})\Bigg)^{\mathrm{T}} \quad , \end{aligned} \tag{4.48}$$

$$\delta = (\upsilon_1, \upsilon_2, \upsilon_3, s_{11}, s_{12}, s_{13}, s_{22}, s_{23})^{\mathrm{T}} \quad , \tag{4.49}$$

and

$$s_6 = -\frac{s_{11}{\upsilon_1}^2 + 2 s_{12} \upsilon_1 \upsilon_2 + s_{22}{\upsilon_2}^2}{{\upsilon_3}^2} - \frac{2 s_{13} \upsilon_1 + 2 s_{23} \upsilon_2}{\upsilon_3} \quad . \tag{4.50}$$

4.4.2 Methods Directly Based on the Optical Flow

Hereafter, 5 different methods which are based directly on the optical flow instead of using the Differential Epipolar constraint are explained. Such methods have been included in the chapter with the aim of providing experimental results covering the whole field of differential egomotion estimation.

Bruss and Horn Estimator: Bruss and Horn [Bruss 83] proposed a method not based on the differential epipolar principle as are the previously described methods. Instead, their method is based on the following equation:

$$\dot{q} = \begin{pmatrix} 1 & 0 & -q_1 \\ 0 & 1 & -q_2 \end{pmatrix} \left(\frac{\upsilon}{Z(q)} + \omega \times q \right) \quad . \tag{4.51}$$

This equation relates the camera velocity with respect to the static scene in which q is the image point, $\dot{q}$ is the point velocity, $Z(q)$ is the point depth related to the camera coordinate system and, finally, υ and ω are the linear and angular camera velocities, respectively.

Then a bilinear constraint can be imposed on the linear υ and angular ω velocities of every pixel in order to remove point depth $Z(q)$ by means of a few algebraic transformations in equation (4.51) described in the following equation.

$$\upsilon^{\mathrm{T}} \left(q \times \dot{q}\right) + \left(\upsilon \times q\right)^{\mathrm{T}} \left(q \times \omega\right) = 0 \tag{4.52}$$

However, the following stages are required. First, an initial guess of the translation which can be estimated by using least-squares technique is needed. Second, a non-linear minimization of equation (4.52) is applied to every image pixel with the constraint $\|\upsilon\| = 1$ with the aim of obtaining υ. Finally, the rotation velocity can be extracted linearly from equation (4.52).

Prazdny Estimator: The method proposed by Prazdny [Prazdny 81] was one of the first to estimate camera movement from optical flow, so it is not based on the epipolar constraint either. This method differs from the previous one because it estimates the rotating velocity first instead of computing the linear velocity. Then a triplet of image points can be obtained from equation (4.51) and by using some algebraic transformations the following equation is given.

$$n_3 \left(n_1 \times n_2\right) = 0 \tag{4.53}$$

where $n_i = (\omega \times q_i + \dot{q}_i) \times q_i = 0$. Once the rotating velocity ω is computed from equation (4.53), the linear velocity is given by equation (4.52).

Heeger and Jepson Estimator: Heeger and Jepson proposed the so-called linear subspace method [Heeger 92, Jepson 91]. Given the optical flow of a set of n image points, the following relationship can be formulated.

$$\tau_i = \sum_{k=1}^{n} c_{ik}(\dot{q}_k \times q_k) \tag{4.54}$$

where vector τ_i is orthogonal to υ. Moreover, $c_i = (c_{i1}, \ldots, c_{in})$ has to be chosen so that it is orthogonal to both quadratic polynomiums q_{k_1} and q_{k_2} with the aim of removing the rotating velocity from the images. Note that given n points, $n - 6$ vectors τ_i are generated. Then υ corresponds to the eigenvector associated to the smallest eigenvalue of $\sum \tau_i {\tau_i}^{\mathrm{T}}$.

Tomasi and Shi Estimator: The method proposed by Tomasi and Shi [Shi 94, Tomasi 93] estimates υ from image deformations. Their method estimates translation from image deformations, defined as the change $\dot{\alpha}$ in the angular distance $\alpha = \arccos(q_i\, q_j)$ between two image points due to camera movement. Image deformations do not depend on the camera rotating movement so the following bilinear equation can be extracted. Note that this equation is only a function of υ and both depth points $Z(q_i)$ and $Z(q_j)$,

$$\dot{\alpha} = \sin\alpha(Z(q_j), Z(q_i), 0)(q_i, q_j, w_{ij})^{-\mathrm{T}}\upsilon \tag{4.55}$$

where $w_{ij} = (q_i \times q_j)/\|q_i \times q_j\|$. Equation (4.55) is then minimized by using the variable projection method [Ruhe 80] from a given set of correspondences with the aim of obtaining υ forcing $\|\upsilon\| = 1$. This minimization leads to estimates of three parameters of the linear velocity υ and the n depth parameters of every point. Hence, computing time depends considerably on the number of point correspondences.

Kanatani Estimator: The last of the surveyed methods considered in this chapter is that proposed by Kanatani [Kanatani 93b, Kanatani 93a] in 1993. One of the first methods to estimate camera movement based on the epipolar geometry was described by Zhuang, Thomas, Ahuja and Haralick [Zhuang 88] in 1998. Later, in 1993, Kanatani reformulated this method. He described the image sphere representation and solved the differential epipolar equation written in terms of the essential parameters and twisted optical flow.

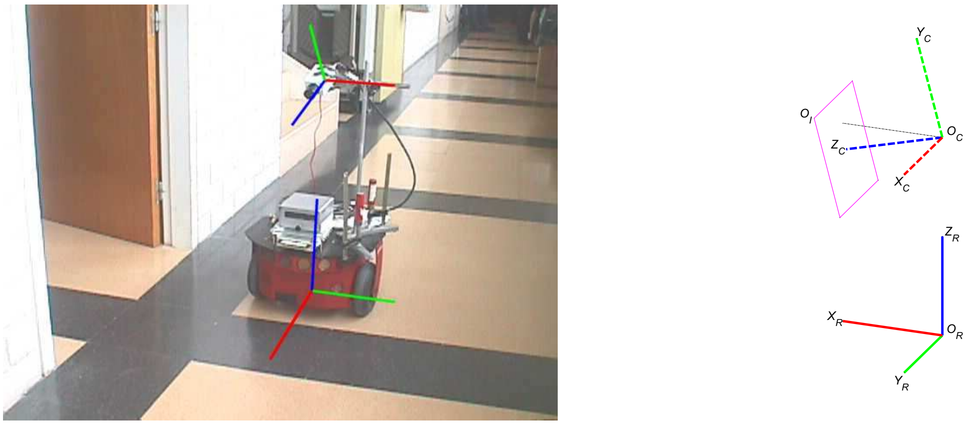
The first approach proposed by Kanatani was statistically biased. In order to remove the bias, he proposed an algorithm called renormalization, also in 1993. This second method automatically adjusts the bias and removes the image noise.

4.5 Adaptation to a Mobile Robot

The aim of this work is to estimate the motion of a mobile robot. Due to the fact that the permitted movements of a robot are limited, it is possible to establish some modifications in the differential epipolar equation by applying new constraints.

Our robot (see Figure 4.3) is constrained to only two independent movements: a translation along X_R axis and a rotation around Z_R axis.

$$^{R}\upsilon_r = (\upsilon_{r_1}, 0, 0)^{\mathrm{T}} \quad , \quad ^{R}\omega_r = (0, 0, \omega_{r_3})^{\mathrm{T}} \tag{4.56}$$



(a) (b)

Figure 4.3: (a) Robot for test the algorithms; (b) Robot and camera coordinate systems.

This limitation implies that the camera placed on and above the robot cannot move freely, so the camera velocity is the same as the robot velocity ($\upsilon_r = \upsilon_c$ and $\omega_r = \omega_c$). The motion with respect to the camera coordinate system depends on its position. The camera is placed in the vertical of the robot at height h and the X_C axis is parallel to Y_R and perpendicular to X_R. There is a known angle α between X_R and Z_C axis (see Figure 4.3). With this configuration the matrices which relate camera and robot coordinate systems are,

$$\begin{array}{rclrcl} {}^R\mathbf{R}_C & = & \mathbf{R_Z}(-\frac{\pi}{2})\,\mathbf{R_X}(-\frac{\pi}{2}-\alpha), & {}^R t_C & = & (0,0,h)^\mathrm{T} \\ {}^C\mathbf{R}_R & = & \mathbf{R_X}(\frac{\pi}{2}+\alpha)\,\mathbf{R_Z}(\frac{\pi}{2}), & {}^C t_R & = & (0,-h\cos\alpha, h\sin\alpha)^\mathrm{T} \quad . \end{array} \tag{4.57}$$

By transforming the velocities to camera coordinate system [Hutchinson 96],

$$ {}^C\upsilon_r = {}^C\mathbf{R}_R\,{}^R\upsilon_r - {}^C\mathbf{R}_R\,{}^R\omega_r \times {}^C t_R \tag{4.58}$$

$$ {}^C\omega_r = {}^C\mathbf{R}_R\,{}^R\omega_r \tag{4.59}$$

we obtain

$$ {}^C\upsilon_r = (0,\, \upsilon_{r_1}\sin\alpha,\, \upsilon_{r_1}\cos\alpha)^\mathrm{T} \tag{4.60}$$

$$ {}^C\omega_r = (0,\, \omega_{r_3}\cos\alpha,\, -\omega_{r_3}\sin\alpha)^\mathrm{T} \quad . \tag{4.61}$$

The camera motion is independent of h and depends only on three unknowns: υ_{r_1},

w_{r_3} and the angle α. With this information it is possible to simplify the symmetric matrix $\mathbf{S}_{\boldsymbol{r}}$ as

$$\mathbf{S}_{\boldsymbol{r}} = \frac{1}{2}(\hat{\omega}_r \hat{\upsilon}_r + \hat{\upsilon}_r \hat{\omega}_r) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \omega_{r3}\upsilon_{r_1} \sin\alpha\cos\alpha & \frac{1}{2}\omega_{r3}\upsilon_{r_1}(\cos^2\alpha - \sin^2\alpha) \\ 0 & \frac{1}{2}\omega_{r3}\upsilon_{r_1}(\cos^2\alpha - \sin^2\alpha) & -\omega_{r3}\upsilon_{r_1}\sin\alpha\cos\alpha \end{pmatrix} \tag{4.62}$$

where $s_{11} = s_{12} = s_{13} = 0$, $s_{22} = \frac{1}{2}\omega_{r3}\upsilon_{r_1}\sin(2\alpha)$, $s_{23} = \frac{1}{2}\omega_{r3}\upsilon_{r_1}\cos(2\alpha)$ and $s_{33} = -s_{22}$.

The following subsections concern the adaptation of the methods described in section 4.4.1 to the movement of a mobile robot by means of the differential epipolar constraint.

Least Squares Simplified Estimator using Eigen Analysis: This method is a simplified version of the method described in section 4.4.1. Starting from the differential epipolar equation,

$$\min_{\upsilon_{r_1}, s_{22}, s_{23}} \sum_{i=1}^{n} \left({q_i}^{\mathrm{T}}\, \hat{\upsilon}_r \dot{q}_i + {q_i}^{\mathrm{T}}\, \mathbf{S}_{\boldsymbol{r}} q_i \right)^2 \quad , \tag{4.63}$$

and knowing the camera motion constraint (equations (4.60), (4.61) and (4.62)), it is possible to rewrite the equation (4.63) in matrix form as

$$\min_{\theta'} \|\mathbf{U}'\theta'\|^2 \tag{4.64}$$

where,

$$\theta' = \left(\upsilon_{r_1}, s_{22}, s_{23} \right)^{\mathrm{T}} \quad , \tag{4.65}$$

$$\mathbf{U}' = \left(u'_1, u'_2, \ldots, u'_n \right)^{\mathrm{T}} \tag{4.66}$$

and

$$u'_i = \left((\dot{q}_{i_1} q_{i_2} - q_{i_1}\dot{q}_{i_2})\cos\alpha + (q_{i_1}\dot{q}_{i_3} - \dot{q}_{i_1}q_{i_3})\sin\alpha \ , \ {q_{i_2}}^2 - {q_{i_3}}^2 \ , \ 2q_{i_2}q_{i_3} \right)^{\mathrm{T}} \quad . \tag{4.67}$$

When there are n points and $n \geq 8$, the solution of θ' is the eigenvector corresponding to the smallest eigenvalue of the square matrix $\mathbf{X}' = \mathbf{U}'^{\mathrm{T}}\, \mathbf{U}'$.

Iteratively Reweighted Least Squares Simplified Estimator: This simplified estimation reduces the number of parameters and the weight function. By rewriting equation (4.25) in matrix form and only using three parameters (υ_{r_1}, s_{22} and s_{23}) we obtain

$$\min_{\|\theta'\|=1} \theta'^{\mathrm{T}}\, w'^{\mathrm{T}}\ \mathbf{U}'^{\mathrm{T}}\,\mathbf{U}'\,\theta' \tag{4.68}$$

$$\min_{\|\theta'\|=1} w'^{\mathrm{T}}\ \mathbf{U}'^{\mathrm{T}}\,\mathbf{U}'\,\theta'\ -\lambda\theta' \tag{4.69}$$

where θ' is defined in equation (4.65), $\mathbf{U}'$ is defined in equation (4.66), $w' = (w'_1, \ldots, w'_n)^{\mathrm{T}}$ and
$w'_i = (\|2\,\mathbf{S}_{\boldsymbol{r}}\,q_i + \hat{\upsilon}_r\dot{q}_i\|^2 + \|\hat{\upsilon}_r q_i\|^2)^{-1}$. It is necessary to use an iterative algorithm (see Figure 4.2) to solve equation (4.68), which is actually the same, already applied in the general method, but uses the simplified equation $\mathbf{X}' = w'^{\mathrm{T}}\,\mathbf{U}'^{\mathrm{T}}\,\mathbf{U}'$.

Modified Iteratively Reweighted Least Squares Simplified Estimator: This method is the adapted version to a mobile robot of the method used to estimate camera motion proposed by Brooks [Brooks 98] and described in section 4.4.1. When considering equation (4.25), the reduced number of parameters to estimate leads to a simplified equation as follows.

$$w_i'^{-1} = \|2\,\mathbf{S}_{\boldsymbol{r}}\,q_i + \hat{\upsilon}_r\dot{q}_i\|^2 + \|\hat{\upsilon}_r q_i\|^2 = \theta'^{\mathrm{T}}\,\mathbf{N}'_i\,\theta' \tag{4.70}$$

where,

$$\mathbf{N}'_i = 4\sum_{\alpha=1}^{3}\boldsymbol{\rho'}_{\alpha}^{\mathrm{T}}\, q_i q_i^{\mathrm{T}}\,\boldsymbol{\rho'}_{\alpha} + 4\sum_{\alpha=1}^{3}\boldsymbol{\rho'}_{\alpha}^{\mathrm{T}}\, q_i\dot{q}_i^{\mathrm{T}}\,\boldsymbol{\sigma'}_{\alpha} - \sum_{\alpha=1}^{3}\boldsymbol{\sigma'}_{\alpha}^{\mathrm{T}}\,\dot{q}_i\dot{q}_i^{\mathrm{T}}\,\boldsymbol{\sigma'}_{\alpha} - \sum_{\alpha=1}^{3}\boldsymbol{\sigma'}_{\alpha}^{\mathrm{T}}\, q_i q_i^{\mathrm{T}}\,\boldsymbol{\sigma'}_{\alpha} \tag{4.71}$$

and

$$\begin{aligned}
\boldsymbol{\rho}'_1 &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, & \boldsymbol{\sigma}'_1 &= \begin{pmatrix} 0 & 0 & 0 \\ -\cos(\alpha) & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \\
\boldsymbol{\rho}'_2 &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, & \boldsymbol{\sigma}'_2 &= \begin{pmatrix} \cos(\alpha) & 0 & 0 \\ 0 & 0 & 0 \\ -\sin(\alpha) & 0 & 0 \end{pmatrix}, \\
\boldsymbol{\rho}'_3 &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix}, & \boldsymbol{\sigma}'_3 &= \begin{pmatrix} 0 & 0 & 0 \\ \sin(\alpha) & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.
\end{aligned} \tag{4.72}$$

Then equation (4.25) is rewritten in matrix form considering the parameters to estimate the robot movement obtaining,

$$\min_{\|\theta'\|=1} 2\,\mathbf{X}'\,\theta' \tag{4.73}$$

where,

$$\mathbf{X}' = \sum_{i=1}^{n} \frac{\mathbf{M}'_i}{\theta'^{\mathrm{T}}\ \mathbf{N}'_i\,\theta'} - \sum_{i=1}^{n} \frac{\theta'^{\mathrm{T}}\ \mathbf{M}'_i\,\theta'}{(\theta'^{\mathrm{T}}\ \mathbf{N}'_i\,\theta')^2}\mathbf{N}'_i \tag{4.74}$$

$$\mathbf{M}'_i = u'_i u'^{\mathrm{T}}_i \tag{4.75}$$

using u'_i which was previously described in equation (4.67).

Finally, θ' is obtained, minimizing $\mathbf{X}'\theta' = 0$ by using eigen analysis and the algorithm shown in Figure 4.3.

Least Median Squares Simplified Estimator: The adaptation of LMedS to robot movement estimation forces the following considerations. First, instead of using the 7-points method to measure of the velocity for each group of points, we will use the simplified least squares method described in this section. In the adaptation to a mobile robot, the movement is constrained to only three unknowns; that is straight forward, translation and rotation. Moreover, the number of sets to generate randomly also changes as it depends on the number of set points. The equation (4.34) relates such a relationship.

The second modification of this method takes place in the last step of the algorithm once all the outliers have been removed, and υ_r and $\mathbf{S}_{\boldsymbol{r}}$ are recalculated using the modified iteratively reweighted least square simplified estimator instead of the general method.

Ma, Košecká and Sastry Simplified Estimator: The method proposed by Ma et al. [Ma 00, Ma 98a] is not affected by the reduction of the number of parameters. Moreover, only the first step of the method has to be modified to adapt the algorithm to the case of a robot moving on a plane, that is υ_0 and $\mathbf{S}$. The equation to minimize is the following,

$$\min_{\theta'} \|\mathbf{U}'\theta'\|^2 \quad . \tag{4.76}$$

Equation (4.76) computes υ_{r_0} and $\mathbf{S}_{\boldsymbol{r}}$, these results are used in the further steps of the algorithm already detailed in section 4.4.1. Then the steps of the algorithm are the following:

1. Estimate υ_{r_0} and $\mathbf{S}_{\boldsymbol{r}}$.
2. Recover the special symmetric matrix.
3. Recover velocities form the special symmetric matrix.
4. Recover velocity.

Baumela, Agapito, Bustos and Reid Simplified Estimator: The adaptation of the general method proposed by Baumela et al. [Baumela 00] described in section 4.4.1 to the simplified case of the robot movement is based on the following modifications. Considering equation (4.9) and using the simplified camera movement described in equations (4.60) and (4.62), the following linear equation is obtained,

$$a'\dot{q}_1 + b'\dot{q}_2 + c' = 0 \tag{4.77}$$

where $a' = \upsilon_{r_1}(q_2 \cos\alpha - \sin\alpha)$, $b' = -\upsilon_{r_1} q_1 \cos\alpha$ and $c' = s_{22}({q_2}^2 - 1) + 2 s_{23} q_2$.

This algorithm minimizes the distance between equation (4.77) and the optical flow of every point by using the following equation.

$$\min_{\upsilon_{r_1}, s_{22}, s_{23}} \sum_{i=1}^{n} \frac{(a'_i \dot{q}_{i_1} + b'_i \dot{q}_{i_2} + c'_i)^2}{{a'_i}^2 + {b'_i}^2} \tag{4.78}$$

Considering that the measures of the optical flow present a given discrepancy, equation (4.78) is rewritten in matrix form including the covariance matrix of the optical flow $\mathbf{\Sigma}_{\dot{q}_i}$, obtaining

$$\min_{\theta'} \sum_{i=1}^{n} \frac{\theta'^{\mathrm{T}}\, f'_i\, {f'_i}^{\mathrm{T}}\, \theta'}{\upsilon_r^{\mathrm{T}}\, \mathbf{H}_i \mathbf{\Sigma}_{\dot{q}_i} {\mathbf{H}_i}^{\mathrm{T}}\, \upsilon_r} \tag{4.79}$$

where $f'_i = \Big((\dot{q}_{i_1} q_{i_2} - q_{i_1} \dot{q}_{i_2}) \cos\alpha - \dot{q}_{i_1} \sin\alpha\ ,\ {q_{i_2}}^2 - 1\ ,\ 2 q_{i_2} \Big)^{\mathrm{T}}$ and ${\mathbf{H}_i}^{\mathrm{T}}$ is defined in equation (4.46).

However, the constraint ${\upsilon_r}^{\mathrm{T}} \mathbf{S}_{\boldsymbol{r}} \upsilon_r = 0$ has to be forced to equation (4.79) in minimization. Hence, the parameter s_{22} is extracted and substituted in equation (4.79), obtaining

$$\min_{\delta'} \sum_{i=1}^{n} \frac{\delta'^{\mathrm{T}}\, g'_i\, g'^{\mathrm{T}}_i\, \delta'}{\upsilon_r{}^{\mathrm{T}}\, \mathbf{H}_i \boldsymbol{\Sigma}_{\dot{q}_i} \mathbf{H}_i{}^{\mathrm{T}}\, \upsilon_r} \tag{4.80}$$

where,

$$g'_i = \Big((\dot{q}_{i_1} q_{i_2} - q_{i_1} \dot{q}_{i_2}) \cos\alpha - \dot{q}_{i_1} \sin\alpha \ , \ 2q_{i_2} - \tan(2\alpha)({q_{i_2}}^2 - 1) \Big)^{\mathrm{T}} \quad , \tag{4.81}$$

$$\delta' = (\upsilon_{r_1}, s_{23})^{\mathrm{T}} \tag{4.82}$$

and

$$s_{22} = -s_{23} \tan(2\alpha) \ \ . \tag{4.83}$$

4.6 Summary

Traditionally, the estimation of the movement of a single camera is based on the application of the epipolar geometry between every two consecutive images. Epipolar geometry is contained in the fundamental matrix, which can be computed by linear or non-linear optimization from a set of correspondences. However, traditional epipolar geometry has some limitations in the common case of a single camera attached to a mobile robot. First of all, the disparities between two consecutive images are rather small at usual image rates, leading to numerical inaccuracies in the computation of the fundamental matrix. Indeed, disparities could be increased by removing intermediate images, but then the obtained movement is a rough approximation of reality. Secondly, the correspondence problem between images consumes a lot of computing time reducing its effectiveness for real time applications.

Some authors have proposed techniques to estimate the camera motion at a high image rate based on the computation of the linear and angular velocities instead of on the orientation and translation between consecutive images. These techniques are commonly based on the optical flow and the differential epipolar constraint. This chapter has presented an up-to-date classification of the methods and techniques used to estimate the movement of a single camera. A survey of several motion recovery methods has been presented. The techniques surveyed were classified and the mathematical notation has been unified in this document. Several methods were described after analyzing their differences with respect to the use of epipolar geometry in both the discrete and differential cases. This chapter was based on giving a better understanding of the relative performances of the 6-DOF camera movement estimators.

The general methods to estimate a 6-DOF movement have been adapted to the common case of a mobile robot moving on a plane. The adaptation was based on constraining the movement from six degrees of freedom to only two degrees, which were: a) the spin rotation of the mobile robot, and b) the forward motion. With these modifications, the number of potential solutions is reduced so the obtained results improve considerably.

Chapter 5

A Mobile Robot Application: Localization and Mapping

Several mobile robotic applications require a localization system with the aim of facilitating the navigation of the vehicle and even the execution of the planned trajectories. The localization is always relative to the map of the environment in which the robot is moving. This chapter presents a computer vision system which deals with the localization of the vehicle while simultaneously mapping the environment. A stereoscopic vision system is proposed in which the theory surveyed in the previous chapters is taken into account during the whole process.

5.1 Introduction

Navigation and map building from an unknown environment are important tasks for future generations of mobile robots. The are several different fields in which it is possible to apply mobile robotic systems, such as applications in industrial environments [Lauterbach 93], the transport of dangerous substances, surveillance tasks and many others. In all these fields, it becomes necessary to equip the mobile robots with a sensor system which allows the vehicle to obtain information from the environment. The processing of these data, provided by the sensor system must be useful to facilitate actions and control planning. A stereo vision system has proved to be a reliable and effective way to extract range information from the environment.

Accurate localization is a prerequisite for building a good map, and having an accurate map is essential for good robot localization. Therefore, simultaneous localization and map building (SLAM) is a critical factor for successful mobile robot navigation. Various authors have been working with in SLAM systems using different types of sensors, such as sonar [Borenstein 96], laser range finders [Weckesser 96] and vision [DeSouza 02]. Sonar is fast but inaccurate, whereas a laser scanning system is active, accurate, but rather slow. Vision systems are passive and provide high resolution. Some systems using a single camera [Se 02] or even a stereo vision composed of two cameras have been proposed [Iocchi 98].

This chapter describes a vision system designed to build a map from the surroundings of the mobile robot and explains how this information is used to locate the vehicle. This proposal is based on a stereo vision system using the techniques already presented in the previous chapters, such as camera calibration and epipolar geometry. Moreover, low level image processing techniques basically used to solve the correspondence problem are briefly introduced. The main objectives of the proposed vision system are the following:

- Building a 3D map from an unknown environment using a stereo camera system.
- Localization of the robot on the map.
- Providing a new useful sensor for robot control architectures to facilitate robot navigation.

This chapter is organized into the followings sections. First, section 5.2 describes the mobile robot used in the set-up. Section 5.3 describes the different steps needed to achieve an algorithm dealing with the localization and mapping by means of stereo vision.

5.2 The Mobile Robot

The platform utilized for the real experiments is a mobile robot called *GRILL*. This robot is based on the commercial robot Pioneer 2 DX from ActivMedia©. This robot was originally developed by Kurt Konolige of SRI International, Inc. and Stanford University [Act 00].

The mobile robot contains the basic components for sensing and actuators including battery power, drive motors, encoders, and a front sonar ring. These components are controlled by a 20 MHz Siemens 88C166-based microcontroller which also runs the Pioneer 2 Operating System (P2OS). This microcontroller performs low-level tasks such as odometry, sonar control, and translational and rotational speed control. The GRILL robot is also equipped with an onboard computer which communicates with the microcontroller by means of a RS-232 serial port. This computer is a Pentium© 233 MHz with Ethernet and a PCI bus and space for PC104+ accessory cards. The GRILL robot uses this onboard computer just for control tasks.

GRILL is equipped with a second computer (Pentium© III 850 MHz) with Ethernet and a PCI bus for two PC104+ frame grabber cards. This second computer is used to run the stereo vision system. The frame grabbers used are PXC200 Imagenation with four channels each one. However, two frame grabbers are required because it is not possible to grab two images simultaneously with the same frame grabber. The cameras utilized are two JAI CV-S3300 connected to both frame grabbers. Both cameras are synchronized in order to simultaneously acquire the two images which form the stereo vision system. Figure 5.1 summarizes schematically the different elements which compose the robot.

The communication between both onboard computers is realized with an Ethernet LAN. The vision computer is also connected by means of a wireless Ethernet with the host computer. The host computer is utilized to monitor and remotely operate the mobile robot. The remote operation of the robot is carried out by using a console application and the image provided by one of the two cameras. This image is acquired by the camera and transmitted by means of a radio link to a video screen closer to the host computer (see Figure 5.2).

Figure 5.3a and Figure 5.3b show two different views of the vision system mounted on the robot. This system is composed of the main computer, two PXC200 frame grabbers, a dedicated laptop hard disk and the power supply. Moreover, there is a fan cooler to refrigerate the whole system. The synchronized stereo camera, together with the Ethernet and video link, is mounted on the robot as shown in Figure 5.3c and Figure 5.3d.

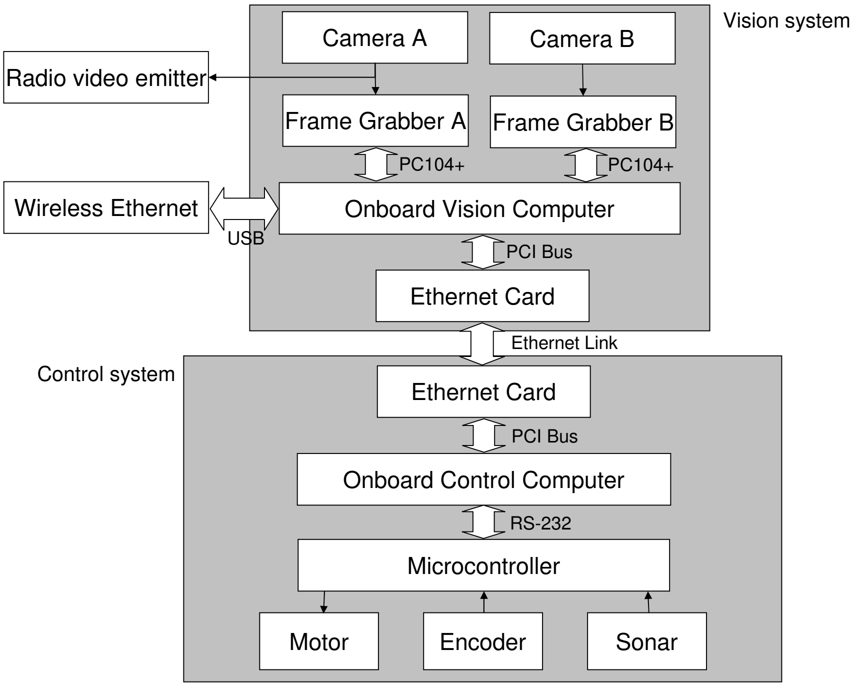


Figure 5.1: Block diagram of the mobile robot GRILL.

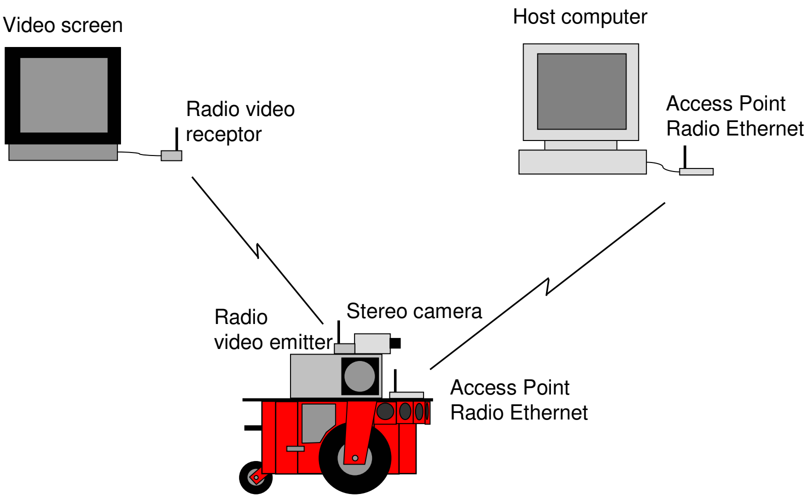
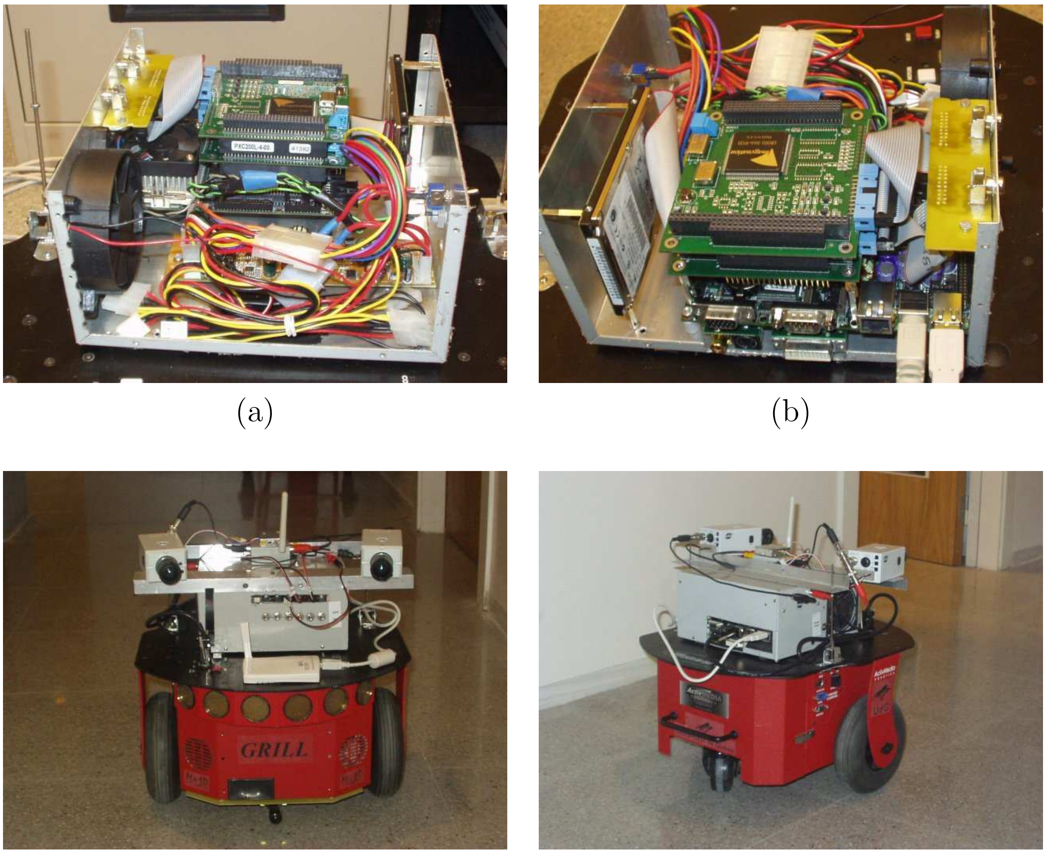


Figure 5.2: Elements which interact with the mobile robot GRILL.



(a) (b)

(c) (d)

Figure 5.3: Mobile Robot called GRILL. (a) Onboard vision computer, frontal view. (b) Onboard vision computer, back view. (c) GRILL frontal view. (d) GRILL back view.

5.3 Localization and Map-Building Algorithm

The proposed localization and map-building algorithm is divided into the following three on-line steps:

1. *2D image processing.* The 2D image processing step deals with finding interest points in each pair of images and further solving the problem of matching two images. Moreover, the correspondence problem is also solved temporally through the sequence of images captured by each camera.

2. *3D image processing.* Given the image point correspondences and the camera calibrations (containing the geometry of the stereo rig), the 3D image pro-

cessing step deals with the computation of the 3D points with respect to the position of the mobile robot by means of triangulation. Then, the 2D and 3D points are tracked through time.

3. *Localization and map-building.* The robot is located on the map using the points connected through time. The map is updated with new points provided by the second step by using the localization of the robot estimated in the last iteration.

These steps are summarized in Figure 5.4. This algorithm requires an off-line process consisting of the calibration of both cameras. Camera calibration is utilized in the following tasks: a) removal of the image distortion caused by the lens; b) Exploitation of epipolar geometry to simplify the correspondence problem, i.e. the epipolar geometry increases the accuracy in solving the correspondence problem; c) triangulation making use of the calibrating parameters of both cameras; and d) localization of the robot on the map by again using the camera parameters.

The following section describes the off-line process with details of the three entire on-line steps introduced in the previous paragraphs.

5.3.1 Off-line Process

The off-line process consists of an explicit calibration because the intrinsic and extrinsic parameters are required. Therefore, Hall's method is not applicable here. Moreover, the fundamental matrix can be directly computed from the calibrating parameters, i.e. consider ${}^{C}\mathbf{K}_{R}$ a 4 × 4 matrix relating the robot coordinate system $\{R\}$ with respect to the first camera coordinate system $\{C\}$, and consider ${}^{C'}\mathbf{K}_{R}$ a second 4 × 4 matrix which relates the robot coordinate system $\{R\}$ with respect to the second camera coordinate system $\{C'\}$. Then, the relative position and orientation of one camera with respect to the other is described by equation (5.1).

$${}^{C}\mathbf{K}_{C'} = {}^{C}\mathbf{K}_{R}\,{}^{C'}\mathbf{K}_{R}^{-1} \tag{5.1}$$

Once ${}^{C}\mathbf{K}_{C'}$ is computed, ${}^{C}\mathbf{R}_{C'}$ and ${}^{C}T_{C'}$ can be extracted and, by using the definition of fundamental matrix, $\mathbf{F}$ is obtained (see equation (3.9)). These relationships between the different coordinate systems are graphically shown in Figure 5.5.

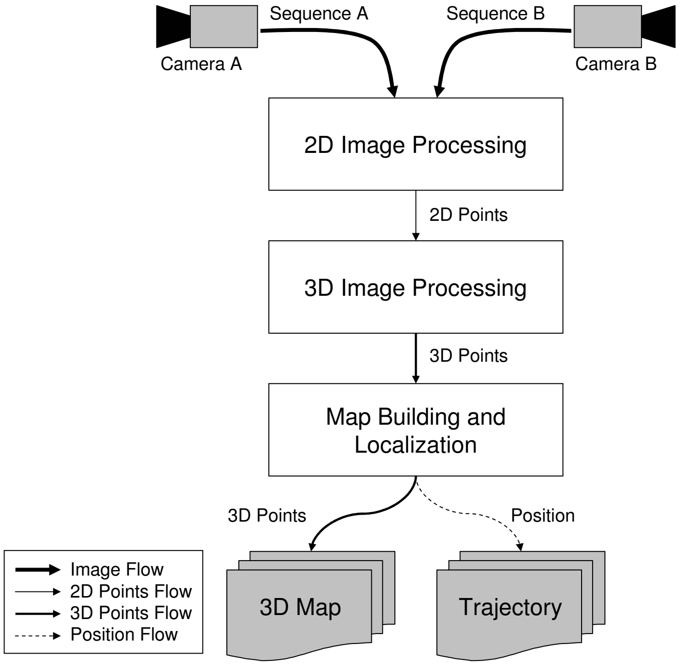


Figure 5.4: Data flow diagram of the three on-line step algorithm.

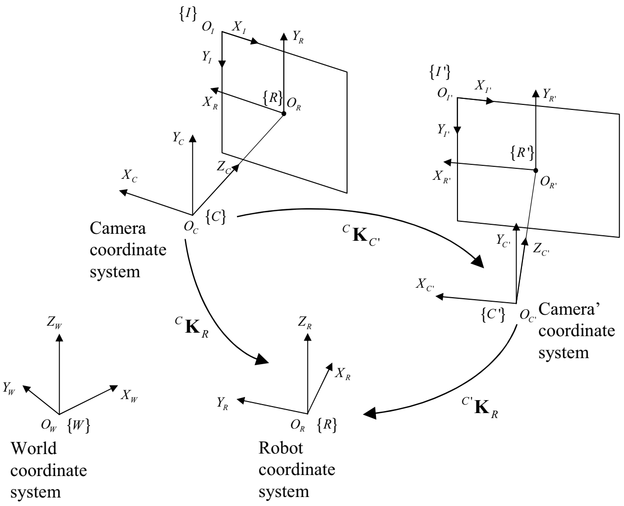


Figure 5.5: Cameras, robot and world coordinate systems.

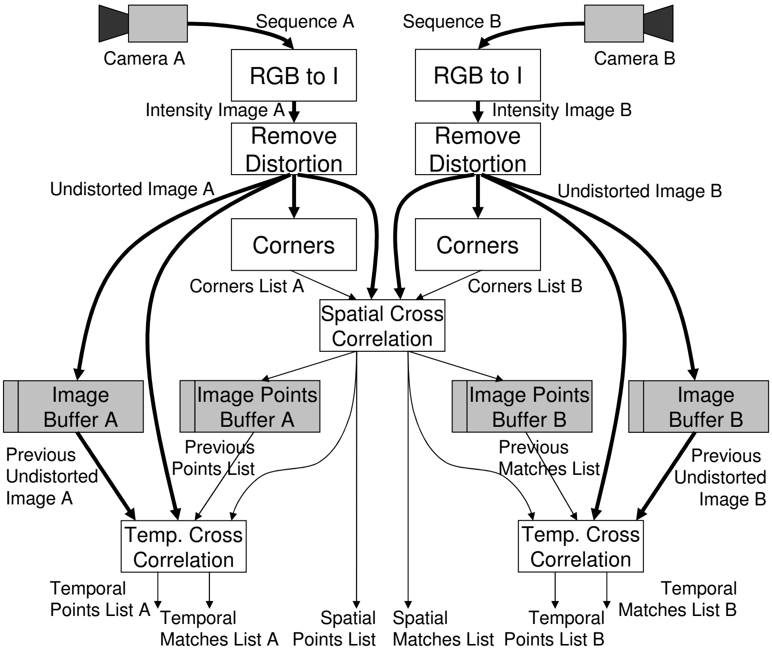


Figure 5.6: Data flow diagram of 2D image processing.

5.3.2 2D Image Processing

2D image processing is divided into five stages: a) image conversion; b) image distortion removal; c) corner detection; d) spatial cross correlation; and e) temporal cross correlation; as shown in Figure 5.6.

The first stage is a trivial conversion from the color image provided by both cameras to two intensity images due to the fact that the proposed algorithm does not consider the use of color information.

The second stage removes lens distortion by means of the camera calibration parameters. This stage considerably improves the accuracy obtained by further stages, such as spatial cross correlation (epipolar lines will be really straight lines) and triangulation (optical rays are again straight lines leading to a simpler and more precise intersection). The image effect given by lens distortion is shown in Figure 2.3 and Figure 2.5.

The third stage consists of finding the points of interest. A variant of the Harris *corner detector* [Harris 88] has been utilized, which was proposed by Garcia and detailed in [García 01b]. In general, most of the image points do not provide any significant information to build a map. So, the use of interest points considerably reduce the computation time of image processing. Corners have been widely used by the scientific community as the simplest way of finding points of interest in any image [McCane 02]. Other possibilities might use scale invariants [Lowe 99] or projective invariants [Vanderkooy 96]. Overall, a corner is defined as a point where a large intensity variation in every direction is presented. Corners are found by computing an unnormalised local autocorrelation of the image intensity at that point in four directions. A 2 ×2 autocorrelation matrix is computed for every pixel,

$$\mathbf{A} = w \begin{pmatrix} I_x^2 & I_{xy} \\ I_{xy} & I_y^2 \end{pmatrix} \tag{5.2}$$

where

$$I_x = \frac{\partial I}{\partial x}, \qquad I_y = \frac{\partial I}{\partial y}, \qquad I_{xy} = \frac{\partial^2 I}{\partial xy}. \tag{5.3}$$

$\mathbf{x} = (x, y)$, $I(\mathbf{x})$ is the intensity image, w is a gaussian smoothing mask. A pixel is flagged as a corner if both eigenvalues are larger than a given threshold. In order to avoid eigen value descomposition of $\mathbf{A}$, the coefficient of cornerness is defined as

$$c(\mathbf{x}) = \det(\mathbf{A}) - k(\mathrm{Tr}(\mathbf{A}))^2 \tag{5.4}$$

where k is a given constant, typically 0.04, and $\mathrm{Tr}(\mathbf{A})$ is the trace of the matrix $\mathbf{A}$. Finally, local minimums in image cornerness are candidates to become a corner.

The fourth stage deals with the computation of spatial cross correlation between stereo images by using the epipolar geometry. We have adapted the technique already utilized in the underwater robots in our lab to make a mosaic of the seabed [García 01a]. The two main differences are:

1. The original algorithm correlates a list of interest points provided by a first image with all the points laying with a search window in the second image. In our case, the list of interest points provided by the first camera is correlated with a second list of interest points provided by the second camera.

2. The geometric position between both cameras is known through camera calibration. The search windows utilized to search the correspondences has been substituted to a search of those interest points along the epipolar line.

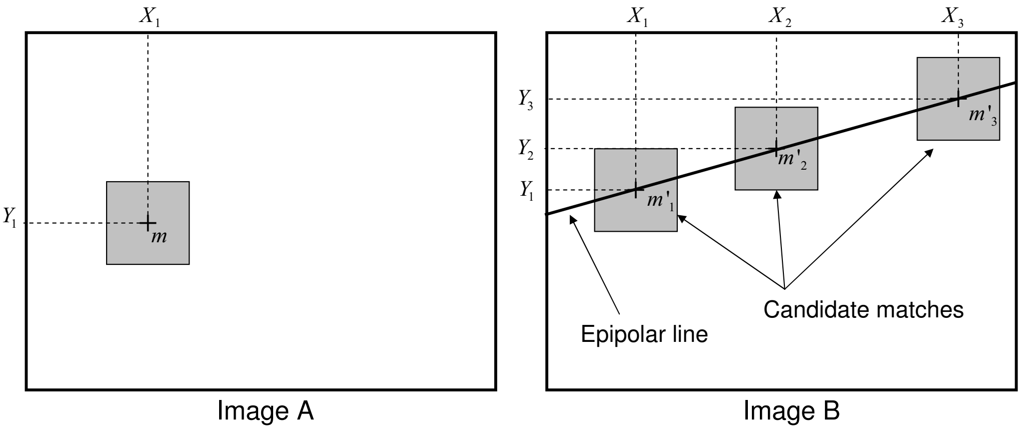


Figure 5.7: Example of a spatial cross correlation where the image point m has three candidates in the second image.

Both modifications of the algorithm considerably reduce the computation time, improving the accuracy of cross correlation and reducing drastically the number of outliers.

The cross correlation coefficient is described as follows: Consider an interest point m in the first image in which the correlation window is centered. Then, the epipolar line given by m in the second image is computed, as can be seen in Figure 5.7. Finally, the correlation operation is performed, computing a normalized correlation score [Zhang 94] for all the candidates near the epipolar line in the following way:

$$\mathrm{CS}(m,m') = \frac{\sum_{i=-\alpha}^{\alpha}\sum_{j=-\alpha}^{\alpha}\Big(I(x_1+i\cdot q, y_1+j\cdot q)-\overline{I(x_1,y_1)}\Big)\cdot\Big(I'(x_2+i\cdot q, y_2+j\cdot q)-\overline{I'(x_2,y_2)}\Big)}{\alpha^2\sqrt{\sigma^2(I)\sigma^2(I')}} \tag{5.5}$$

where $\alpha = \frac{n-1}{2q}$, n is the size correlation window, q is a subsampling coefficient, $\sigma^2(I)$ is the variance of the image computed in the correlation window (see equation (5.6)), and $\overline{I(x,y)}$ is the average of the correlation window as shown in equation (5.7).

$$\sigma^2(I) = \frac{\sum_{i=-\alpha}^{\alpha}\sum_{j=-\alpha}^{\alpha} I(x+i\cdot q, y+j\cdot q)^2}{\alpha^2} - \overline{I(x,y)}^2 \tag{5.6}$$

$$\overline{I(x,y)} = \frac{\sum_{i=-\alpha}^{\alpha}\sum_{j=-\alpha}^{\alpha} I(x+i\cdot q, y+j\cdot q)}{\alpha^2} \tag{5.7}$$

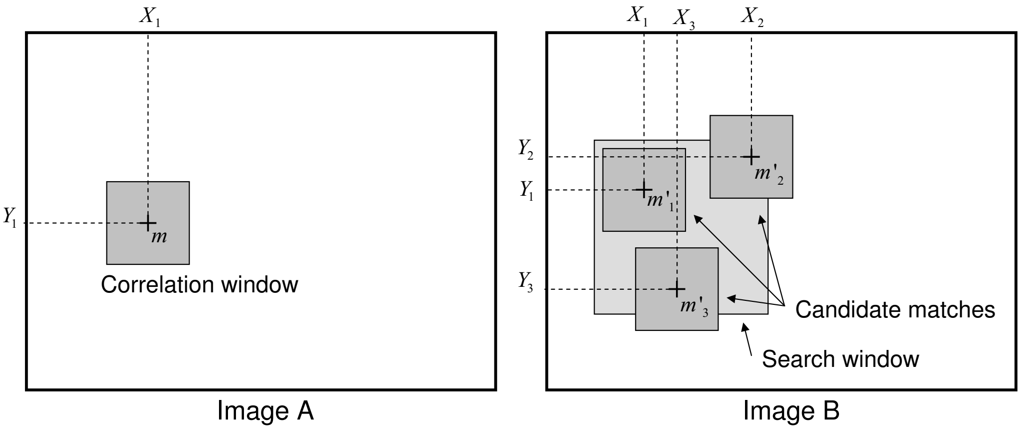


Figure 5.8: Example of temporal cross correlation where the image point m has three candidates in the second image.

The correlation score of equation (5.5) gives a value in the interval [-1,+1] where +1 means that the point m and m' are identical, and -1 indicates no similarity at all.

Finally, the last stage of 2D image processing is based on temporal cross correlation. Until this moment, the algorithm has been established as a spatial relationship between image sequences provided by both cameras. Temporal cross correlation consists of establishing the temporal relationship between the images and is computed starting from the corners spatially correlated between both cameras. The cross correlation algorithm utilized is the same as was used in spatial cross correlation with just one difference: the epipolar geometry between two consecutive images is unknown due to its dependence on the robot's motion. Search windows are used to determine the correspondences instead of the epipolar lines used in the spatial cross correlation (see Figure 5.8).

5.3.3 3D Image Processing

Once the 2D Image processing step is accomplished for every two images, a list of 2D spatially and temporally correlated image points are obtained. Then, the 3D image processing step is divided into three tasks as shown in Figure 5.9.

First, the third dimension is computed from every pair of 2D spatially correlated image points by triangulation. So, the *stereo reconstruction* task is based on the

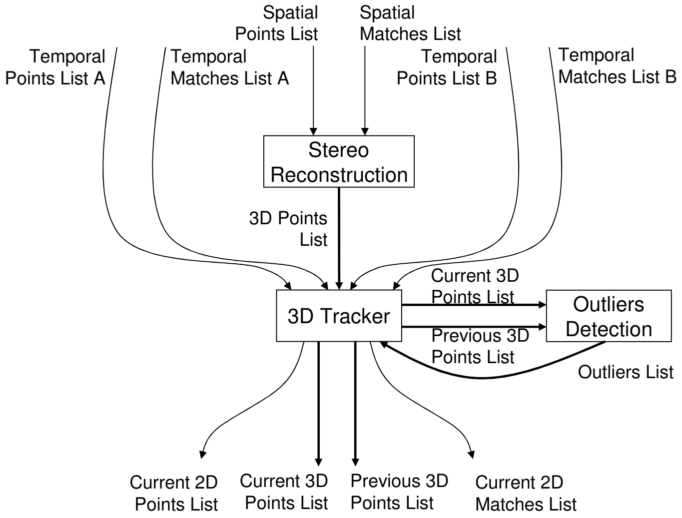


Figure 5.9: Data flow diagram of 3D processing subsystem.

triangulation technique described in section 3.2 and illustrated in Figure 3.2.

Second, the *3D tracker* task consists of computing the displacement of these 3D object points through the whole image sequence. In order to accomplish this task, the list of image points correlated temporally and the list of reconstructed 3D object points are utilized. This task stores the image points together with the 3D object points and tracks the point in the image plane and in the 3D space (see Figure 5.10), removing any point whose tracking has been broken. Moreover, the algorithm also removes those points whose displacement in both image planes is inconsistent.

Finally, the last task consists of detecting and removing the bad correspondences produced by 2D image processing which are easily detectable at this moment. In some cases, bad image correspondences are persistent throughout the time. In general, these outliers are produced by spatial cross correlation and are not detected by the tracker. Then, an additional task is necessary to detect and remove any left over bad correspondences. This task is called *outlier detector* and consists of detecting inconsistent 3D displacements.

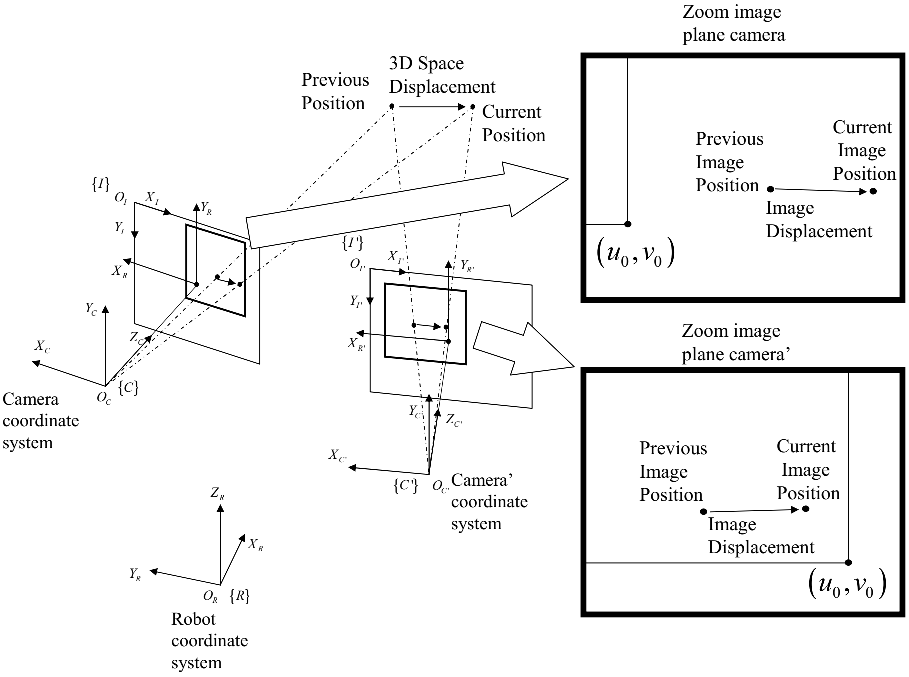


Figure 5.10: Example of tracker in 3D space and 2D image planes.

5.3.4 Localization and Map-Building

The last step described in Figure 5.4 consists of the robot localization and map-building. This step provides a global map of the environment and the trajectory carried out by the robot with respect to this map. There are three tasks to accomplish as shown in Figure 5.11.

First, the local displacement of the robot is estimated by using the current 3D object points and the map build up to this moment. The robot odometry gives only a rude estimation and is prone to errors such as drifting, slipping, etc. We propose to improve the odometry by using stereo vision system. The 3D motion of the stereo vision system is computed by using the matches stored in the tracker. The robot coordinate system is linked with the stereo vision system, consequently the robot displacement is directly obtained. The displacement is restricted to a 2D planar motion with just 3 degrees of freedom (two translations and one orientation). The idea is to minimize the distance between the projection in both image planes of a set of 3D object points (contained in the map) with the observed position of those image points. Figure 5.12 illustrates this idea showing a map formed by a

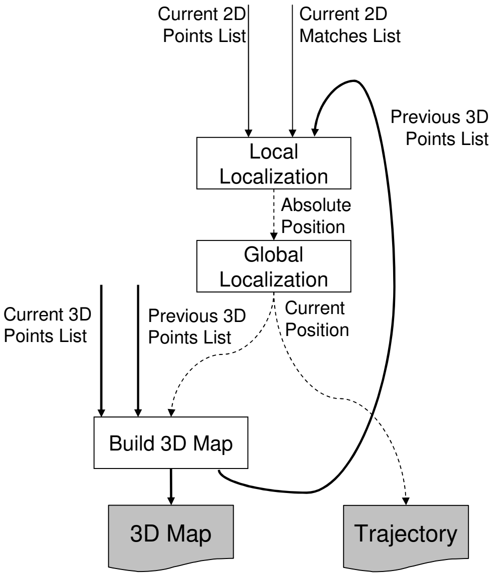


Figure 5.11: Data flow diagram of localization and map-building subsystem.

single cube and its projections on both image planes at the present instant, and the discrepancy with respect to the projection of the same cube at the previous instant. The 3D motion of the robot is computed by minimizing both image projections using the discrepancies between the 2D points. A Newton-Raphson solution is computed using the previous position estimated by the robot as the initial guess.

Second, the trajectory realized by the robot is updated with the new position by using the task called *Global Localization*.

Third, the map is updated including the points provided by the last stereo image, consisting of making a 3D map and integrating different scene reconstructions and the estimated localization. The map is continually updated especially when the current position of the robot changes and the set of object points changes due to the robot movement or new object detection. Existing object points are updated if their relative position with respect to the robot is closer than the relative position on the map, as is the case when the robot approaches the object point, thereby improving the accuracy of the map.

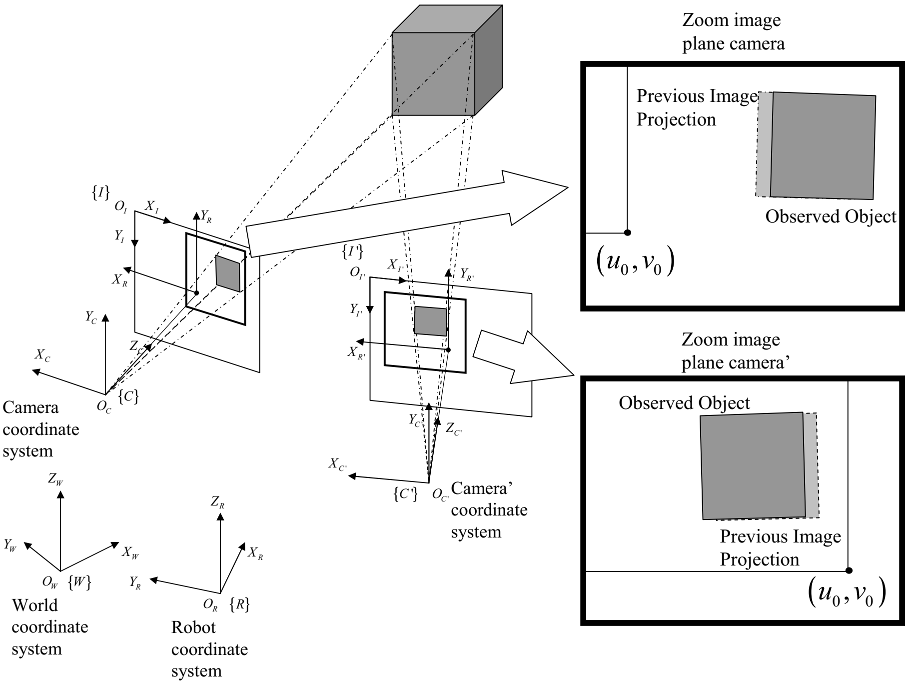


Figure 5.12: Local localization minimizing the discrepancy between the observed points at the present moment with respect to the previous moment.

Chapter 6

Experimental Results

This chapter presents the experimental results obtained in synthetic and real scenarios by means of the algorithms described in the previous chapters. Basically, this chapter presents the results on camera calibration, fundamental matrix estimation and motion recovery which are the subjects of the first chapters of this thesis. The results are compared and analyzed bringing insightful information concerning the potential and robustness of each method surveyed. Also, this chapter presents qualitative results obtained when applying the algorithm described in chapter 5, that is, the mapping and localization approach for a mobile robot.

6.1 Introduction

This chapter presents the experimental results obtained from the implementation of the different methods described in the previous chapters. Also, a brief discussion of these results is included with the aim of giving the reader details of the accuracy and robustness of each method. First, camera calibration results are given in section 6.2. The results obtained with synthetic images allow us to compare the accuracy of the camera calibration surveyed methods without considering gaussian noise leading to a pure analyzation of the accuracy of the method with a set of known points apart from segmentation discrepancies. Furthermore, real cameras have been considered when describing the calibration process, the segmentation approach and the experimental results obtained. Second, the experiments carried out on the estimation of the fundamental matrix are described in section 6.3. Synthetic data are generated aggregating gaussian noise and potential outliers to evaluate the accuracy and robustness of the surveyed estimators. Again, images from a wide variety of real scenes are used and the epipolar geometry is recovered. Third, the results obtained by means of motion recovery are presented in section 6.4 considering both a camera with 6 degrees of freedom in a free 3D movement and a camera with 2 degrees of freedom in the case of a mobile robot. Finally, the chapter ends with the presentation of the qualitative results obtained by using a stereo vision system mounted on a mobile robot. Then, section 6.5 shows the results obtained in each step of the whole mapping and localization approach.

6.2 Camera Calibration Experimental Results

The experimental results presented in this section were carried out with the following two purposes. On the one hand, the comparison of the camera calibration surveyed methods in terms of accuracy by using synthetic data with the aim of facilitating the choice of the most convenient method for any given application. On the other hand, the description of how a camera is calibrated step by step, including the results obtained in several real scenarios.

6.2.1 Camera Calibration with Synthetic Images

Instead of using our own experimental setup, we decided to download a list of corresponding points from the well-known Tsai's Camera Calibration Software Webpage (`http://www.cs.cmu.edu/~rgw/TsaiCode.html`). Actually, results are always conditioned to the structure of the 3D points and the image processing tools used in segmentation and further points extraction. Hence, this decision was just taken to allow the scientific community to reproduce the same conditions.

Then, the surveyed calibrating techniques have been implemented and their accuracy measured using the following criteria: a) Distance with respect to the optical ray; b) Normalized Stereo Calibration Error; c) Accuracy of distorted image coordinates; and d) Accuracy of undistorted image coordinates. The two first criteria calculate the accuracy with respect to a world coordinate system. The other two calculate the discrepancy on the image plane (see section 2.4 for details on accuracy measurement). First, table 6.1 shows the accuracy measured by using the first

Table 6.1: Accuracy of 3D coordinate measurement with synthetic data.

	3D position (mm.)			NSCE
	Mean	σ	Max	
Hall	0.1615	0.1028	0.5634	n/a
Faugeras	0.1811	0.1357	0.8707	0.6555
Faugeras NR[1] without distortion	0.1404	0.9412	0.0116	0.6784
Faugeras NR with distortion	0.0566	0.0307	0.1694	0.2042
Tsai	0.1236	0.0684	0.4029	0.4468
Tsai optimized	0.0565	0.0306	0.1578	0.2037
Tsai with principal point of Tsai optimized	0.0593	0.0313	0.1545	0.2137
Tsai optimized with principal point of Tsai optimized	0.0564	0.0305	0.1626	0.2033
Weng	0.0570	0.0305	0.1696	0.2064

Table 6.2: Accuracy of 2D coordinate measurement with synthetic data.

	2D distorted image (pix.)			2D undist. image (pix.)		
	Mean	σ	Max	Mean	σ	Max
Hall	0.2676	0.1979	1.2701	0.2676	0.1979	1.2701
Faugeras	0.2689	0.1997	1.2377	0.2689	0.1997	1.2377
Faugeras NR without distortion	0.2770	0.2046	1.3692	0.2770	0.2046	1.3692
Faugeras NR with distortion	0.0840	0.0458	0.2603	0.0834	0.0454	0.2561
Tsai	0.1836	0.1022	0.6082	0.1824	0.1011	0.6011
Tsai optimized	0.0838	0.0457	0.2426	0.0832	0.0453	0.2386
Tsai with principal point of Tsai optimized	0.0879	0.0466	0.2277	0.0872	0.0463	0.2268
Tsai optimized with principal point of Tsai optimized	0.0836	0.0457	0.2500	0.0830	0.0454	0.2459
Weng	0.0845	0.0455	0.2608	0.0843	0.0443	0.2584

criteria and the second criteria, respectively. Note that the NSCE method is not applicable to Hall because the method of Hall does not provide the camera parameters. Second, table 6.2 shows the results of calculating the accuracy by using the third and fourth criteria, respectively. Note that the first three calibrating methods which do not include the modelling of lens distortion (i.e. Hall, Faugeras-Toscani and iterative Faugeras-Toscani without distortion) obviously give the same accuracy with distorted and undistorted 2D points as has been considered $P_d = P_u$.

These tables show the accuracy obtained by each of the camera calibration techniques surveyed. It can be observed that the techniques, which do not model lens

---

[1] Newton-Raphson.

distortion (the first three rows in the tables) provide less accuracy than the others, which do model the lens. Moreover, the technique of Hall appears as the best linear method because it is based on computing the transformation matrix without including any constraint. The other two techniques are based on a model which imposes a determined form of the transformation matrix. This fact ill effects the calibration. However, the discrepancy between their accuracy is not significant. Furthermore, the results show that the use of an iterative algorithm does not improve the accuracy obtained by using the pseudo-inverse in the technique of Faugeras-Toscani without distortion. In order to improve accuracy it has to go to lens modelling.

It can be observed from the tables that the non-linear techniques, which model lens distortion (the last 6 rows of the tables), obviously obtain better results than the linear techniques. However, the improvement obtained by the method of Tsai without optimization (fifth row) is not very significant because only a few parameters are iteratively optimized (i.e. f, t_z and k_1). Nevertheless, when the whole set of parameters is optimized, the method of Tsai (sixth row) shows the best accuracy obtainable despite needing more computing time. Note that accuracy is limited due to image segmentation and also that the model used always approximates the real behavior of the image sensor. However, if a real principal point is known instead of the image center approximation, the Tsai method without optimization is as accurate as any iterative method, and allows a rapid computation. Note that the use of the Tsai optimized method by using the real principal point in the initial guess does not suggest an important improvement in accuracy. Finally, results show that any iterative method which models lens distortion provides the same accuracy without depending on the kind of modelled lens. That is, the complete method of Weng does not obtain a better accuracy than the simple iterative method of Faugeras modelling only radial distortion. Even so, the accuracy is slightly less due to the complexity of this model which ill effects the calibration. The modelling of a camera including a large quantity of parameters does not imply that the accuracy obtained will be better.

6.2.2 Camera Calibration with Real Images

The results obtained when using synthetic data allows us to compare the error produced by the camera model and the numerical method used in minimization.

When a real camera is calibrated, other factors modify the calibration error, such as the accuracy of the position of 3D object points used in the calibration and the error given by 2D image points measurement due to digitalization of 3D object points projection.

The next few pages will show the algorithm which clarifies how a camera is calibrated is described step by step. Then, an example of camera calibration is presented as an application of 3D reconstruction. Finally, a second example concerning a camera mounted on a mobile robot is calibrated and used in robot navigation.

Camera calibration step by step

The six steps to calibrate a camera are the following:

1. Place the camera to be calibrated in a location where the entire working area is viewed by the given camera.
2. Set a world coordinate system in a fixed position with respect to the camera.
3. Place a set of 3D object points spread throughout the working area. Some calibration methods use a set of coplanar 3D object points [Batista 99] while others use a set of non-coplanar 3D points, which is our case. The positions of object points with respect to the world coordinate system are given and their 2D image projections in the camera image plane must be measured. The number of object points depends on the size of the working area, however the minimum should be six. More accurate results are obtained when using at least twenty object points together with their 2D image projections well distributed throughout the image plane.
4. Capture an image of the working area and measure the 2D projections of each one of the object points. Depending on the distribution of the 3D object points, it is necessary to capture more than a single image, as in the case of 3D object points situated on a set of planes at different distances where an image might be captured for every plane.
5. Solve the correspondence between the 3D object points with their 2D image projections.

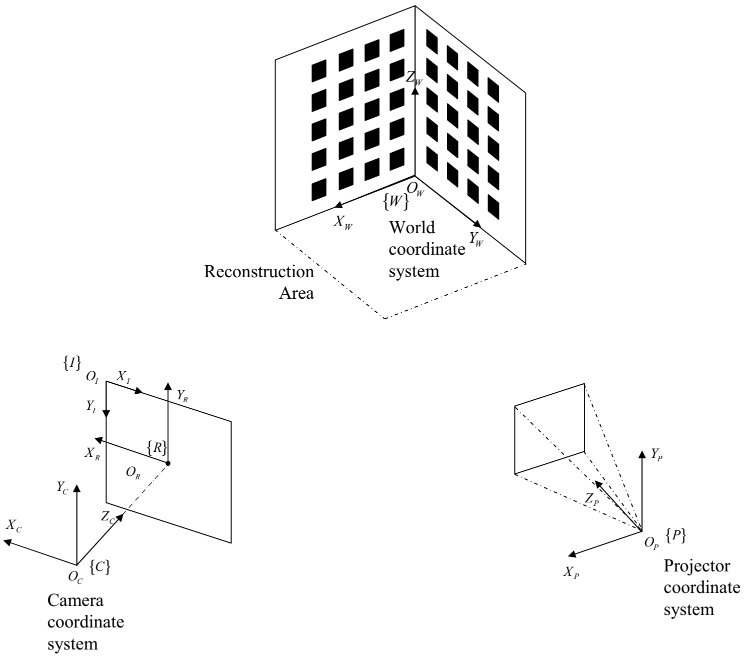


Figure 6.1: Camera and projector system for scene reconstruction.

6. Calibrate the camera. Sometimes it is necessary to know some parameters related to the CCD sensor and image size, which is usually provided by manufacturers as in the Tsai method.

Example of camera calibration in a reconstruction system

This is an example of calibrating a camera situated in a reconstruction system using a coded light projection technique [Pagès 03]. This system is composed of three elements: a camera, a light projector and a reconstruction area. Following the steps explained previously, we must fix a world coordinate system, in this case the origin has been situated in a corner of the reconstruction area (see Figure 6.1). After that, two white planes with black squares have been placed in the working area. The positions of squares corners are known and will be utilized in the calibration. Then, an image is grabbed, as shown in Figure 6.2. The squares corners are obtained by hand without subpixel accuracy. Finally, each 3D object point is related to its corresponding 2D image point to calibrate the camera.

Using these points, the camera can be calibrated with five methods. The calibra-

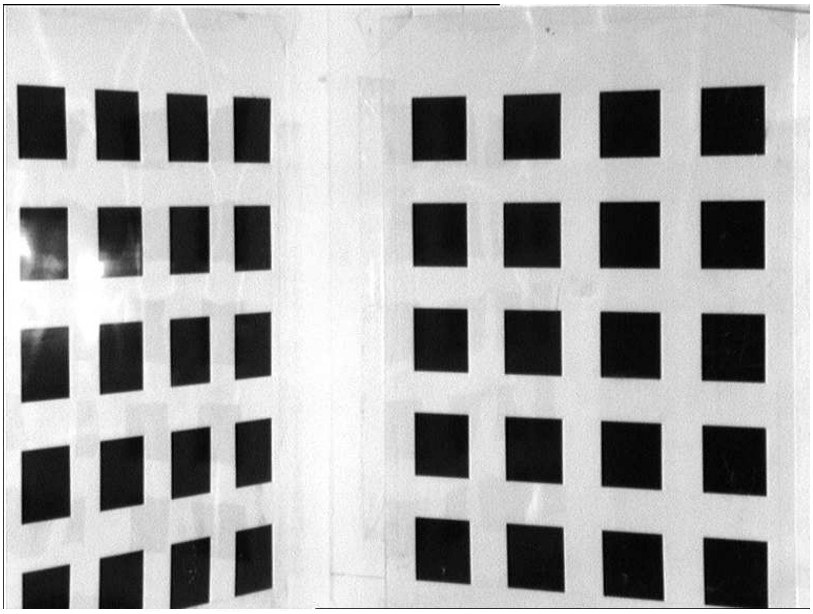


Figure 6.2: Image of the calibrating pattern viewed by a given camera.

Table 6.3: Computing time results for a fixed camera by using a set of 40 calibrating points.

	Time (ms.)
Hall	1
Faugeras	1
Faugeras with distortion	10
Tsai	10
Weng	51

tion accuracy has been computed for each method and shown in the following tables. Table 6.3 shows the computing time for each method using a laptop Pentium® III Computer at 1000 MHz. It is necessary to point out that linear methods (Hall and Faugeras) are faster than iterative methods and the Weng method is the slowest because it uses the most complex camera model.

In terms of accuracy, the measuring methods followed the same criteria previously presented in synthetic data. Table 6.4 shows the distance between the 3D object points and their optical ray. Moreover, the computation of the ratio called Normalized Stereo Calibration Error is also given. Table 6.5 shows the accuracy measured by using distorted and undistorted image coordinates in the computation. The large mean error (more than one pixel in all cases) is due to the manual segmentation of the 2D image points. The accuracy obtained by comparing tables 6.4 and

Table 6.4: Accuracy of 3D coordinate measurement for a fixed camera by using a set of 40 calibrating points.

	3D position (mm.)			NSCE
	Mean	σ	Max	
Hall	0.5219	0.2595	1.1370	n/a
Faugeras	0.7782	0.4253	2.0210	4.0649
Faugeras with distortion	0.4967	0.3367	1.5642	2.5489
Tsai	0.4815	0.3023	1.4014	2.4836
Weng	0.4740	0.2904	1.2669	2.4556

Table 6.5: Accuracy of 2D coordinate measurement for a fixed camera by using a set of 40 calibrating points.

	2D distorted image (pix.)			2D undist. image (pix.)		
	Mean	σ	Max	Mean	σ	Max
Hall	1.1124	0.5667	2.3322	1.1124	0.5667	2.3322
Faugeras	1.6549	0.9033	4.1251	1.6549	0.9033	4.1251
Faugeras with distortion	1.0661	0.7283	3.2365	1.0397	0.7283	3.2365
Tsai	1.0283	0.6474	2.8767	1.0137	0.6474	2.8767
Weng	1.0151	0.6217	2.6073	1.0010	0.6217	2.6073

6.5 with tables 6.1 and 6.2 is, in all cases, worse. In order to improve the calibration, it would be necessary to compute the 2D image points automatically, including subpixel accuracy. In short, iterative methods improve the accuracy considerably when compared to linear methods.

Example of camera calibration in a mobile robot

The second example considers the calibration of a camera placed on a mobile robot. The mobile robot is equipped with a stereo vision system (see Figure 6.3) which is further used in the robot navigation (see the application concerning the mapping of a robot localization in chapter 5). This section describes the calibration of one of the cameras (the left) of the stereo vision system. Note that the same procedure is applied to calibrate the other camera.

Following the same algorithm described at the beginning of this section, we have to first fix a world coordinate system. In this case, the origin of this world coordinate system cannot move with respect to the camera position. Therefore, it is not possible to fix such a coordinate system in any place in the scene due to the fact that it has to move according to the motion of the robot. In this case, the

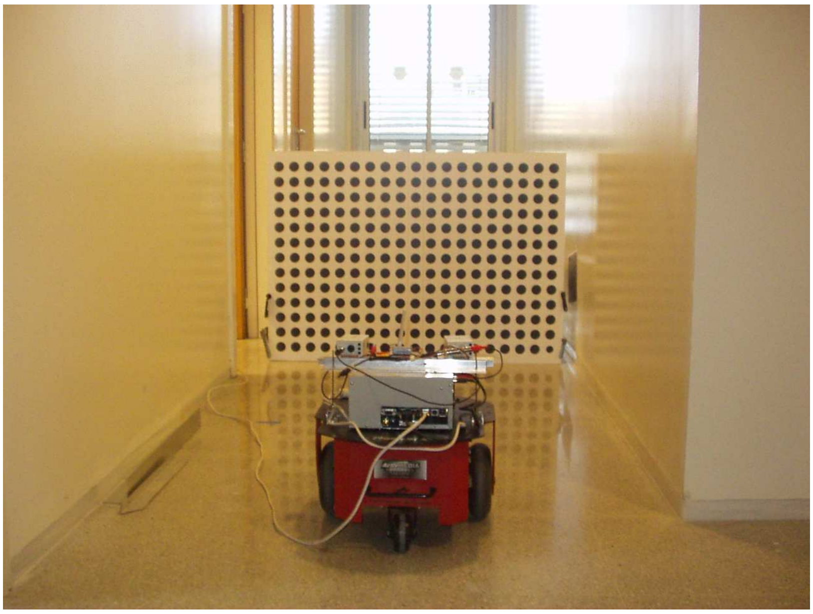


Figure 6.3: Stereo camera over a mobile robot.

world coordinate system coincides with the robot coordinate system whose origin is located at the bottom of the robot between the wheels. Now, a set of 3D object points spread throughout the working area in the camera view has to be located. Note that the range of distances from 1 to 4 meters in the chosen working area is quite large. This decision was taken considering that distant objects are irrelevant to robot navigation. Besides, closer objects are objects which have been previously seen by the system and already considered. We have not considered that any object could appear unexpectedly at distances closer than 1 meter.

The set of object points utilized in the camera calibration are structured into 9 planes placed from 0.8 meters to 4.40 meters with respect to the robot coordinate system. Each plane is composed of a set of equidistant black circles on a white background where each circle center the 3D coordinates are known. Then, an image is taken for every plane position (see Figure 6.4). The center of every circle in the image plane is computed with subpixel accuracy as the mass center. Note that the center of a circle coincides with the mass center only when the circle is perpendicular to the optical ray of the camera. Otherwise, the image of the circle has an egg-shape, leading to the use of other kinds of patterns, usually the vertexes of a square. Finally, the whole set of 3D points (1904 in our set up) is related to its 2D projective position to calibrate the camera.

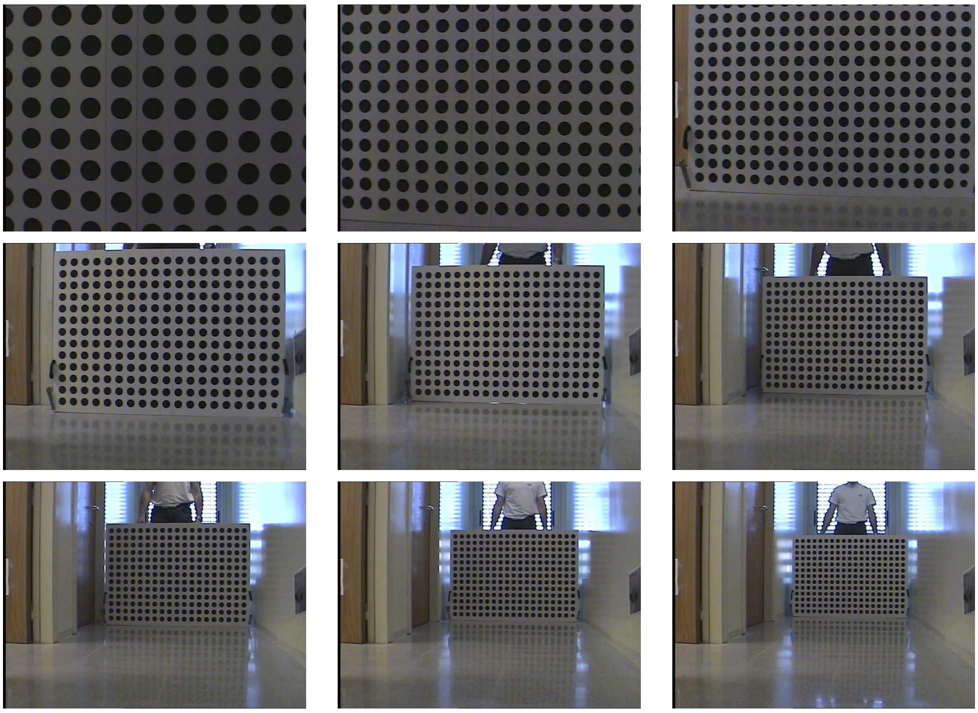


Figure 6.4: Images utilized to calibrate a camera on the mobile robot.

Table 6.6: Computing time results for a camera mounted on a mobile robot by using a set of 1904 calibrating points.

	Time (ms.)
Hall	70
Faugeras	70
Faugeras with distortion	380
Tsai	530
Weng	4216

Table 6.6 shows the camera calibration results in terms of computation time which are compared to the previous example (see Table 6.3). Note that the total time depends directly on the number of points used in the calibration with Weng's method always being the slowest.

Table 6.7 shows the accuracy when considering two different criteria. The first 3 columns show the discrepancy between each 3D object point with respect to its optical ray. The last column is the Normalized Stereo Calibration Error. Apparently, the comparison of the results given in Table 6.7 with the results previously shown in Table 6.4 when using only the first criteria shows a large discrepancy, indicating that the calibration of the first example was more accurate. This should be considered

Table 6.7: Accuracy of 3D coordinate measurement for a camera mounted on a mobile robot by using a set of 1904 calibrating points.

	3D position (mm)			NSCE
	Mean	σ	Max	
Hall	1.5698	0.9842	8.9249	n/a
Faugeras	1.6187	0.9856	8.8812	2.0175
Faugeras with distortion	0.9930	0.5660	3.2386	0.9909
Tsai	0.9927	0.5655	3.2311	0.9908
Weng	0.9896	0.5724	3.3526	0.9869

Table 6.8: Accuracy of 2D coordinate measurement for a camera mounted on a mobile robot by using a set of 1904 calibrating points.

	2D distorted image (pix.)			2D undist. image (pix.)		
	Mean	σ	Max	Mean	σ	Max
Hall	0.8061	0.8981	6.3560	0.8061	0.8981	6.3560
Faugeras	0.8235	0.9183	6.8152	0.8235	0.9183	6.8152
Faugeras with distortion	0.3996	0.2124	1.3225	0.4045	0.2172	1.3828
Tsai	0.3995	0.2124	1.3333	0.4045	0.2172	1.3944
Weng	0.3981	0.2105	1.3516	0.4029	0.2153	1.4344

a mistake because the first criteria depends on the distance of the 3D points with respect to the camera. Note that, in the first example, the calibration points were located closer (about one meter) to the camera than in the second example, in which they were located from one to four meters away. In conclusion, the first accuracy evaluation criteria is useless to compare calibrations in different camera configurations, but it is the most convenient to find the given error in metric coordinates. Besides, the NSCE ratio is independent of the distance of the calibration points with respect to the camera. Considering the second example, the NSCE ratio is smaller when in every method, leading us to conclude that the camera calibrations obtained are more accurate basically due to the fact that more calibrating points were used.

In Table 6.8 the reader can observe the accuracy measured with respect to distorted and undistorted image coordinates. The mean error is smaller compared to Table 6.5 due to the fact that the method utilized to extract the 2D image point coordinates used subpixel accuracy and was automatic.

6.2.3 Discussion

The surveyed methods have been implemented and their accuracy analyzed. Results show that only non-linear methods obtain a 3D accuracy smaller than 0.1 mm. with a reasonable standard deviation using synthetic data. Moreover, the accuracy of non-linear methods on the image plane is much higher than linear methods. Results show, moreover, that the modelling of radial distortion is quite sufficient when high accuracy is required. The use of more complicated models does not improve the accuracy significantly. It should be kept in mind that segmentation introduces a discrepancy between observable and modelled projections which poses conditions on the accuracy. Moreover, when low accuracy is sufficient, the fast and simple method of Hall is sufficient for most applications.

When comparing the results obtained using the same camera configuration, it can be seen that a relationship exists between the different criteria. Accuracy measuring methods obtain similar results if they are relatively compared. That is, good calibrating algorithms obtain acceptable accuracy results independently from the accuracy evaluation method used. Obviously, the results only prove something already demonstrated by the authors. However, in this chapter the accuracy has been measured by using the same test points for all the methods, so the results can be reliably compared. Hence, the reader can choose any method depending on the accuracy required.

Finally, the NSCE ratio of accuracy evaluation is the most convenient because it is independent of the distance between the object points and the calibrated camera. Besides, the methods used to evaluate the accuracy considering the image coordinates are the most suitable when the NSCE is not available, i.e. the Hall's method. Moreover, the accuracy method based on the 3D error measurement is not a good criteria to compare calibrations with different camera configurations, but is the most convenient when the reconstruction error measurement is needed.

6.3 Fundamental Matrix Estimation Experimental Results

The surveyed methods explained in section 3.4 have been implemented[2] and their accuracy analyzed with synthetic images varying the gaussian noise and the number of outliers. Moreover, the surveyed methods have been tested by using real images in different scenarios, that is urban scenes, mobile robot indoor environment, seabed, road images, aerial and images of a potential kitchen. Tables presented in this section show the accuracy of every method computed as the mean and standard deviation of the distances between points and epipolar lines.

6.3.1 Fundamental Matrix Estimation with Synthetic Images

Table 6.9 illustrates the results obtained with the linear methods. The *seven points* algorithm obtains a solution using only seven points. However, the accuracy depends greatly on the points used. The *least-squares technique* is based on using at least 8 points and its accuracy depends on the amount of badly located points used, usually obtaining better results by increasing the amount of points. The *eigen analysis* is the linear method that obtains the best results because an orthogonal least-squares minimization is more realistic than the classical one. However, all these methods obtain a rank-3 fundamental matrix, which means that the epipolar geometry is not properly modeled. The *analytic method with rank-2 constraint* obtains a rank-2 fundamental matrix in which distances between points and epipolar lines are worse than in the linear methods and it is more expansive in computing time (see Figure 6.5).

Table 6.10 presents results obtained with iterative methods. The *iterative linear method* improves considerably the least-squares technique but can not cope with outliers. The *iterative Newton-Raphson algorithm* gets even better results than the previous method if the presence of outliers is not considered. Although the *nonlinear minimization in parameter space* also obtains a rank-2 matrix, the distances of points to epipolar lines are the worst and sometimes the method diverges obtaining a false

---

[2] FNS, CFNS, MLESAC and MAPSAC implementations have been provided by the original authors.

Table 6.9: Synthetic image results of linear methods[3]. Every cell show the mean and standard deviation of the discrepancy between points and epipolar lines.

Methods	**Linear**			
	1	**2**	**3**	**4**
$\sigma = 0.0$	14.250	0.000	0.000	1.920
outliers 0%	13.840	0.000	0.000	1.143
$\sigma = 0.0$	25.370	339.562	17.124	30.027
outliers 10%	48.428	433.013	31.204	59.471
$\sigma = 0.1$	135.775	1.331	0.107	0.120
outliers 0%	104.671	0.788	0.088	0.091
$\sigma = 0.1$	140.637	476.841	19.675	70.053
outliers 10%	104.385	762.756	46.505	63.974
$\sigma = 0.5$	163.839	5.548	0.538	0.642
outliers 0%	178.222	3.386	0.362	0.528
$\sigma = 0.5$	140.932	507.653	19.262	26.475
outliers 10%	109.427	1340.808	49.243	54.067
$\sigma = 1.0$	65.121	21.275	1.065	1.319
outliers 0%	58.184	12.747	0.744	0.912
$\sigma = 1.0$	128.919	429.326	21.264	61.206
outliers 10%	100.005	633.019	53.481	64.583

solution. The eighth and ninth methods are two different versions of the *gradient-based method* using least-squares and orthogonal least-squares, respectively. Both methods obtain better results than their equivalent linear methods. Nevertheless, the eigen analysis once more obtains better results than the other linear methods. Results obtained and computing time spent by the method *FNS* are quite similar to the gradient technique. Besides, *CFNS* improve slightly the results obtained by FNS, but spending more computing time. Summarizing, iterative methods improve the computation of the fundamental matrix but they cannot cope with outliers.

The last surveyed methods are classified into robust (see in Table 6.11), which means that they might detect and remove potential outliers and compute the fundamental matrix by using only inliers. Three versions of the *M-estimators* based on the Huber weight function have been programmed: least-squares, eigen analysis and the method proposed by Torr [Torr 97]. The three methods start from a linear initial guess and become fully dependent on the linear method used to estimate it. Moreover, least-squares and eigen values get a rank-3 matrix, while Torr forces a rank-2 matrix in each iteration giving a more accurate geometry. Besides, two

---

[3] Methods: 1.- seven points; 2.- least-squares (LS) 3.- orthogonal LS; 4.- rank-2 constraint.

Table 6.10: Synthetic image results of iterative methods[4]. Every cell show the mean and standard deviation of the discrepancy between points and epipolar lines.

Methods	**Iterative**						
	5	**6**	**7**	**8**	**9**	**10**	**11**
$\sigma = 0.0$ outliers 0%	0.000 0.000	0.000 0.000	0.000 0.000	0.000 0.000	0.000 0.000	0.000 0.000	0.000 0.000
$\sigma = 0.0$ outliers 10%	161.684 117.494	20.445 30.487	∞ ∞	187.474 197.049	18.224 36.141	17.124 31.204	16.978 29.015
$\sigma = 0.1$ outliers 0%	1.328 0.786	0.107 0.088	1.641 0.854	1.328 0.786	0.112 0.092	0.107 0.088	0.110 0.091
$\sigma = 0.1$ outliers 10%	158.961 124.202	32.765 67.308	146.955 94.323	183.961 137.294	15.807 40.301	14.003 38.485	14.897 39.388
$\sigma = 0.5$ outliers 0%	5.599 3.416	0.538 0.361	7.017 3.713	5.590 3.410	0.554 0.361	0.538 0.362	0.543 0.368
$\sigma = 0.5$ outliers 10%	161.210 136.828	31.740 59.126	∞ ∞	217.577 368.061	19.409 51.154	22.302 59.048	22.262 59.162
$\sigma = 1.0$ outliers 0%	20.757 12.467	1.068 0.772	345.123 294.176	21.234 12.719	1.071 0.745	1.065 0.744	1.066 0.748
$\sigma = 1.0$ outliers 10%	158.849 120.461	37.480 52.762	∞ ∞	152.906 120.827	18.730 38.644	18.374 39.993	19.683 42.112

different versions of *LMedS* using again least-squares and eigen analysis have been studied. Although the accuracy of LMedS seems worse compared to M-estimators, LMedS removes the outliers more efficiently so that the epipolar geometry is properly obtained. *RANSAC* is the last surveyed method. However, RANSAC does not obtains any better results than LMedS with eigen analysis due to the method used to select the outliers which is quite permissive. MLESAC is a generalization of RANSAC obtaining more or less the same results. Besides, *MAPSAC* improves considerably the results obtained by RANSAC but MAPSAC does not improve the results obtained by LMedS.

Figure 6.5 shows the mean computing time spent by the whole methods in synthetic and real scenarios. On the whole, computing time is linear dependent to complexity of the algorithm. So, least-squares turn out to be the quickest linear method, while Newton-Raphson and gradient techniques are the quickest iterative methods. Summarizing the robust methods, M-Estimators are quicker than the methods in which a set of points have to be selected aleatory from the images.

---

[4] Methods: 5.- iterative linear; 6.- iterative Newton-Raphson; 7.- minimization in parameter space; 8.- gradient using LS; 9.- gradient using eigen; 10.- FNS; 11.- CFNS.

[5] Methods: 12.- M-Estimator using LS; 13.- M-Estimator using eigen; 14.- M-Estimator pro-

Table 6.11: Synthetic image results of robust methods[5]. Every cell show the mean and standard deviation of the discrepancy between points and epipolar lines.

Methods	**Robust**							
	12	**13**	**14**	**15**	**16**	**17**	**18**	**19**
$\sigma = 0.0$	0.000	0.000	0.000	0.000	0.000	0.000	0.100	0.011
outliers 0%	0.000	0.000	0.000	0.000	0.000	0.000	0.079	0.009
$\sigma = 0.0$	273.403	4.909	4.714	0.000	0.000	16.457	19.375	0.115
outliers 10%	360.443	4.493	2.994	0.000	0.000	26.923	70.160	0.115
$\sigma = 0.1$	0.355	0.062	0.062	1.331	0.107	0.107	0.139	0.168
outliers 0%	0.257	0.042	0.041	0.788	0.088	0.088	0.123	0.155
$\sigma = 0.1$	73.354	4.876	4.130	0.449	0.098	2.389	21.784	0.701
outliers 10%	59.072	4.808	2.997	0.271	0.077	5.763	97.396	0.740
$\sigma = 0.5$	2.062	0.392	0.367	5.548	0.538	0.538	0.550	0.762
outliers 0%	1.466	0.237	0.207	3.386	0.362	0.362	0.377	0.618
$\sigma = 0.5$	143.442	3.887	3.147	47.418	0.586	18.942	23.859	0.629
outliers 10%	111.694	3.969	2.883	29.912	0.434	53.098	79.890	0.452
$\sigma = 1.0$	8.538	0.794	0.814	21.275	1.065	1.065	1.089	1.072
outliers 0%	6.306	0.463	0.463	12.747	0.744	0.744	0.768	0.785
$\sigma = 1.0$	120.012	3.921	4.089	25.759	1.052	14.076	19.298	1.041
outliers 10%	122.436	3.752	4.326	15.217	0.803	30.274	65.149	0.822

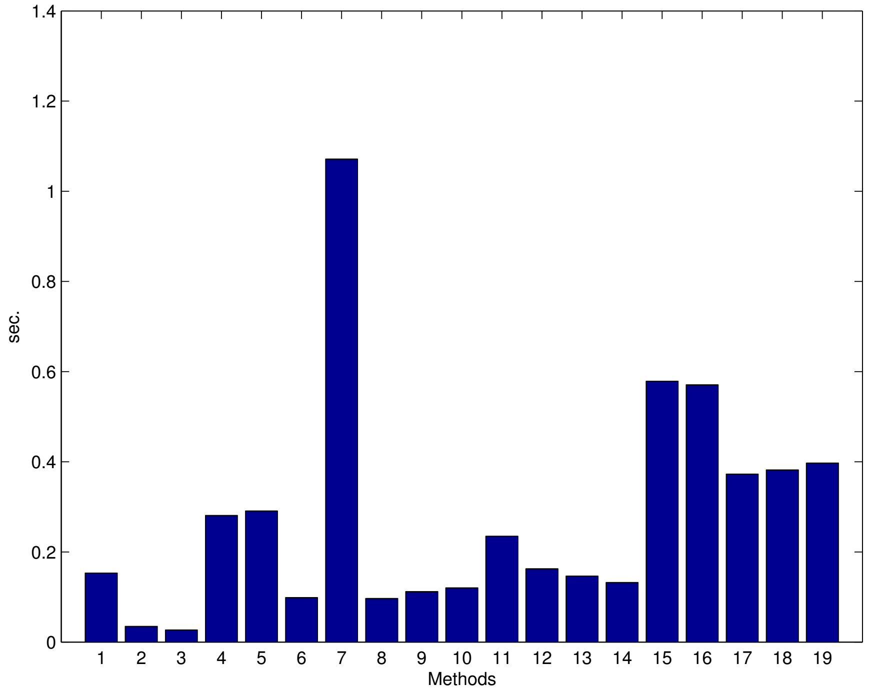


Figure 6.5: Computing time of methods[3,4,5].

6.3.2 Fundamental Matrix Estimation with Real Images

The surveyed methods have been tested by using real images in different scenarios, that is, urban scenes, mobile robot indoor environment, aerial and seabed outdoors, road images, and the special case of a kitchen scene (see Figure 6.6). The correspondence problem between images have been solved by using the algorithm proposed by Zhang[6] [Zhang 94] and the obtained results are shown in Figure 6.7. The corresponding points have normalized by the method proposed by Hartley [Hartley 95] described in section 3.4.5.

Continuing with the methodology applied with synthetic images, the following Tables 6.12, 6.13 and 6.14 show the results obtained using real images with linear, iterative and robust methods, respectively.

As it has been observed with synthetic images, robust methods improve the accuracy of linear and iterative methods. For example a robust method should be used in the mobile robot scene where there are a lot of outliers. Nevertheless, if the number of outliers is can be neglected, an iterative method is enough, that is the case of the urban and aerial scenes. Finally, a linear method is the most suitable where there are no outliers since it is the one that minimizes the computing time, i.e. in the case of the road scene.

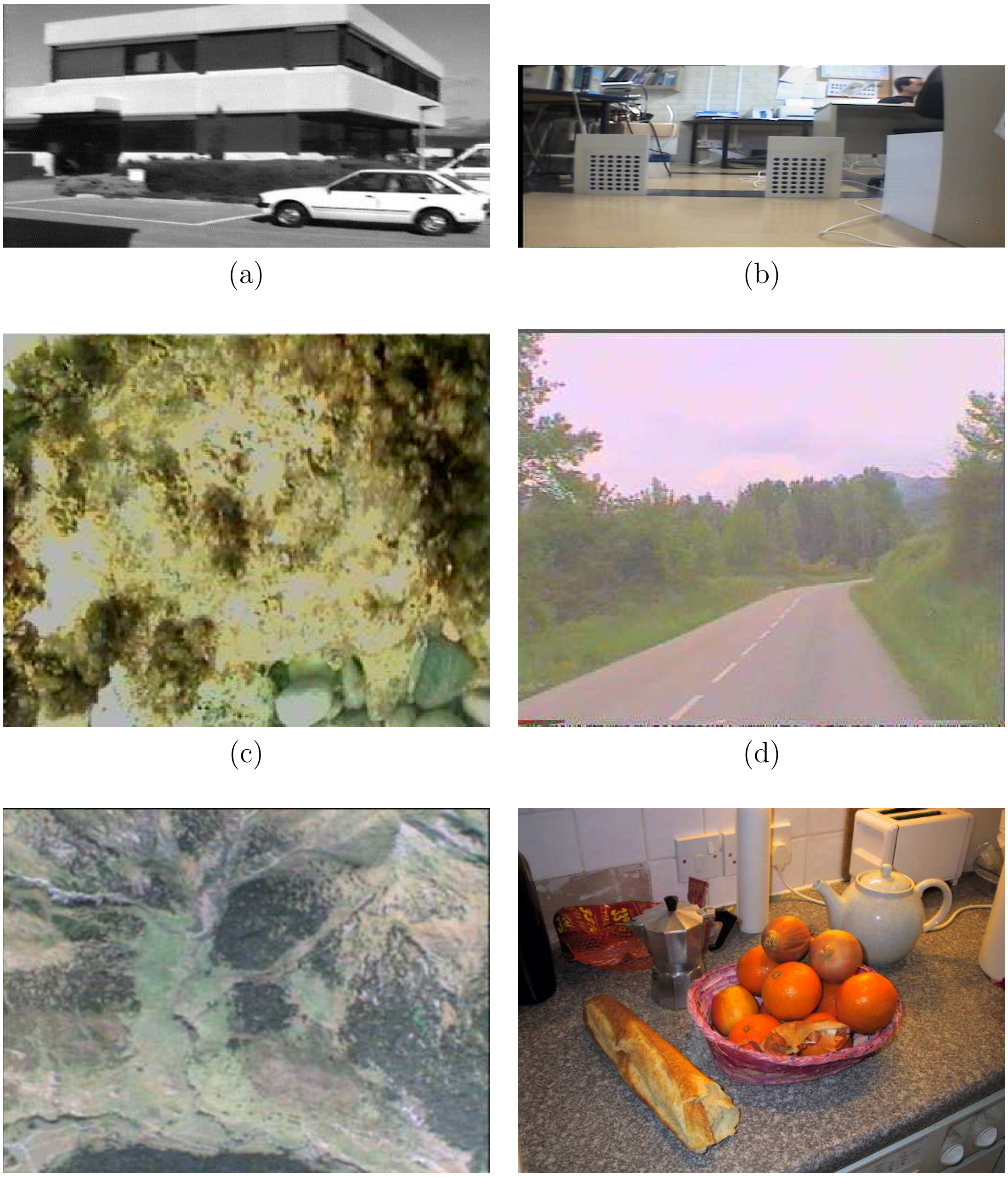
Figure 6.8a shows the matchings obtained by using the method proposed by Zhengyou Zhang [Zhang 93, Zhang 94]. First, a Harris corner detector is applied to obtain a list of interest points. Then the matching between both images is computed by using a pixel-based correlation. Note that matches might not be unique. Finally, a relaxation method is used to improve the local consistency of matches, reducing their ambiguity.

Figure 6.8b shows the list of matchings kept by M-estimator based on eigen values. Depending on the weighting function, the removed matchings vary due to both noise and outliers. Note that some good matchings are also removed while potential outliers are kept as inliers. Figure 6.8c shows the results obtained by LMedS, while Figure 6.8d shows the results obtained by RANSAC. In both cases, every single outlier is detected and removed, obtaining comparatively the same results.

---

posed by Torr; 15.- LMedS using LS; 16.- LMedS using eigen; 17.- RANSAC; 18.- MLESAC; 19.- MAPSAC.

[6] Available at `http://www-sop.inria.fr/robotvis/demo/f-http/html`

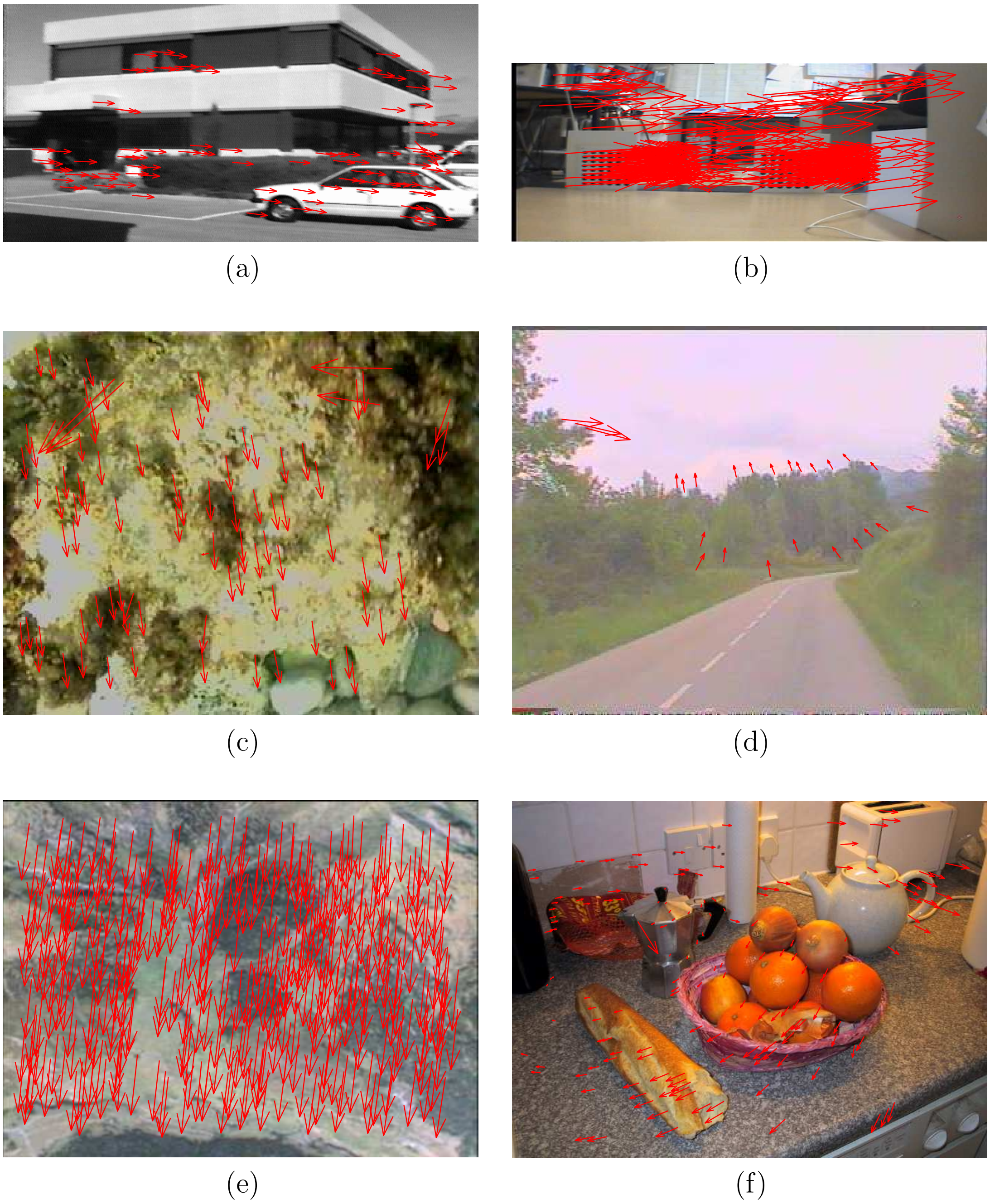


(a) (b)

(c) (d)

(e) (f)

Figure 6.6: Real images utilized to fundamental matrix estimation: (a) Urban scene; (b) Mobile Robot Scene; (c) Underwater Scene; (d) Road Scene; (e) Aerial Scene; (f) Kitchen Scene.



(a) (b) (c) (d) (e) (f)

Figure 6.7: Correspondences used in real images: (a) Urban scene; (b) Mobile Robot Scene; (c) Underwater Scene; (d) Road Scene; (e) Aerial Scene; (f) Kitchen Scene.

Table 6.12: Real image results of linear methods[7]. Every cell show the mean and standard deviation of the discrepancy between points and epipolar lines.

Methods	**Linear**			
	1	**2**	**3**	**4**
Urban Scene	51.633	1.724	0.440	1.023
	35.724	1.159	0.334	1.012
Mobile Robot Scene	119.439	35.525	4.080	16.511
	46.268	64.175	7.684	18.964
Underwater Scene	97.977	4.683	1.725	5.242
	66.223	3.941	2.138	4.286
Road Scene	27.668	0.825	0.609	1.078
	39.688	1.144	0.734	2.118
Aerial Scene	99.635	0.179	0.149	1.480
	62.162	0.158	0.142	0.979
Kitchen Scene	16.956	5.014	2.623	2.681
	16.696	5.177	3.327	4.175

Table 6.13: Real image results of iterative methods[8]. Every cell show the mean and standard deviation of the discrepancy between points and epipolar lines.

Methods	**Iterative**						
	5	**6**	**7**	**8**	**9**	**10**	**11**
Urban Scene	1.102	0.468	2.974	1.109	0.446	0.437	0.437
	0.796	0.341	3.066	0.803	0.368	0.333	0.334
Mobile Robot Scene	46.216	5.611	24.010	18.665	4.787	4.080	3.199
	35.011	8.729	22.270	22.170	9.255	7.684	5.541
Underwater Scene	3.068	1.752	5.575	2.949	1.581	1.599	1.609
	2.804	2.249	4.337	2.798	2.056	2.019	2.010
Road Scene	0.511	0.559	1.920	0.512	0.809	0.466	0.595
	0.422	0.709	2.498	0.427	0.986	0.419	0.543
Aerial Scene	0.179	0.149	0.497	0.179	0.342	0.149	0.209
	0.158	0.143	0.472	0.158	0.339	0.142	0.178
Kitchen Scene	3.217	2.966	7.563	3.176	1.901	2.623	1.892
	3.310	5.576	2.508	1.499	3.327	2.504	0.425

Table 6.14: Real image results of robust methods[9]. Every cell show the mean and standard deviation of the discrepancy between points and epipolar lines.

Methods	**Robust**							
	12	**13**	**14**	**15**	**16**	**17**	**18**	**19**
Urban Scene	1.668 0.935	0.309 0.228	0.279 0.189	1.724 1.159	0.319 0.269	0.440 0.334	0.449 0.373	0.440 0.348
Mobile Robot Scene	5.775 50.701	0.274 0.192	0.593 0.524	24.835 38.434	1.559 2.715	3.855 6.141	2.443 5.629	1.274 2.036
Underwater Scene	0.557 0.441	0.650 0.629	0.475 0.368	2.439 2.205	0.847 0.740	1.725 2.138	3.678 12.662	1.000 0.761
Road Scene	0.373 0.635	0.136 0.113	0.310 0.256	0.825 1.144	0.609 0.734	0.609 0.734	0.427 0.410	0.471 0.403
Aerial Scene	0.099 0.063	0.085 0.058	0.161 0.106	0.179 0.158	0.149 0.142	0.149 0.142	0.216 0.186	0.257 0.197
Kitchen Scene	0.584 0.425	0.280 0.207	0.263 0.191	1.350 1.200	0.545 0.686	2.623 3.327	0.864 3.713	0.582 0.717

Also, the geometry modeled by every robust method is quite different. Figures 6.9a and 6.9b show the epipolar geometry given by M-Estimator based on eigen values, wherein it is shown how the epipolar lines do not cross in a single epipole due to the rank-3 matrix obtained. LMedS obtains a completely different geometry in which epipoles have been located outside the image plane, but they are unique (see Figure 6.9c and Figure 6.9d). RANSAC obtains a geometry with the epipole located near the image centre. Comparing the obtained geometries related to the position of the camera and its motion, the geometry modeled by RANSAC is the closest to reality.

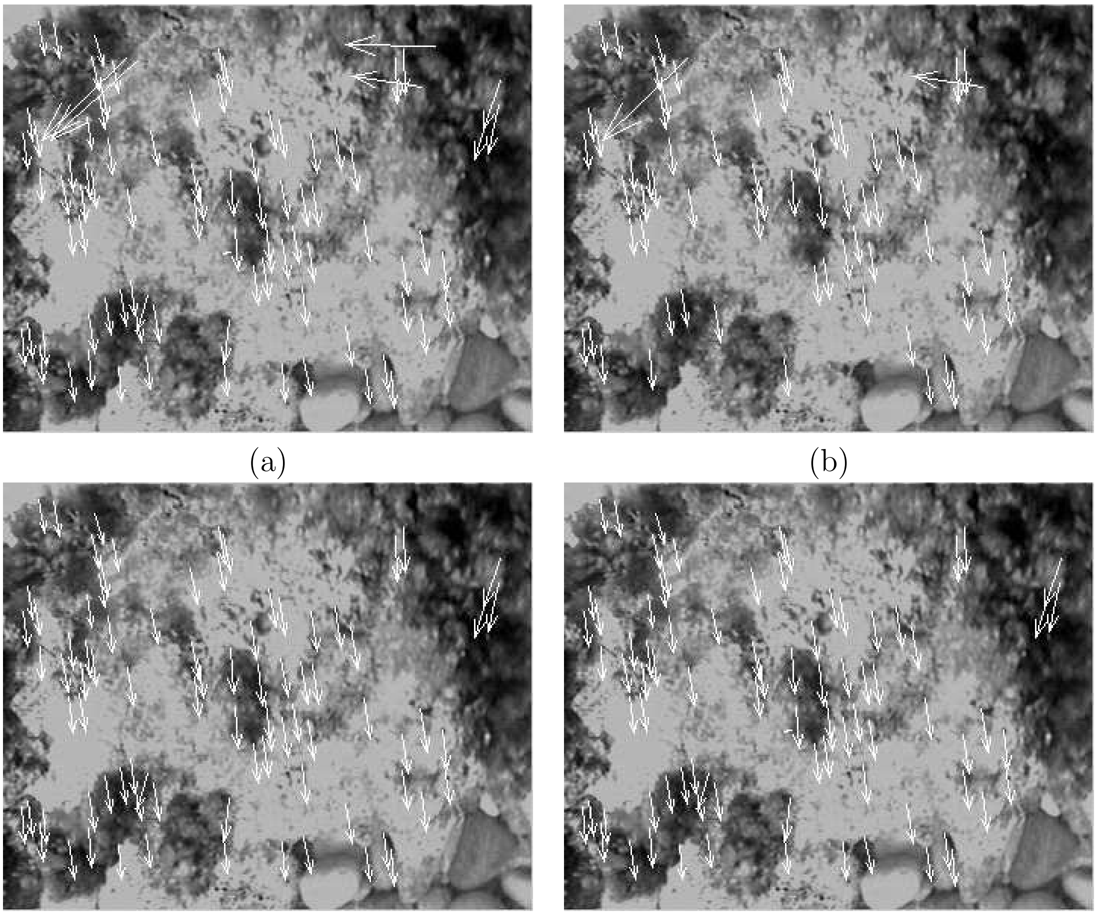
The same study has been done considering the urban scene showing that the obtained results are a bit different. The reader can see these results in Figure 6.10 and Figure 6.11. The number of potential outliers is lower than in the underwater scene and the location of image points is more accurate because of better image quality (see Figure 6.10a). Figure 6.10b shows the poor results obtained by the eigen value M-Estimator, in which a lot of matchings are removed while some of the outliers remain. In this case, LMedS is the only method which detects the set of

---

[7] Methods: 1.- seven points; 2.- least-squares (LS) 3.- orthogonal LS; 4.- rank-2 constraint.

[8] Methods: 5.- iterative linear; 6.- iterative Newton-Raphson; 7.- minimization in parameter space; 8.- gradient using LS; 9.- gradient using eigen; 10.- FNS; 11.- CFNS.

[9] Methods: 12.- M-Estimator using LS; 13.- M-Estimator using eigen; 14.- M-Estimator proposed by Torr; 15.- LMedS using LS; 16.- LMedS using eigen; 17.- RANSAC; 18.- MLESAC; 19.- MAPSAC.

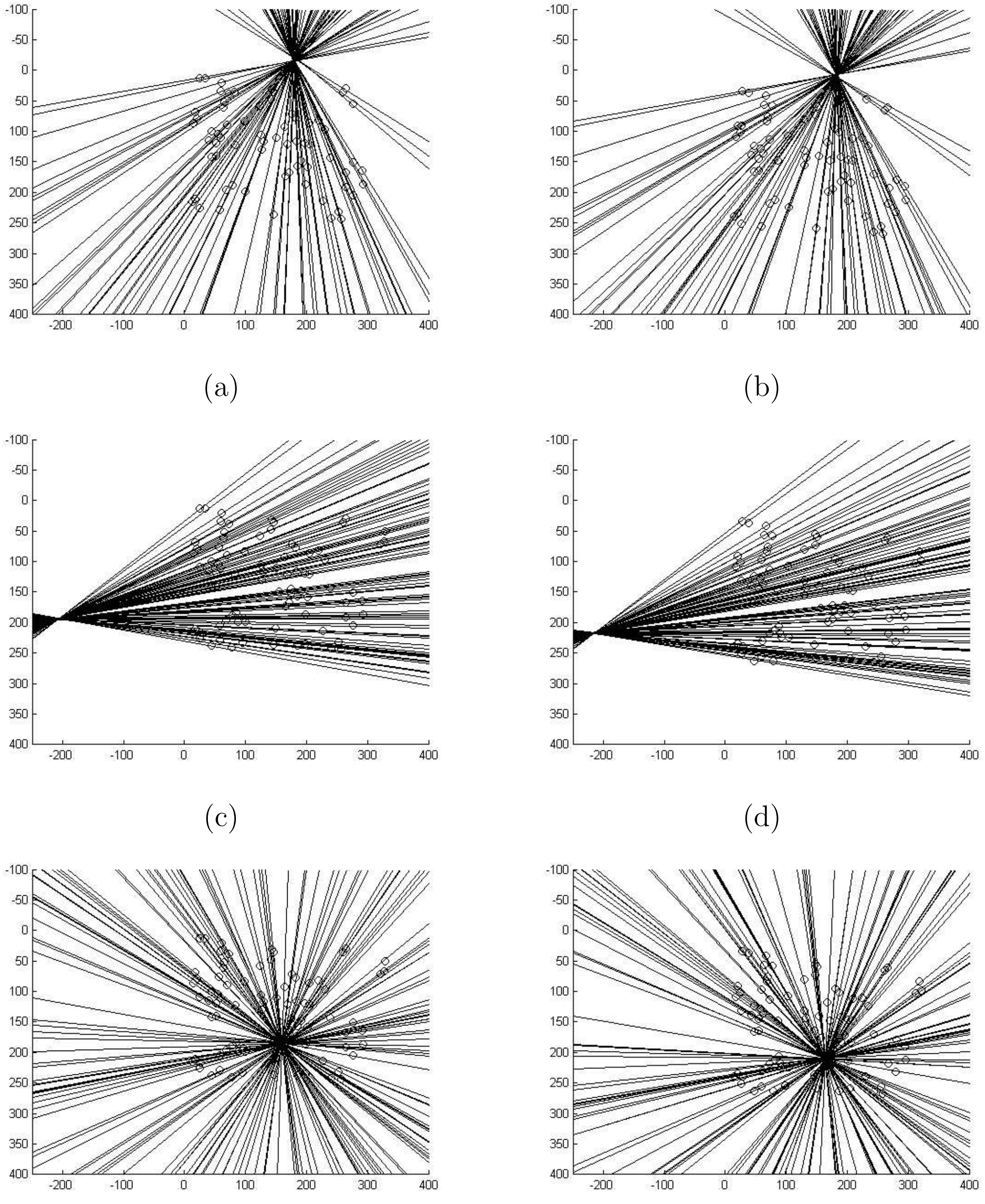


(a) (b) (c) (d)

Figure 6.8: Underwater scene and matchings: (a) set of initial correspondences; and the remaining matchings: (b) M-Estimators; (c) LMedS; (d) RANSAC.

outliers located at the right side of the image (see Figure 6.10c). Besides, RANSAC does not detect any outlier so results are not accurate enough.

The geometry obtained in the urban scene largely depends on the method utilized. Figure 6.11 shows the three different geometries given by M-Estimator, LMedS and RANSAC. In this case, M-Estimator and RANSAC model a similar geometry in which the epipoles are located outside the image near the top-right corner, which is not the right situation. LMedS obtains the right geometry with the epipoles located in the left side of the image.



(a)

(b)

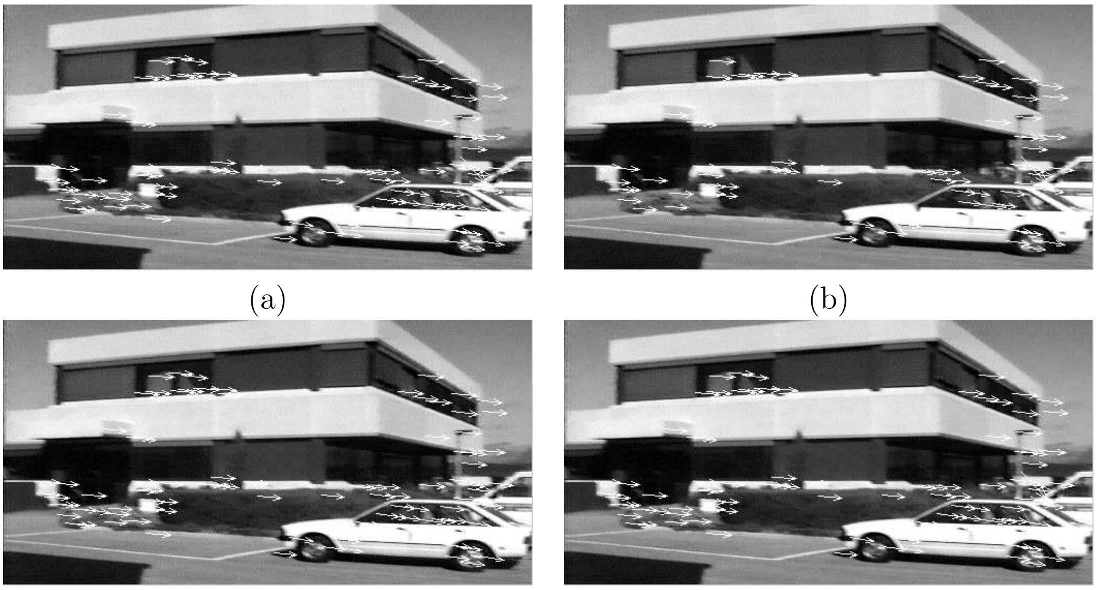
(c)

(d)

(e)

(f)

Figure 6.9: Points and epipolar lines in the underwater scene: (a) left and (b) right views obtained by M-Estimator; (c) left and (d) right views obtained by LMedS; (e) left and (f) right views obtained by RANSAC.



(a) (b)

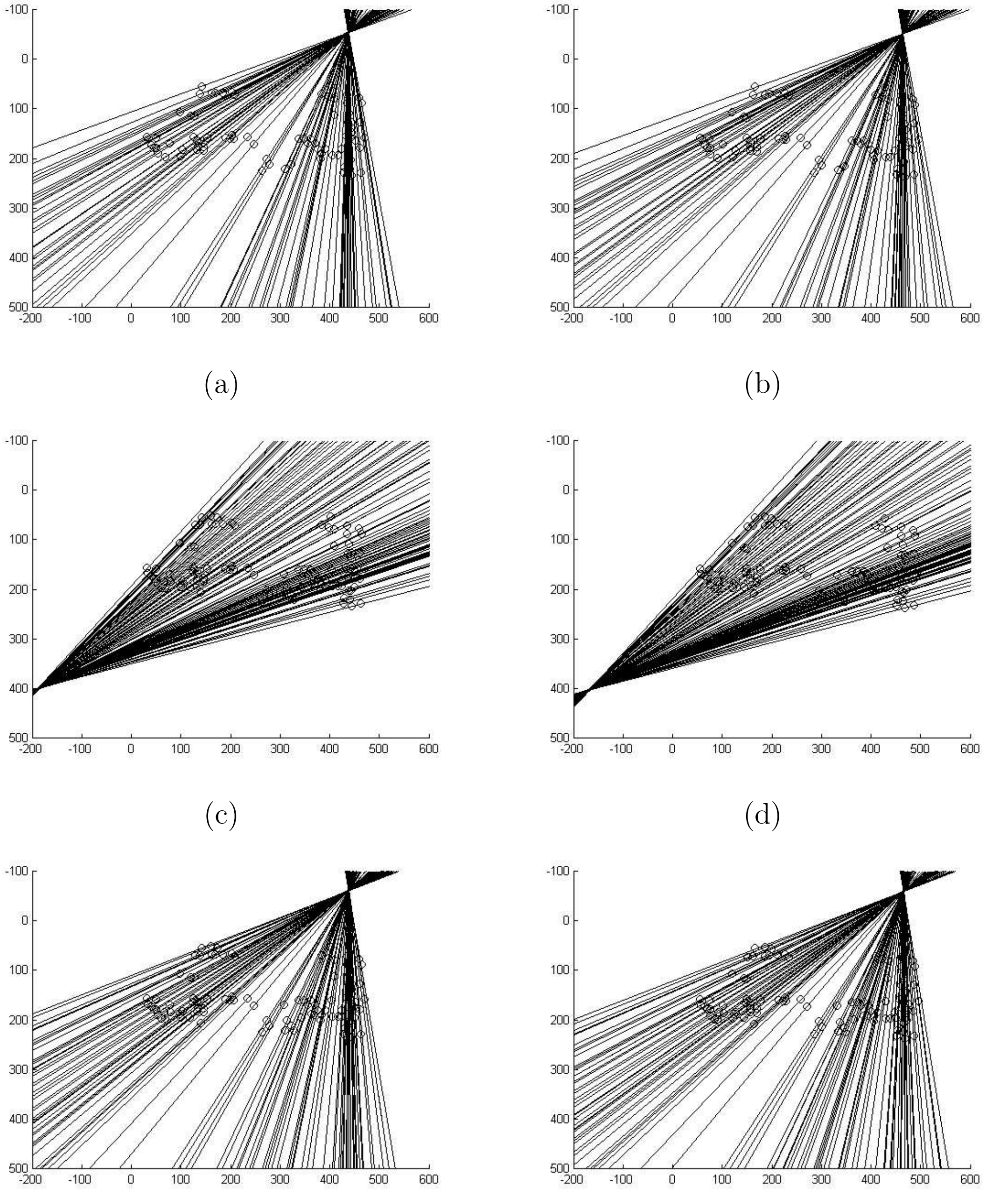
(c) (d)

Figure 6.10: Urban scene and matchings: (a) set of initial correspondences; and the remaining matchings: (b) M-Estimators; (c) LMedS; (d) RANSAC.

6.3.3 Discussion

The different methods have been programmed and their accuracy analyzed in synthetic and real images. The methodology used has been compared and a useful overall schema is presented. Experimental results show that: a) linear methods are quite good if the points are well located in the image and the correspondence problem previously solved; b) iterative methods can cope with some gaussian noise in the localization of points, but become really inefficient in the presence of outliers; and c) robust methods can cope with both discrepancy in the localization of points and false matchings. Nevertheless, the linear methods present an interesting reduced computing time.

Experimental results show that the orthogonal least-squares method using eigen analysis gives better results than the classic least-squares technique of minimization. Moreover, a rank-2 method is preferred because it models the epipolar geometry with all the epipolar lines intersecting at a single epipole. Moreover, experimental results show that the corresponding points have to be normalized and the best results have been obtained by using the method proposed by Hartley [Hartley 00]. In conclusion, the recently proposed method of MAPSAC obtains quite good results



(a) (b)

(c) (d)

(e) (f)

Figure 6.11: Points and epipolar lines in the urban scene: (a) left and (b) right views obtained by M-Estimator; (c) left and (d) right views obtained by LMedS; (e) left and (f) right views obtained by RANSAC.

with a low computing time. However, LMedS still obtains the best results when a low computing time is not required.

The uncertainty in fundamental matrix computation was studied in detail by Csurka et al. [Csurka 97] and Torr and Zisserman [Torr 98]. The surveyed methods model the epipolar geometry without considering lens distortion which considerably influences their discrepancy. Thus, some efforts have been made recently in the presence of radial lens distortion [Zhang 96a]. Among all, LMedS is the most appropriate for outlier detection and removal. However, with the aim of obtaining an accurate geometry, it is better to combine it with M-Estimator, which, in our case, modeled a proper geometry in synthetic data, both in the presence of noise and outliers.

6.4 Differential Epipolar Constraint Experimental Results

All the surveyed methods have been programmed and tested under the same conditions of image noise with the aim of giving an exhaustive comparison of most of 6-DOF motion estimation methods. Hence, section 6.4.1 compares the twelve surveyed methods of 3D motion estimation and section 6.4.2 deals with the six proposed adaptations to a 2-DOF mobile robot movement estimation. Section 6.4.3 shows results in real image sequences.

6.4.1 Results on 3D Motion Estimation

The surveyed methods based on the differential epipolar constraint explained in sections 4.4.1 have been programmed in MATLAB®. The others, that is, the ones described section 4.4.2, have been taken from the comparative survey and MATLAB® toolbox given by Tian, Tomasi and Heeger [Tian 96]. The partial use of a previous toolbox permits us to validate the programmed methods and compare the obtained results.

Several tests were done using synthetic data with the goal of comparing the robustness of the methods in the presence of image noise. We have used a methodology similar to the one proposed by Tian et al. [Tian 96] and Ma et al. [Ma 00]. Moreover, the camera movement is estimated from a cloud of 50 3D points located in front of the camera and distributed throughout the field of view image (we considered a field of view varying between 30° and 90°). Next, the optical flow of every point is computed. Once the optical flow of the 50 points is computed, gaussian noise is added to every velocity component with a standard deviation varying from 0.05 pixels up to 0.5 pixels. With the aim of studying the robustness of every method in any potential camera movement, all the potential camera orientations and translations are considered in ranges of 22.5°. Ten movement estimations are carried out for every camera pose.

Then the optical flow of every point is computed by using equation (4.51), in which $\dot{q}$ is the velocity of the image point $q = (q_1, q_2, 1)$ on the image plane; υ and ω are the camera's linear and angular velocities; and $Z(q)$ is the depth of every pixel q.

In the following experiments the angular velocity has been considered fixed and equal to 0.23°/frame, while the coefficient of linear/angular velocity varies from 1 up to 10. Once the optical flow of the 50 points is computed, a gaussian noise is added to every velocity component varying its standard deviation from 0.05 up to 0.5 pixels. With the aim of studying the robustness of every method in any potential camera movement, all the camera orientations and translations have been considered in ranges of 22.5° and 10 movement estimations are carried out for every camera pose. Hence, given an image field of view of 30°, 60° and 90°, a linear/angular coefficient of 1, 5 and 10 and a gaussian noise of 0.05 up to 0.5 pixels, an amount of 655,360 movement estimations have been computed for each surveyed method. Every estimation has been compared to the real movement where the discrepancy in the linear velocity estimation is the angle between the real movement vector υ and the estimated υ_{est} which is computed using the following equation.

$$error_{lineal} = \cos^{-1}(\upsilon \cdot \upsilon_{est}) \quad . \tag{6.1}$$

The discrepancy between the rotation matrix of the real angular movement $\mathbf{R}$ with respect to the estimated rotation matrix $\mathbf{R_{est}}$ obtained from the vector of angular velocities ω_{est} is used to compute the angular velocity error. Then, the difference rotation matrix is defined as follows, $\Delta\mathbf{R} = \mathbf{R}^{\mathrm{T}}\,\mathbf{R_{est}}$. The matrix $\Delta\mathbf{R}$ is defined by a rotation axis and an angle. The measuring error presented in this angle

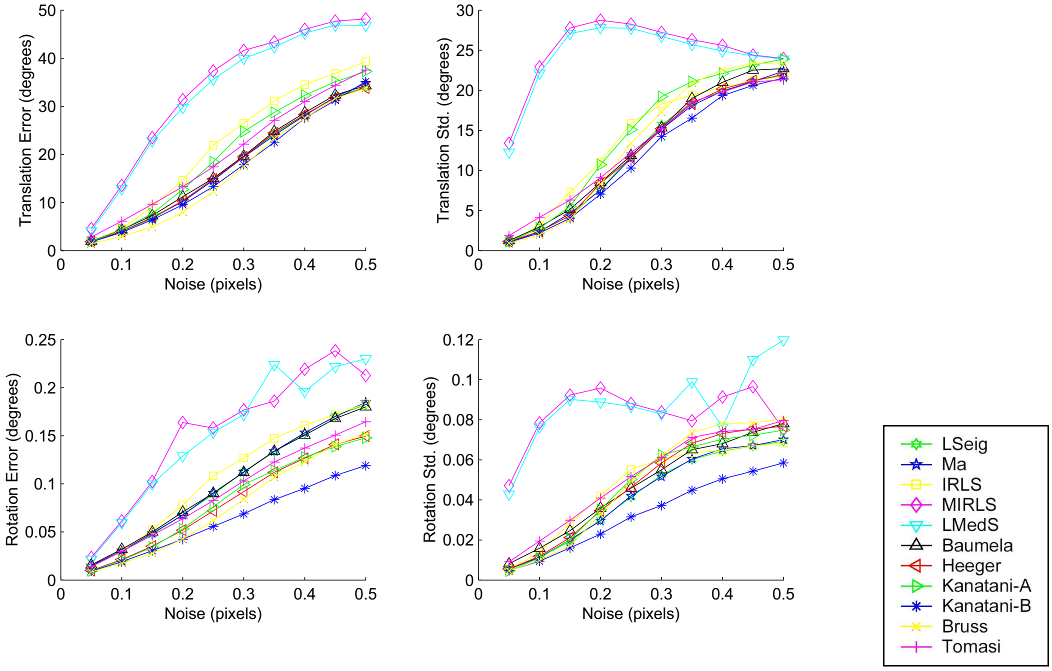


Figure 6.12: Estimation results of general methods with synthetic dates.

is computed by using the following equation.

$$error_{angular} = \cos^{-1}\left(\frac{\mathrm{Tr}(\Delta\mathbf{R}) - 1}{2}\right) \tag{6.2}$$

where $\mathrm{Tr}(\Delta\mathbf{R})$ is the trace of the matrix.

Figure 6.12 shows the obtained results of every method considering a linear/angular velocity coefficient equal to unity, an image field of view of 90° and a gaussian noise varying from 0.05 up to 0.5 pixels. This is a worst case, which leads us to compare the robustness of every method. The best results are obtained when the linear/angular velocity coefficient is bigger than the unity and the image field of view smaller than 90°.

The 7-points method and the method proposed by Prazdny are not shown in Figure 6.12 due to the poor results obtained which condition the illustrative comparison of the Figure. The Figure shows errors in the linear velocity angle and its standard deviation and errors in the angular velocity angle and its standard deviation.

Summarizing, the surveyed methods which present the worst results are the Modified Iteratively Reweighed Least Squares (MIRLS) and the Least Median Squares

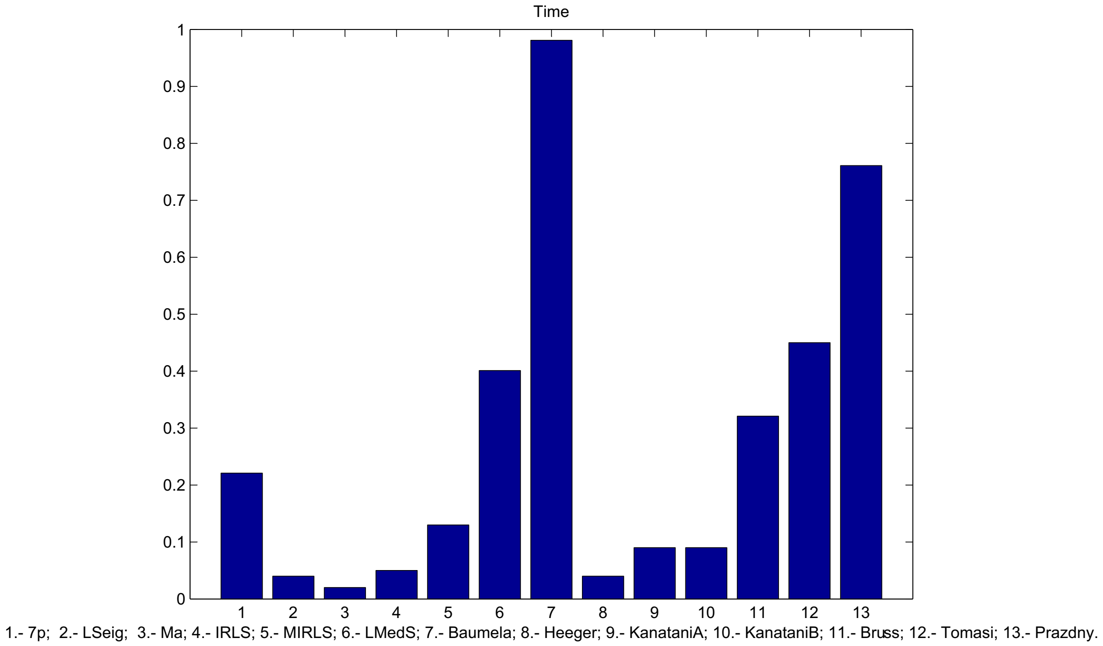


Figure 6.13: Computing time results of general methods with synthetic dates.

(LMedS) due to the lack of convergence in minimization given by MIRLS. Our implementation of LMedS optimizes the solution, once the outliers have been removed, by using MIRLS minimization, showing a poor estimate of the camera movement. The other surveyed methods obtain similar results, especially in the estimation of the translation movement, where an angle error of approximately 20° is obtained with a gaussian noise of only 0.3 pixels. Note that we are comparing these methods in the most unfavorable conditions. These conditions can be improved by reducing the image field of view of the camera or increasing the linear/angular velocity coefficient. Nevertheless, the results indicate that in different conditions the results obtained by the surveyed methods are comparatively similar. However, Figure 6.12 shows that the method proposed by Kanatani with a previous data normalization obtains by far the best results in angular velocity estimation even with increasing gaussian noise. Finally, comparison of the execution time of every method in MATLAB$^{®}$ running on a Pentium$^{®}$ III Computer at 800 MHz. is shown in Figure 6.13. The Figure shows that while almost half of the methods obtain a solution in less than 0.1 sec.; eleven, such as the 7-points method, Bruss and Horn, and Tomasi and Shi, obtain a motion estimation in less than 0.5 sec., and the two others, Prazdny, and Baumela,

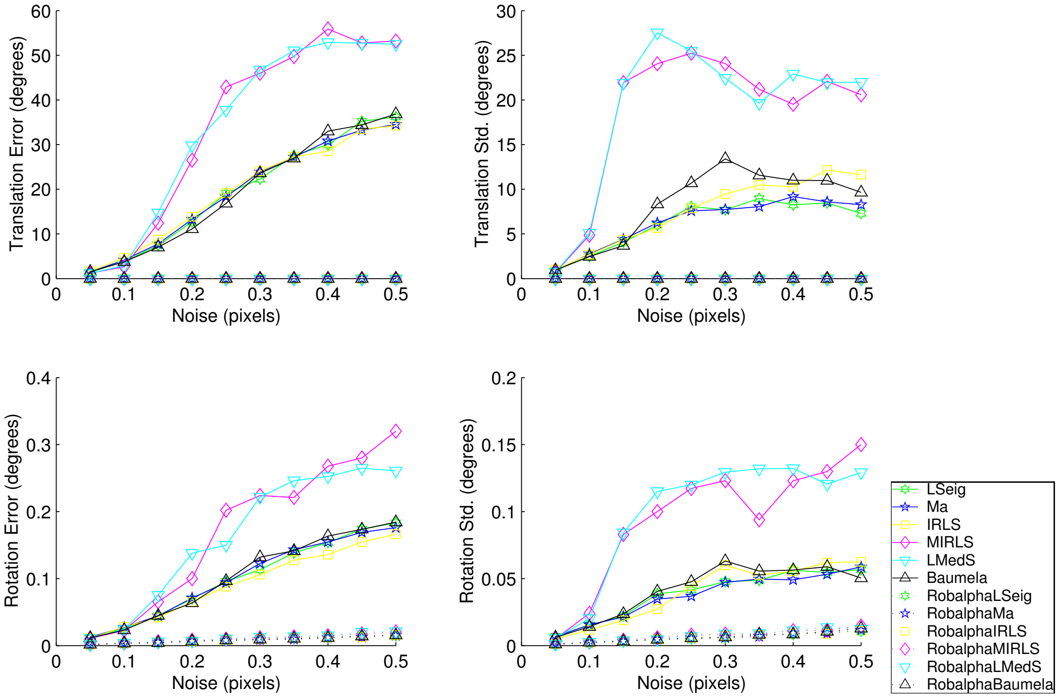


Figure 6.14: Estimation results of general and simplified methods with synthetic images.

Agapito, Bustos and Reid spend more than 0.5 sec.

6.4.2 Results on Mobile Robot Motion Estimation

In order to compare the methods adapted to estimate mobile robot movement, tests are based on the same settings used in the previous section but constrained to the common case of a mobile robot. A new parameter α, corresponding to the angle between the optical axis of the camera and the ground plane has been considered. Tests were done for several values of α, that is: 0^o, 15^o, 30^o, 45^o, 60^o, 75^o and 90^o. It has been observed that the motion estimation presents a slight error at $\alpha = 45^o$, which was the worst case.

Figure 6.14 shows the results obtained by the movement estimation methods for the case of a mobile robot with a field of view of 90^o, a coefficient linear/angular velocity equal to unity considering the worst case $\alpha = 45^o$.

The methods adapted to robot motion do not present an error in the linear velocity estimation because the methods intrinsically fix its direction. Hence, the error

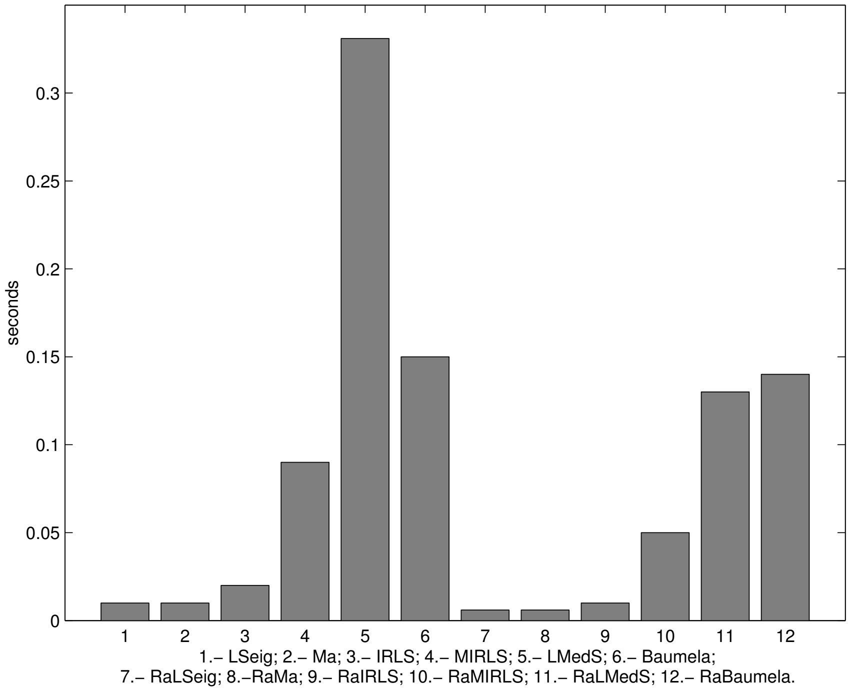


Figure 6.15: Computing time results of general and simplified methods with synthetic images.

in the translation estimate for every method shown in Figure 6.14 is zero. Actually, this fact implies that the error presented in the estimation of the angular velocity decreases considerably. Results on rotation estimation show that the adapted methods are more robust in the presence of image noise than their general versions (i.e. including all the 6-DOF). Figure 6.15 shows the computation times obtained by using MATLAB® and a PC Pentium® III at 800 MHz, showing that seven of the twelve methods yield an estimate in 0.05 seconds or less, permitting their use in real-time applications.

6.4.3 Results on Mobile Robot Motion with Real Images

The results obtained with real images are also quite accurate. Figures 6.18, 6.20 and 6.22 compare the results given by LS and its adaptation to the mobile robot (RobalphaLSeig), considering up to 80 test images where the camera has a tilt angle of 0°, 10° and 20° respectively and the robot progresses and rotates with an angle of -0.1° in every two consecutive images. The Figure shows the accuracy

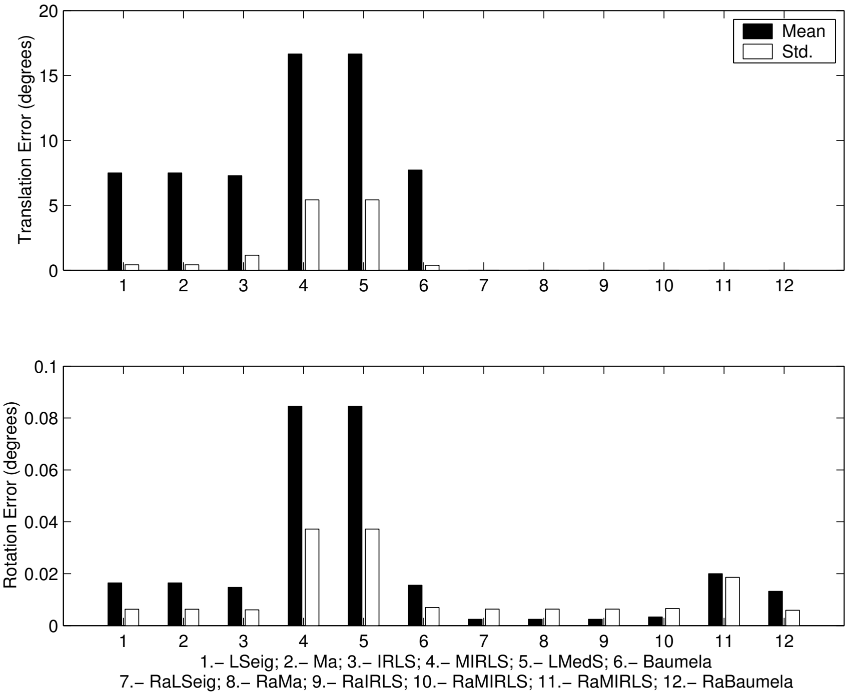


Figure 6.16: Results of general (1 through 6) and adapted (7 through 12) methods with real images.

on the estimation of rotation and translation estimation and the obtained vectors. The error on the estimation of translation is zero in the adapted method while the general method (6-DOF estimation) has an error in the Z-axis, since, in that case, it is difficult to distinguish between a camera rotation around Y-axis and a camera translation along X-axis. The error increases for α equal to 45^o being minimum at 0^o and 90^o. Finally, the rotation estimation is also more accurate using the adapted RaLS than that of the general method.

The same example described in the previous paragraph was tested for all the surveyed methods. The results obtained are presented in Figure 6.16, which shows that the adapted 2-DOF methods are always more accurate than the general 6-DOF.

6.4.4 Discussion

Traditionally, the estimation of the movement of a single camera is based on the application of epipolar geometry between every two consecutive images. However,

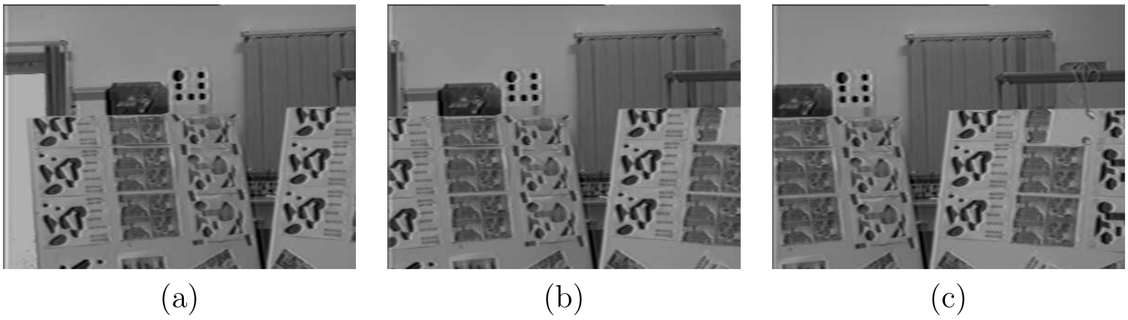

(a) (b) (c)

Figure 6.17: Image sequence with $\alpha = 0^o$. (a) 1st frame; (b) 40th frame; (c) 80th frame.

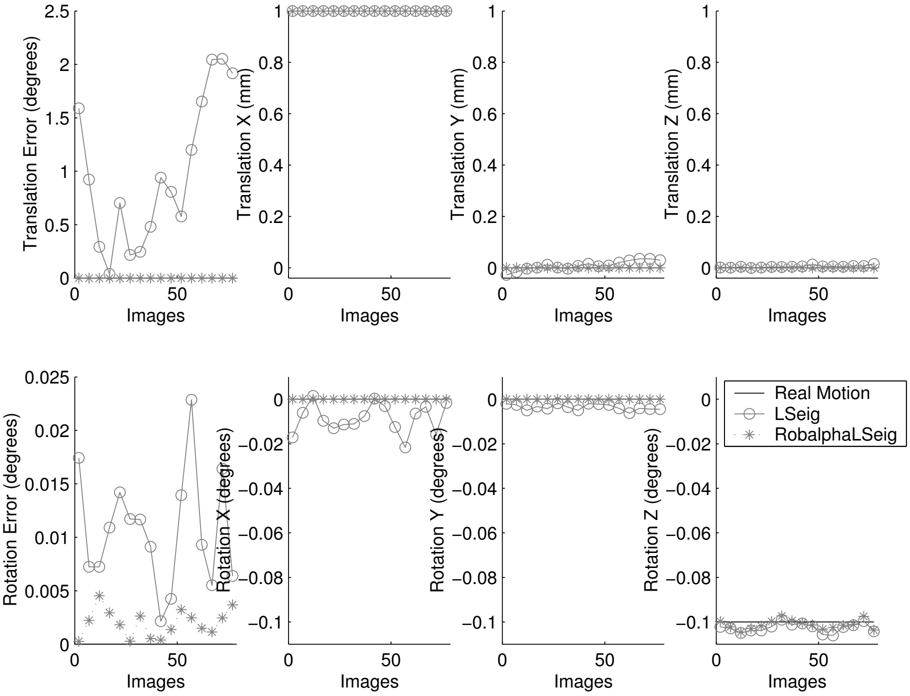
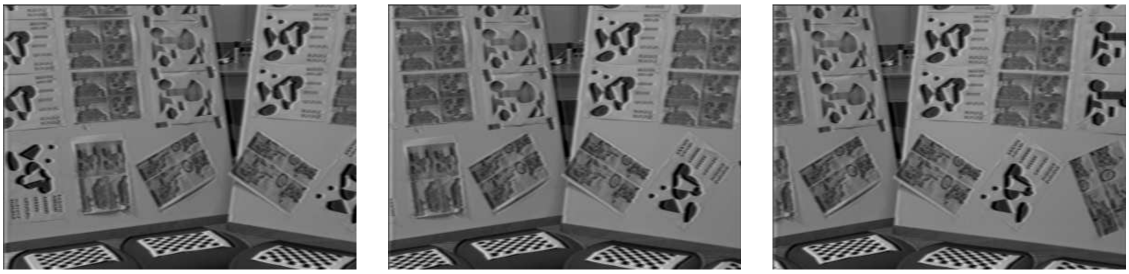


Figure 6.18: Example of motion estimation with real images of 6-DOF (LSeig) and its adaptation to 2-DOF (RobalphaLSeig) with $\alpha = 0^o$.



(a) (b) (c)

Figure 6.19: Image sequence with $\alpha = 10^o$. (a) 1st frame; (b) 40th frame; (c) 80th frame.

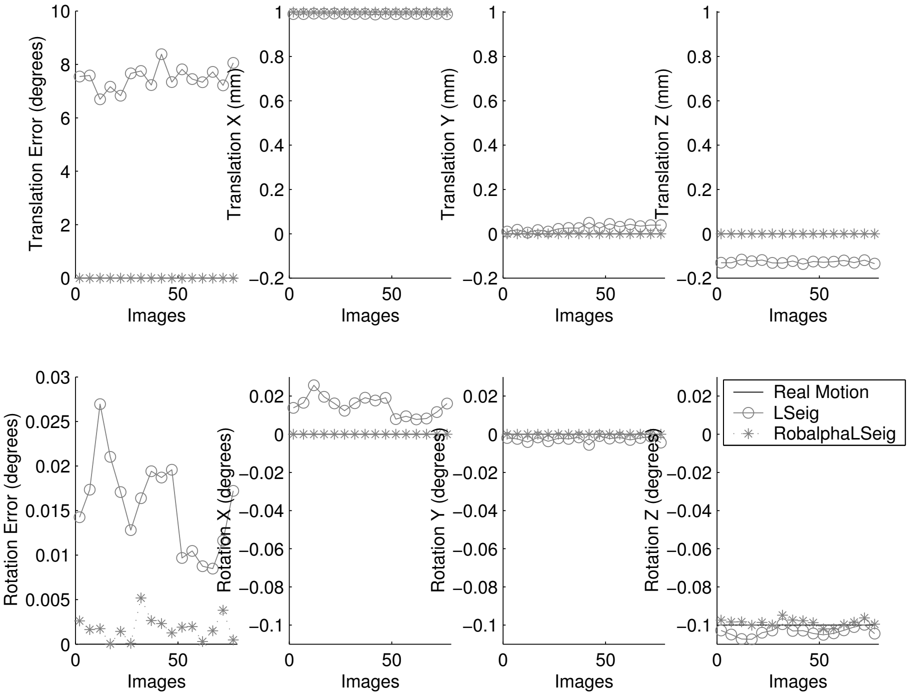


Figure 6.20: Example of motion estimation with real images of 6-DOF (LSeig) and its adaptation to 2-DOF (RobalphaLSeig) with $\alpha = 10^o$.


(a) (b) (c)

Figure 6.21: Image sequence with $\alpha = 20^o$. (a) 1st frame; (b) 40th frame; (c) 80th frame.

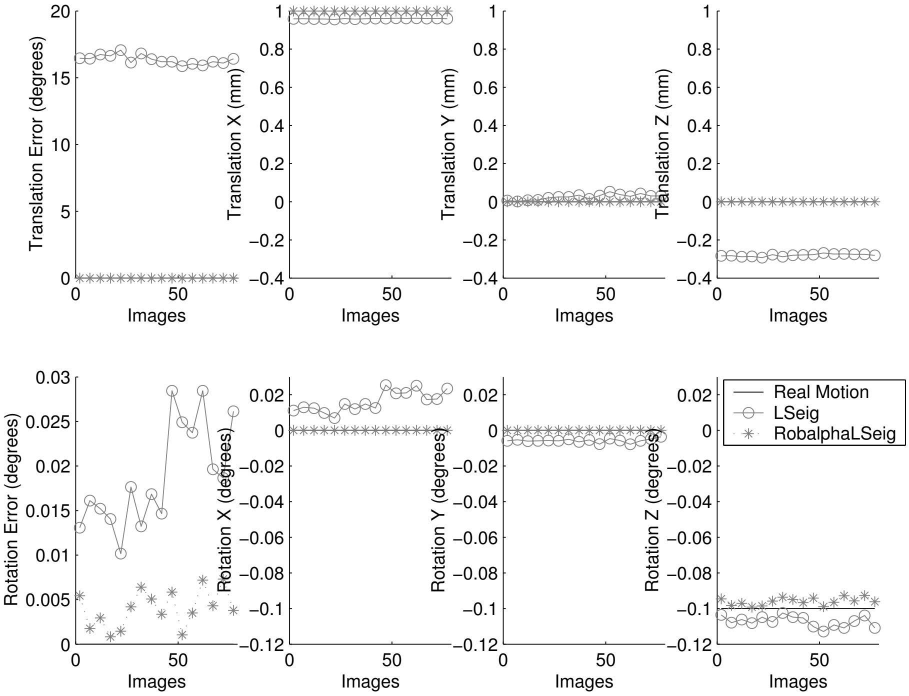


Figure 6.22: Example of motion estimation with real images of 6-DOF (LSeig) and its adaptation to 2-DOF (RobalphaLSeig) with $\alpha = 20^o$.

the traditional case of epipolar geometry has some limitations in the common case of a single camera attached to a mobile robot. First of all, the disparities between two consecutive images are rather small at the usual image rates, leading to numerical inaccuracies on the computation of the fundamental matrix. Indeed, disparities could be increased by removing intermediate images, but then the movement obtained is only a rough approximation of reality. Secondly, the correspondence problem between images is quite time consuming, reducing its effectiveness for real time applications.

Some authors have proposed techniques to estimate the camera motion at a high image rate based on the computation of the linear and angular velocities instead of the orientation and translation between consecutive images. These techniques are commonly based on the optical flow and the differential epipolar constraint. This section presents an exhaustive comparison of these techniques. It shows the robustness of each technique with respect to both synthetic and real images.

Nevertheless, all these techniques generate poor results unless they are adapted to the case of a mobile robot moving on a plane. The adaptation is based on constraining the movement from six degrees of freedom to only two degrees, which are: a) the spin rotation of the robot, and b) forward motion. Results highlight the robustness of the adapted methods. Moreover, motion is estimated faster, making its use easier at a high image rate.

In summary, the 6-DOF movement estimator methods are quite sensitive to noise. Hence, these methods should be adapted constraining the number of DOF with the aim of reducing error. In this section, the 2-DOF common case of a mobile robot has been considered and results show better movement estimation and stability due to the fixed direction of the translation movement constrained by the structure of the robot.

6.5 Mobile Robot Localization and Mapping Experimental Results

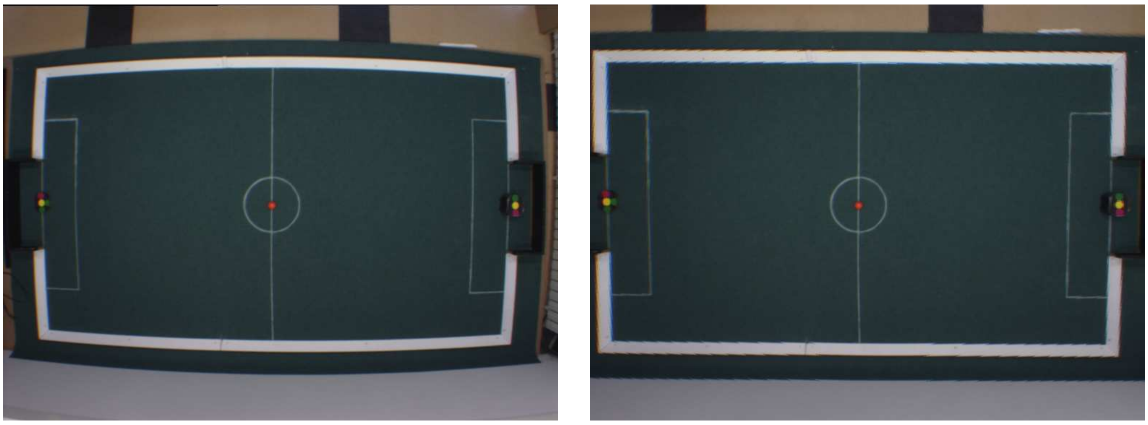
This section deals with the experimental results obtained by the mobile robot application described in chapter 5 and is structured in the following way. First, section 6.5.1 shows the 2D image processing results given by the following steps: a) lens distortion removal, b) corner detection; c) spatial cross correlation; and d) temporal cross correlation. Then, section 6.5.2 presents the results concerning 3D image processing; a) stereo reconstruction; b) object points tracking; and c) outlier rejection. Finally, section 6.5.3 shows some samples of reconstructed maps of the mobile robot surroundings obtained from a sequence of stereo images. The estimation of the robot trajectory is also detailed.

6.5.1 2D Image Processing Results

The 2D image processing tasks (described in section 5.3.2) are the following:

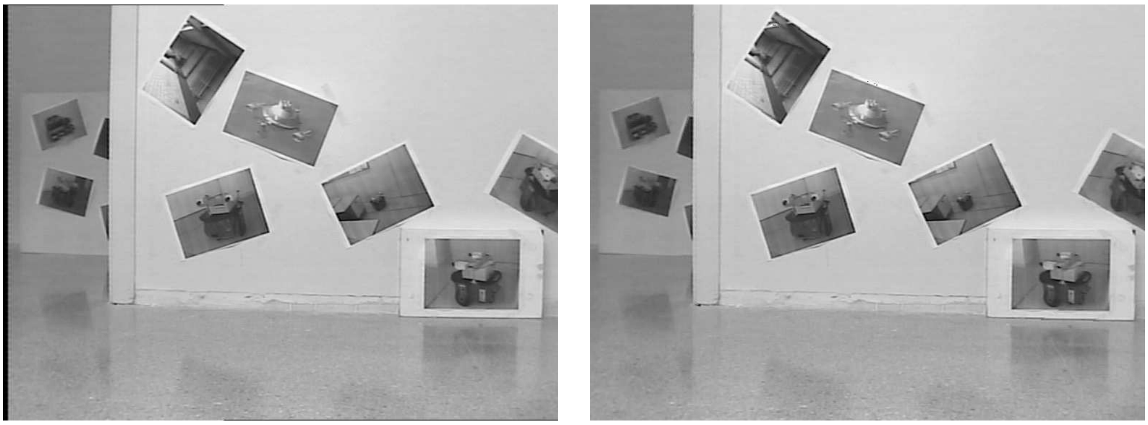
- Conversion of the color images to intensity images.
- Removal of the image distortion.
- Corner detection.
- Spatial cross correlation.
- Temporal cross correlation.

First, the color images are transformed to an intensity image with 256 different grey levels. This is a trivial task because it simply needs on computing, per every pixel, the mean value obtained from the Red, Green and Blue component. Then, the image distortion caused by the lenses is removed by means of using the camera calibration parameters. Figure 6.23 shows an example of image distortion removal, in which a 3 mm. lens camera was used to obtain a wide viewing field consequently causing great image distortion. As can be seen, the image on the left shows image distortion in which straight lines become curves, while in the image on the right, the distortion has been removed, recovering the straight lines.



(a) (b)

Figure 6.23: A sample of image distortion: a) the captured image with lens distortion; b) the processed image without lens distortion.



(a) (b)

Figure 6.24: A sample of image distortion in the case of a mobile robot: a) the captured image with lens distortion; b) the processed image without lens distortion.

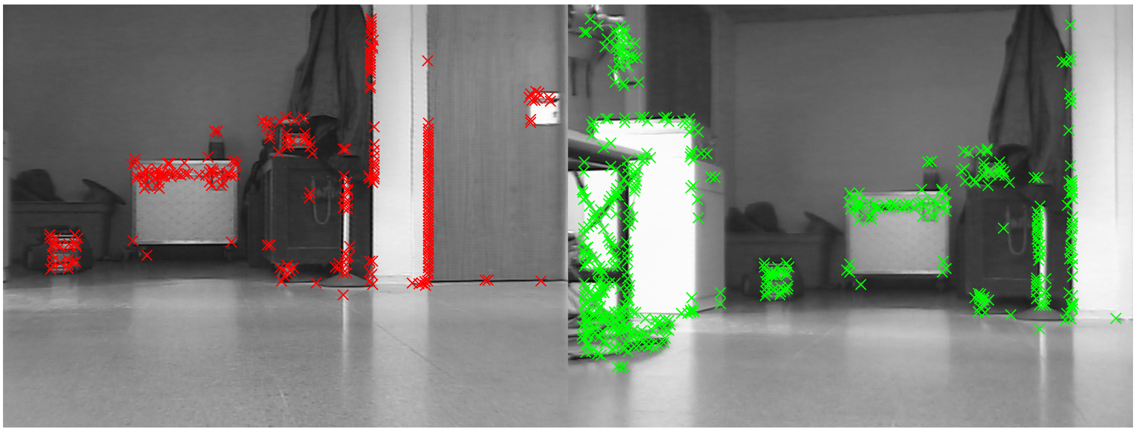


Figure 6.25: Stereo image example of corner detection in a typical indoor scene.

Now the image distortion removal is applied to the images obtained by the robot. Figure 6.24 shows the image captured by one of the two cameras of the stereo vision system and the image obtained after removing the distortion caused by lens. In this example, the effect of lens distortion is smaller because the focal length has been fixed to 6 mm. Although it is rather difficult see any difference between the two images, a closer study will show that the lines in the image on the left are slightly curved while in the image on the right, they are straight. The camera calibration method utilized is the method proposed by Tsai. However, the results given by using the non-linear method of Faugeras and Weng are basically the same (see accuracy evaluation in Table 6.7 and Table 6.8), indicating that any of the three methods is a good choice.

The third task of 2D image processing concerns the detection of corners as points of interest. The idea is to find as many corners as possible with the aim of acquiring as many potential interest points as possible in this phase. Subsequent tasks concern the solving of the matching problem and concentrate on discarding the corners which are of no interest. Figure 6.25 shows a stereo image acquired in the laboratory, showing the corners detected on all the various different objects in the scene. Figure 6.26 shows the results obtained by corner detection when applied to both images of the stereo vision system mounted on the mobile robot. Note that the corner detection has been carried out after removing lens distortion.

The fourth task deals with the spatially 2D image processing cross correlation between the two views provided by the stereo camera. Figure 6.27 shows an example

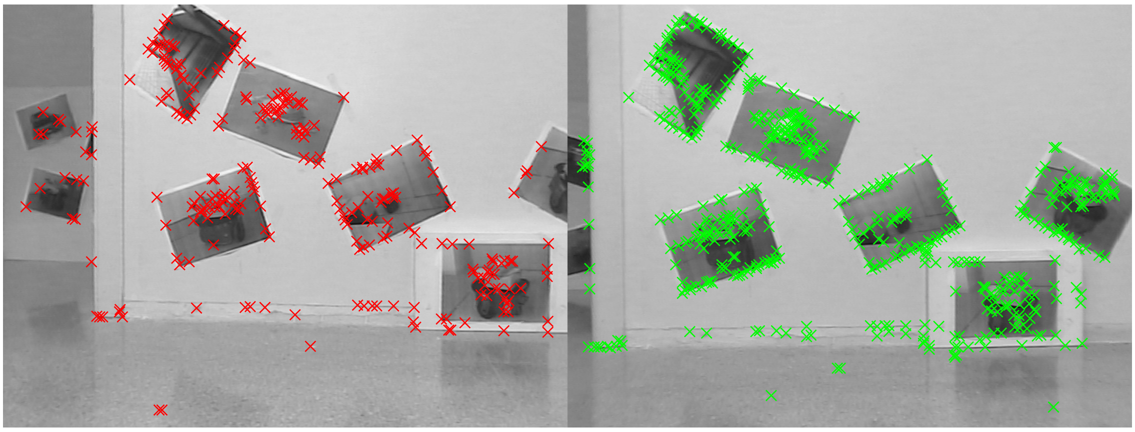


Figure 6.26: Stereo image example of corner detector in a controlled environment.

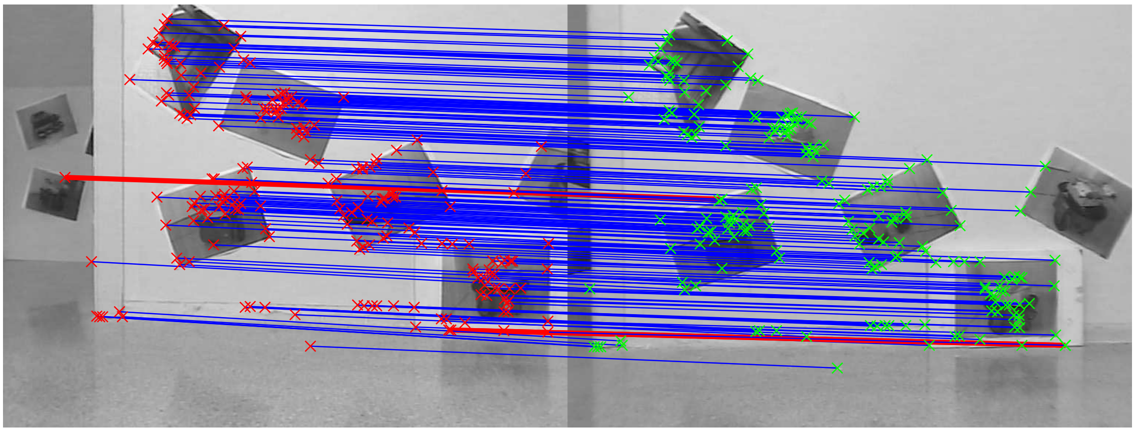


Figure 6.27: Example of spatial cross correlation in a pure translation movement.

of spatial cross correlation using the corners shown in Figure 6.26. In this case, the geometric relationship between both cameras is known from the calibration (see section 5.3.1). Given an image point in the first image, the correspondence is searched along its epipolar line in the second image. All the miscorrelated corners are discarded. Although most of the image corners are correctly correlated, the system was not able to provide any outliers. In this case, two outliers have been marked (red lines) in Figure 6.27. This sort of outlier is not detectable by using the epipolar geometry alone because the bad correspondence is located along the epipolar line as well. Figure 6.28 is a second example of spatial cross correlation. This example was obtained from an image sequence while the robot was executing a

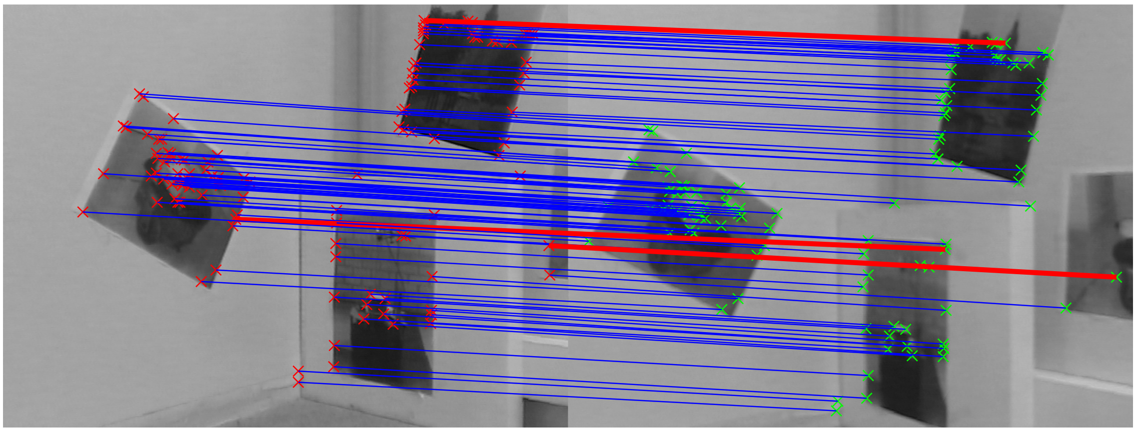


Figure 6.28: Example of spatial cross correlation in a pure rotation movement.

pure rotation. Note that in this case, there are three outliers marked with red lines which should be detected and rejected in subsequent steps.

Finally, the fifth and last task concerning 2D image processing is based on a temporal cross correlation. Consider the points of interest provided by the spatial cross correlation task applied to a stereo frame (see Figure 6.27) at the present and last instant of time. Figure 6.29 shows the results obtained by the temporal cross correlation applied to two consecutive stereo images. In this case, the robot is executing a pure translation and there are no outliers. In general, pure translations do not provide many outliers because consecutive frames are quite similar. Although the origin of displacement of both cameras is the same, both temporal cross correlations are independent because of the different relative displacement of each cameras with respect to the world, especially when the robot is turning. Therefore, when the robot is performing a pure rotation, the situation changes, leading to the fact that two consecutive frames may differ considerably. Figure 6.30 shows an example of temporal cross correlation when the robot is performing a pure rotation, in which the right camera comes closer to the scene while the left camera moves away from it. Note that in such a situation, the displacement of points on the image plane is larger, reducing the similarity between correspondences and leading to the presence of outliers (see Figure 6.29 where as many as three outliers are marked in red).

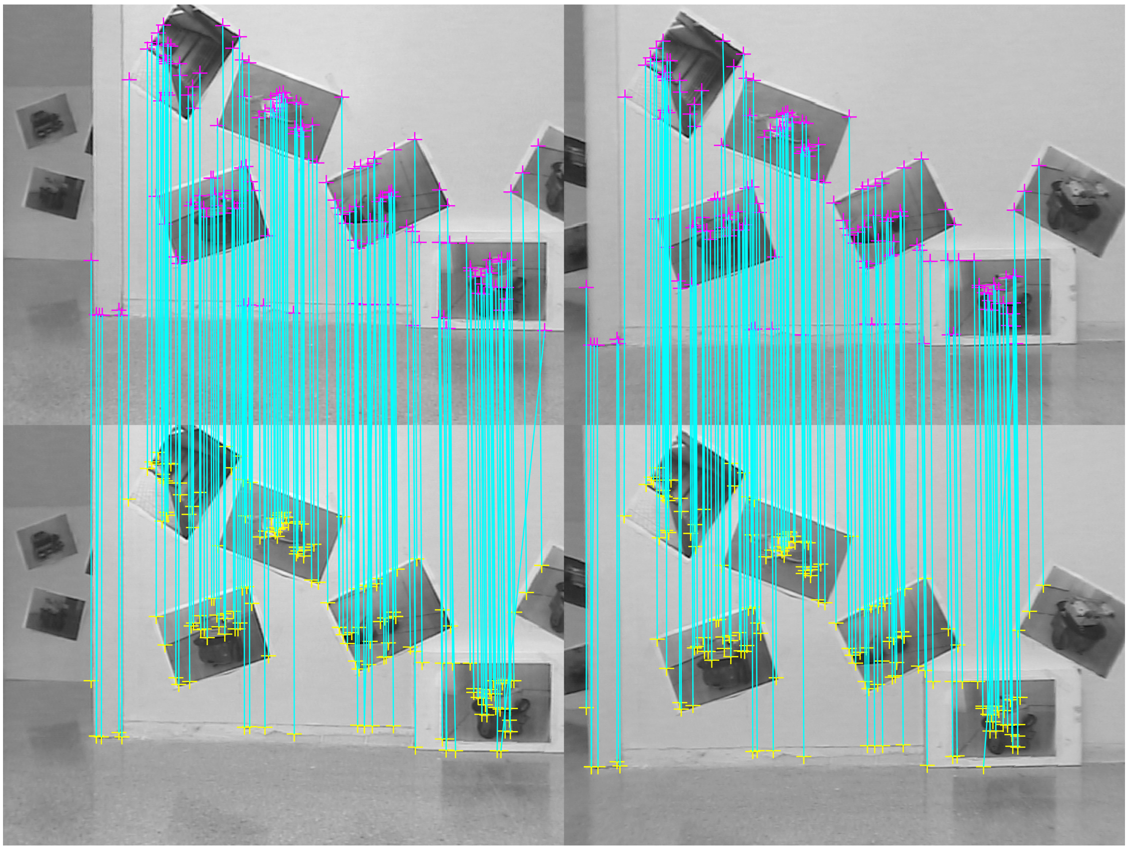


Figure 6.29: Example of temporal cross correlation in a pure translation movement.

6.5.2 3D Image Processing Results

The 3D image processing step involves the following three tasks:

- Stereo reconstruction.
- Tracking of 2D image points and 3D object points.
- Outlier rejection.

Given a set of image point correspondences, stereo reconstruction obtains a set of 3D object points. For instance, Figure 6.31 and Figure 6.32 are two different examples of stereo reconstruction in which four different views of the same reconstruction are shown. First, Figure 6.31 shows the reconstruction obtained from the image points correlated in Figure 6.27, in which blue crosses are the 3D object points, and

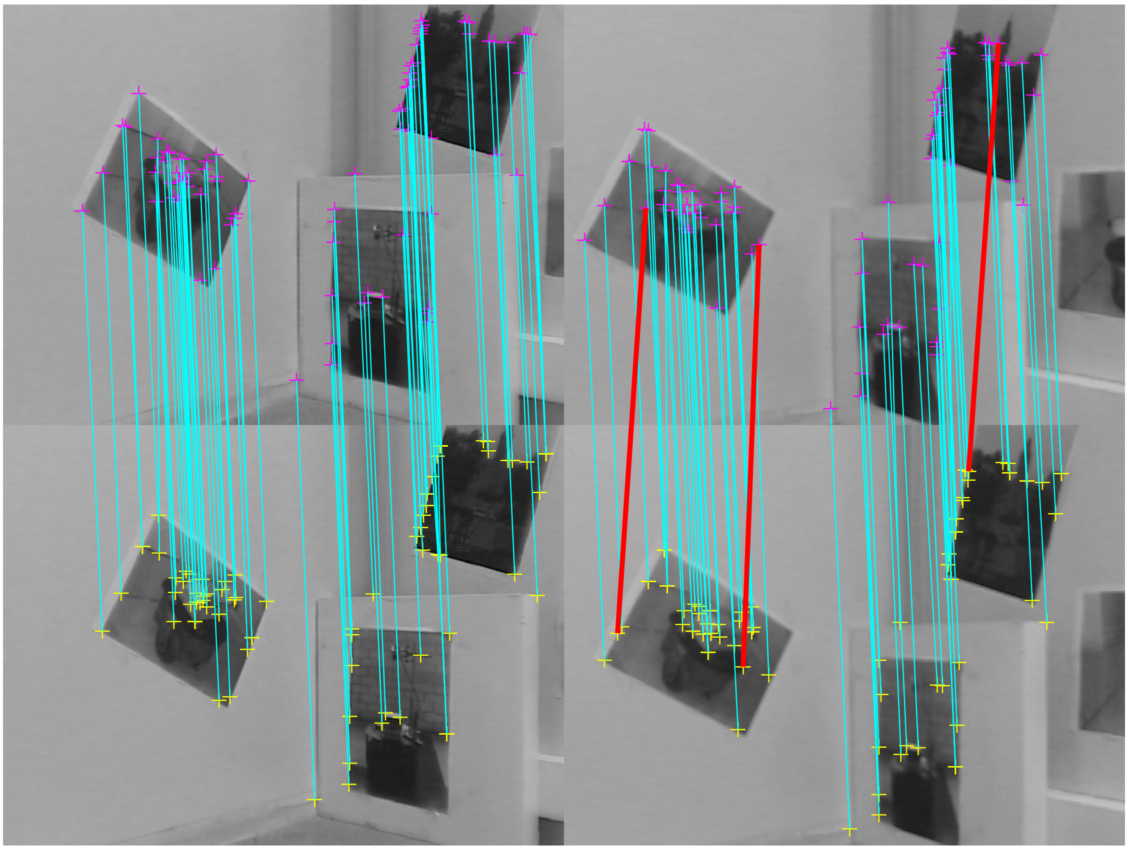


Figure 6.30: Example of temporal cross correlation in a pure rotation movement.

red and green crosses are the image points in the two cameras, respectively. Note that almost all the 3D object points (in blue) are aligned because they all come from the same wall in the scene. However, it can be also observed that some 3D object points are incorrectly reconstructed caused by false correspondences which must be further removed. Second, Figure 6.32 shows the reconstruction obtained from the image points correlated in Figure 6.28 where it can be seen, in the top view, that the points form two walls of the scene.

The second task deals with the tracking of image and object points divided into the storage of these 2D and 3D points and keeping track of these points on the image plane (see Figure 6.33) and in the 3D space (see Figure 6.34). This algorithm removes those points whose displacement on both image planes is inconsistent as a result of a wrong temporal cross correlation. All the incorrect 3D object points presented in Figure 6.31 are removed in Figure 6.34. Figure 6.35 and Figure 6.36

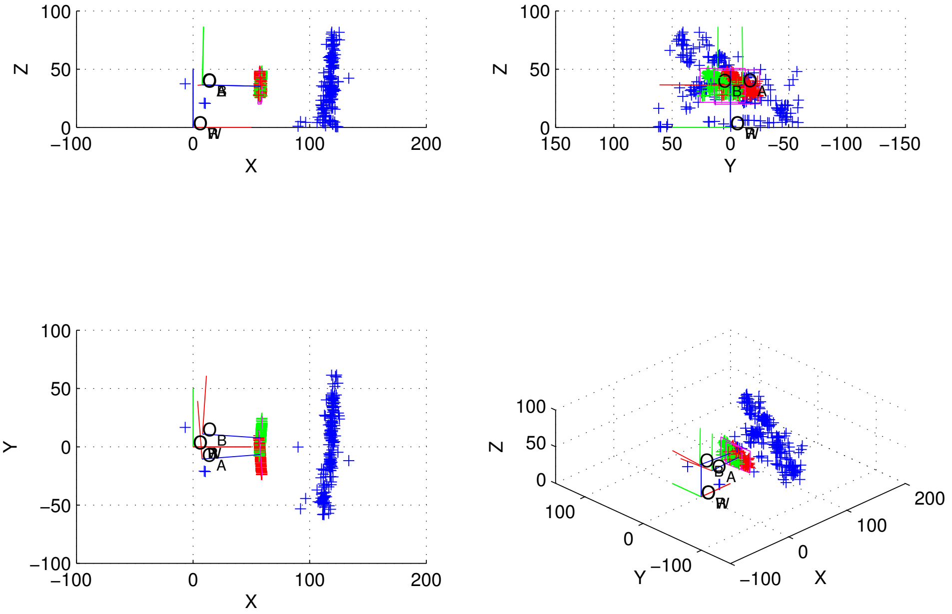


Figure 6.31: Stereo reconstruction example measured in centimeters and considering the mobile robot in front of a wall: a) lateral view; b) back view; c) top view; and d) 3D view.

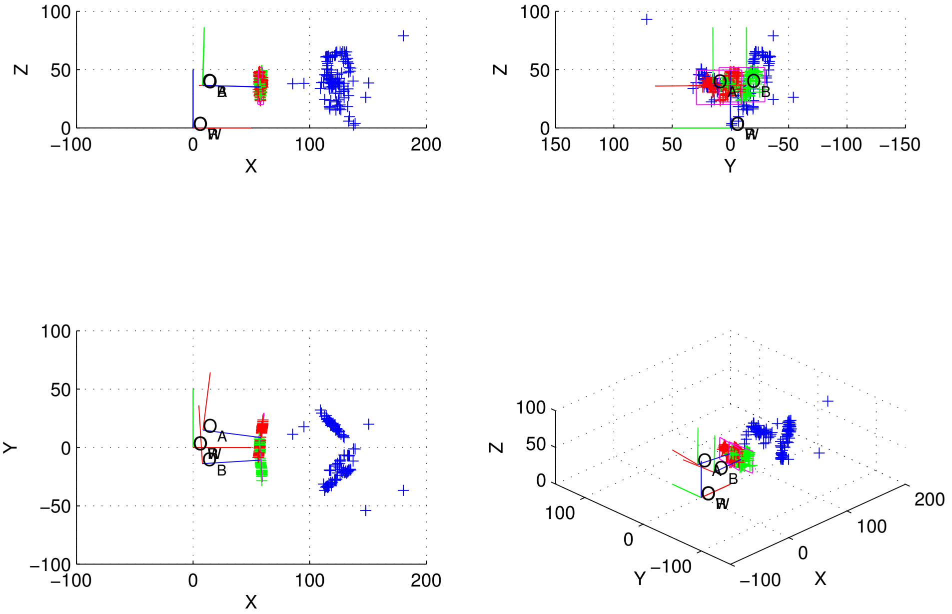


Figure 6.32: Stereo reconstruction example measured in centimeters and considering the mobile robot in front of two walls: a) lateral view; b) back view; c) top view; and d) 3D view.

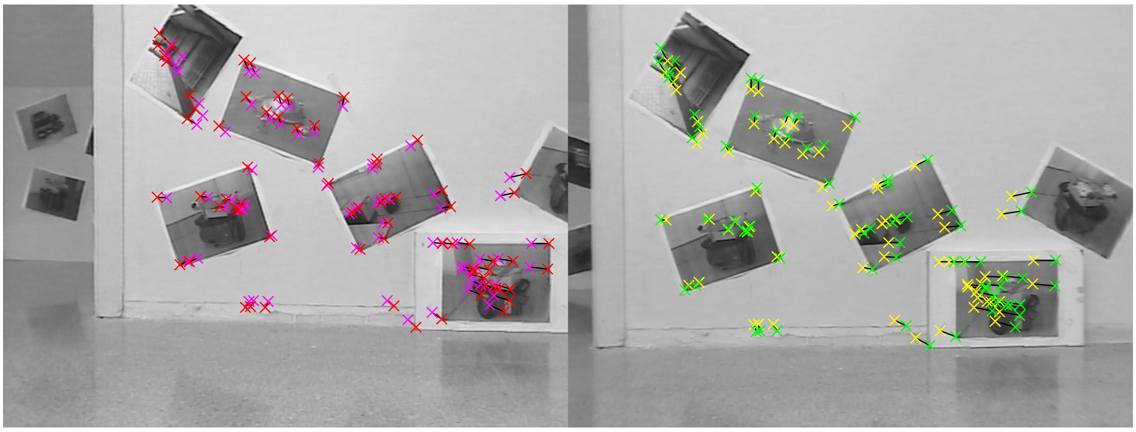


Figure 6.33: Tracking example in a pure translation movement throughout a sequence of five images provided by the stereo frame.

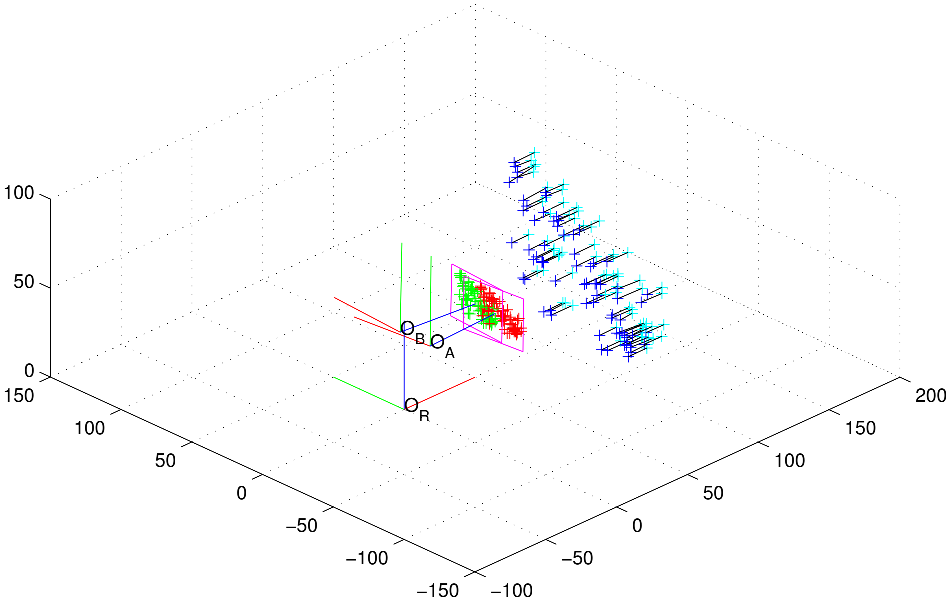


Figure 6.34: Tracking example in a pure translation movement throughout a sequence of five images in the 3D space.

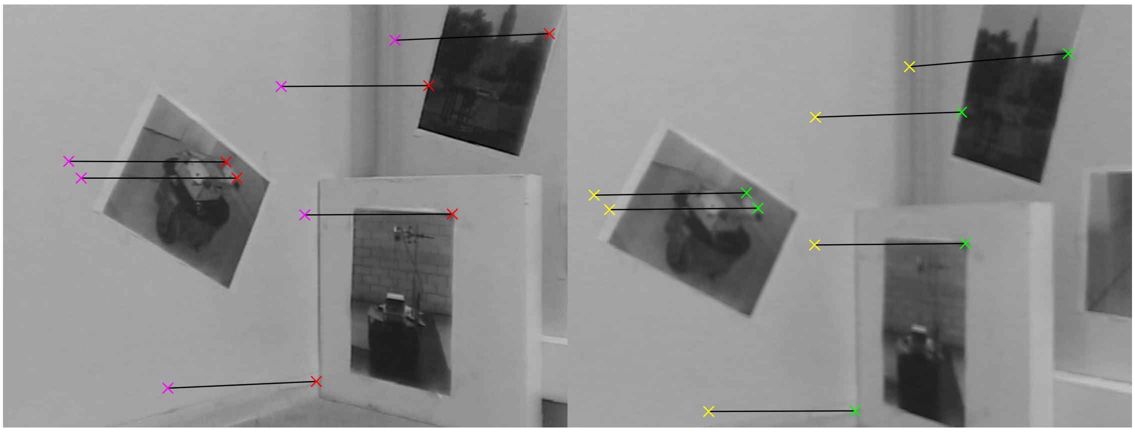


Figure 6.35: Tracking example in a pure rotation movement throughout a sequence of five images provided by the stereo frame.

show the tracking results when the robot is performing a pure rotation in the stereo frame and 3D space, respectively. The number of interest points has been reduced drastically because in a sequence of five images the point of view of both cameras has changed considerably and most of the corners are projected outside the camera field of view.

Although the tracking task considers the detection and removing of potential outliers, some of these outliers are not detected because the 2D movement remains similar. Figure 6.37 shows a stereo image in which an outlier has not been detected by the tracking because, although the point has been tracked correctly through the image sequence, the spatial cross correlation was erroneous. The first possibility we could think of was improving the spatial cross correlation, modifying the image processing by fine tuning the algorithm. However, this possibility leads to drastically reducing the set of interest points considered in the correlation. Besides, this sort of outlier can be detected easier by tracking the object points in 3D space. In Figure 6.38 it can be seen that most of the object points have a small displacement due to the robot motion while the outliers present a large displacement caused by the false matching.

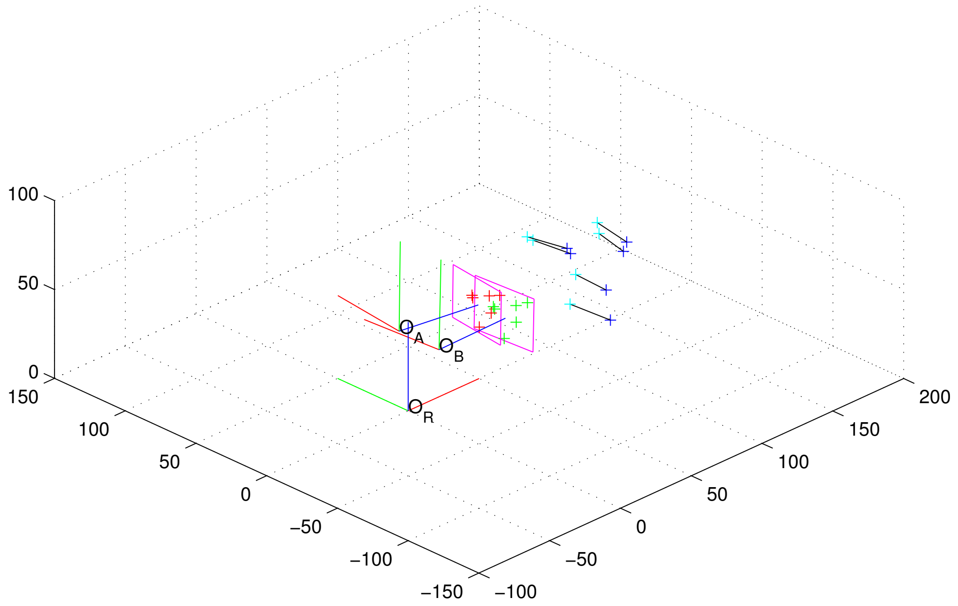


Figure 6.36: Tracking example in a pure rotation movement throughout a sequence of five images in the 3D space.

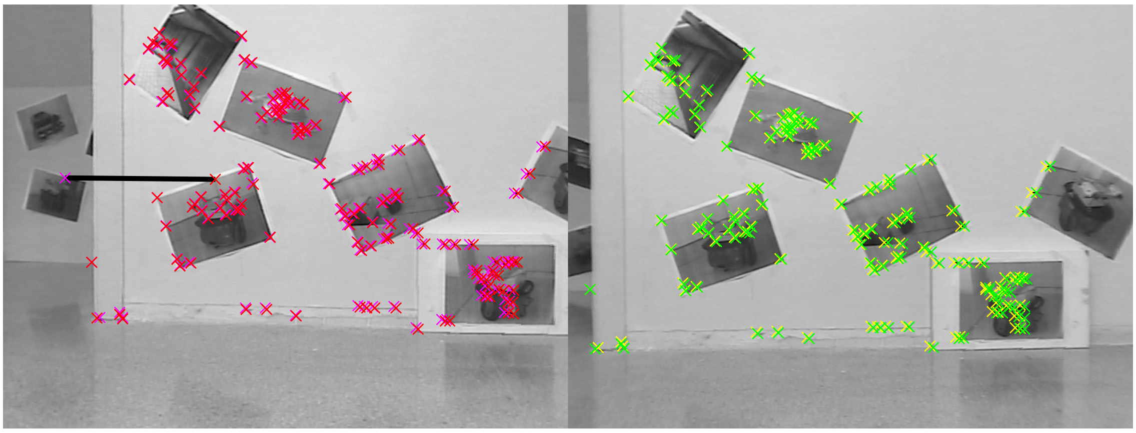


Figure 6.37: Stereo image with an outlier due to an erroneous image point correlation.

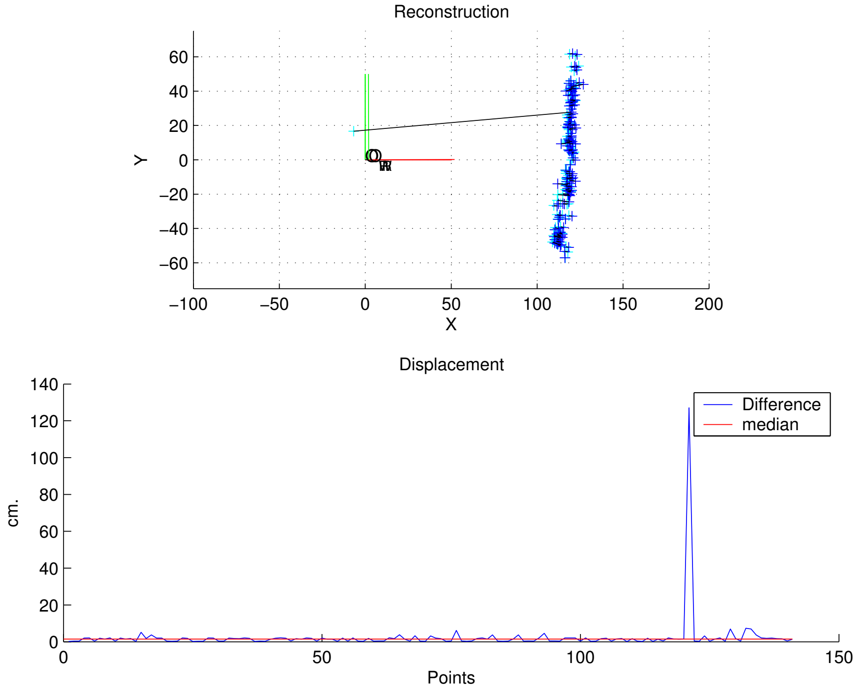


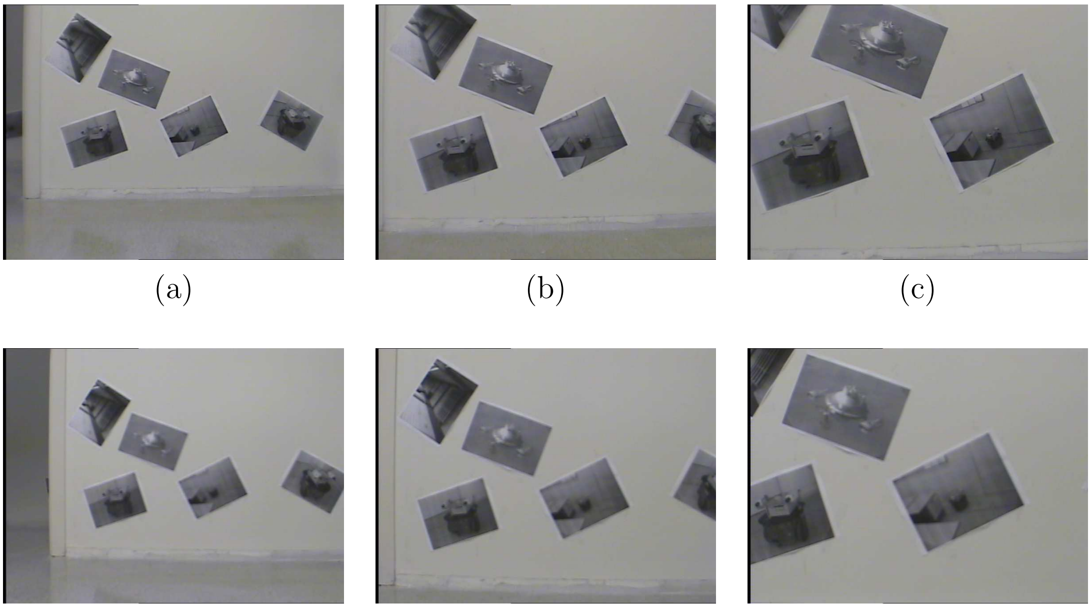
Figure 6.38: Reconstruction and displacement of 3D object points.

6.5.3 Localization and Map-Building Results

The experimental results obtained by using the mobile robot GRILL are described in the following paragraphs. First, an image sequence is presented in Figure 6.39 in which three representative image frames from a stereo image is shown. In this case, the robot is moving forward in a pure translation movement leading to: a) small displacement between every two consecutive images; b) the different objects present in the scene are maintained in the camera field of view, especially the objects placed around the image center. This kind of robot motion is the easiest to process because the static position of the objects in the stereo images simplifies the temporal cross correlation. Moreover, the tracking task can keep track of the points throughout several consecutive frames in the sequence.

Initially, the robot was placed at 160 cm. from a wall and was moved forward about 100 cm. taking an image every 2 cm. Figure 6.40 shows the reconstructed 3D map and the trajectory estimated.

A second image sequence is obtained considering that the robot is performing



(a) (b) (c)

(d) (e) (f)

Figure 6.39: Frames acquired by a translation image sequence: (a) 1st frame from the left camera; (b) 25th frame from the left camera; (c) 50th frame from the left camera; (d) 1st frame from the right camera; (e) 25th frame from the right camera; (f) 50th frame from the right camera.

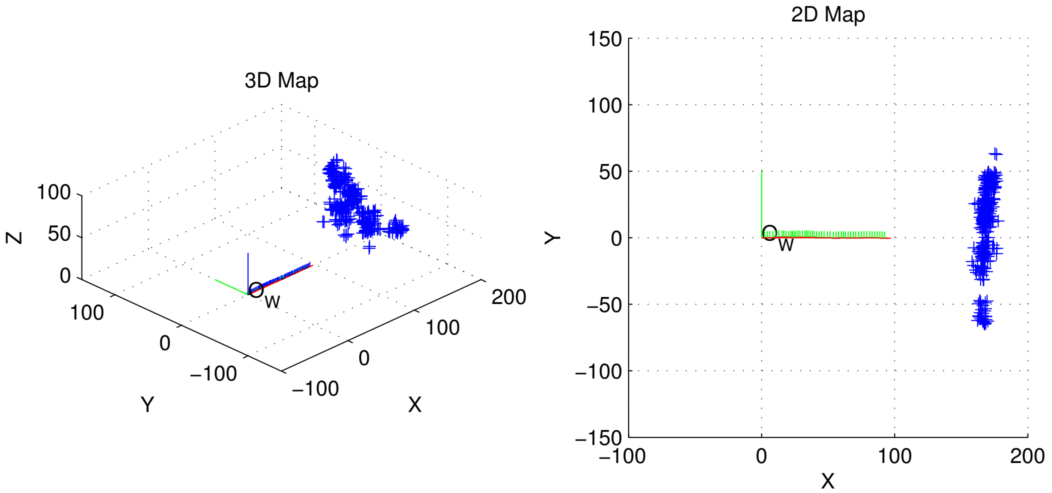
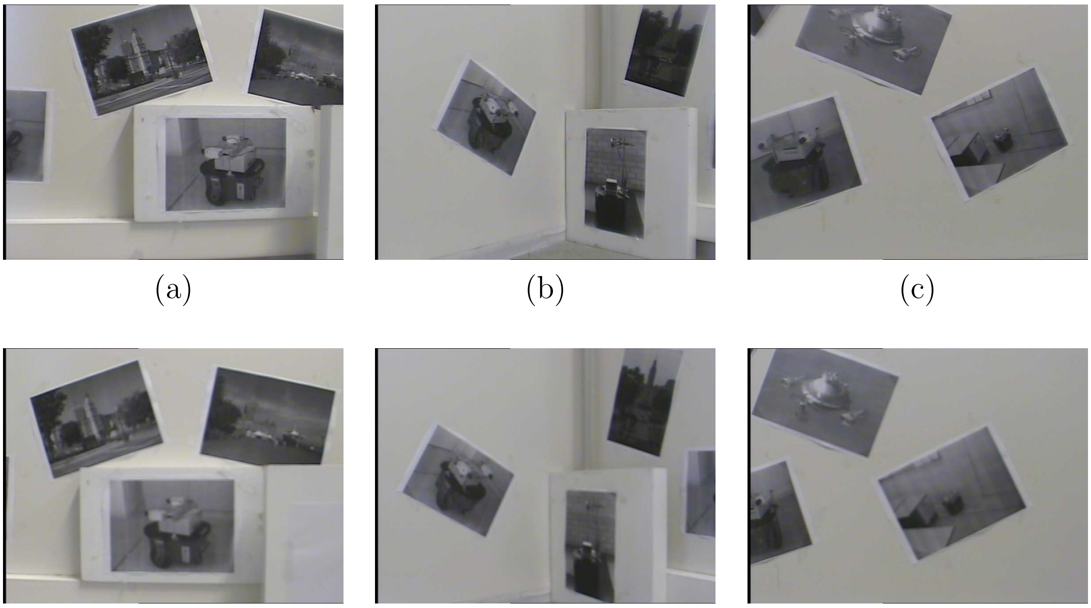


Figure 6.40: Localization and map building result in a pure translation sequence (units in centimeters).


(a) (b) (c)

(d) (e) (f)

Figure 6.41: Frames acquired by a rotation image sequence: (a) 1st frame from the left camera; (b) 45th frame from the left camera; (c) 90th frame from the left camera; (d) 1st frame from the right camera; (e) 45th frame from the right camera; (f) 90th frame from the right camera.

only a pure rotation movement (see Figure 6.41). In this case, the displacement of the objects on the image plane is relatively large compared to the previous example of a single translation. Additionally, objects vanish off of the image plane after just a few degrees of rotation. This fact makes temporal cross correlation considerably more difficult and, consequently, the task realized by the tracker. In this case, it is necessary to acquire the stereo frames at a high rate, especially if the rotational velocity is important.

First, the robot was placed at 100 cm. in front of a wall. A second wall was located at 100 cm. to its left side. The robot movement was a spin rotation around its center from right to left. The whole rotation was approximately 90 degrees, acquiring a stereo frame at every degree. Figure 6.42 shows the map built once the whole rotation was performed, in which it is easy to distinguish the corner formed by both walls and that only one outlier is present.

The last example is a sequence combined of rotational and translational movements in which a counterclockwise rotation of 45 degrees is applied to the robot

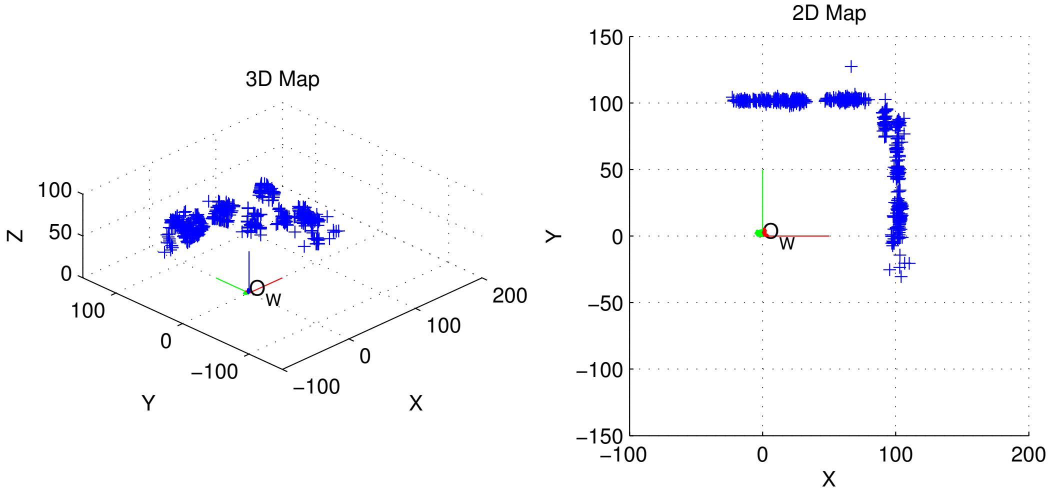


Figure 6.42: Localization and map building result in a pure rotation sequence (units in centimeters).

followed by a forward movement of 50 cm. Immediately following, a clockwise rotation of 135 degrees was performed followed by a forward movement of 100 cm. The mapping and localization of the robot is shown in Figure 6.43.

6.5.4 Discussion

A system to localize a mobile robot and reconstruct a map of the surroundings by means of stereo vision has been proposed. This stereo vision system is composed of two cameras mounted on a mobile robot. Several stereo image sequences have been acquired analyzing the partial results given by all the atomic tasks which compose the whole algorithm, i.e. lens distortion removal, corner detection, spatial and temporal cross correlation, tracking and so on. Also, some complete results of mobile robot localization and mapping in a structured surrounding are included.

The removal of lens distortion considerably improves the accuracy of the system and simplifies the further tasks of spatial cross correlation, stereo reconstruction and local localization. Actually, spatial cross correlation is affected by image distortion because the correlation is based on epipolar geometry. Then, the removal of lens distortion increases the accuracy in computing the epipolar lines. Also, stereo reconstruction computes the 3D position of imaged correspondences by means of

(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 6.43: Frames acquired by a combination of rotation and translation movements: (a) 1st frame; (b) 30th frame; (c) 50th frame; (d) 70th frame; (e) 90th frame; (f) 105th frame; (g) 120th frame; (h) 135th frame; (i) 150th frame; (j) 165th frame; (k) 180th frame; (l) 200th frame.

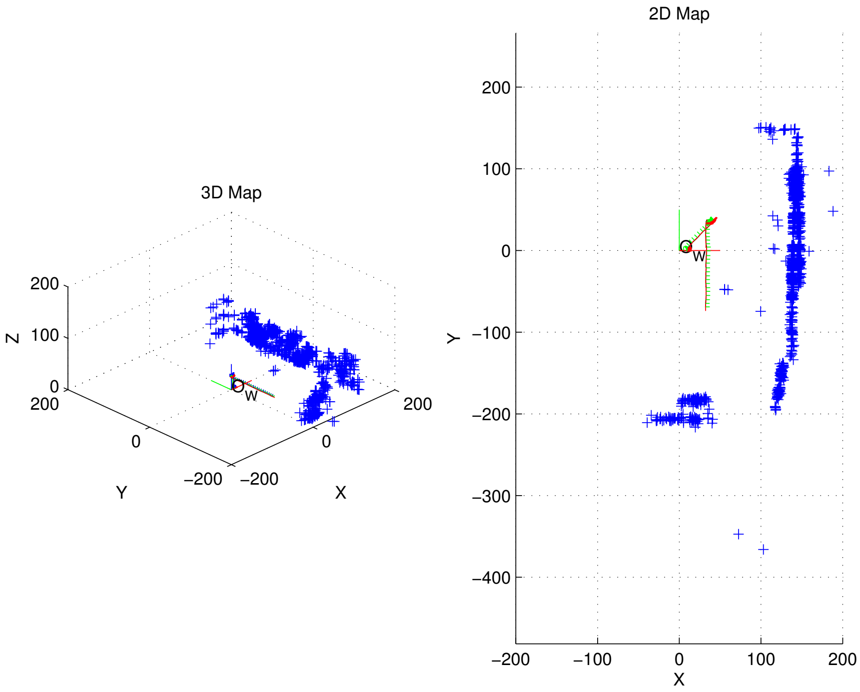


Figure 6.44: Localization and map building results in a combination of rotations and translations (units in centimeters).

triangulation, so that the undistorted image points are required. Then, the previous removal of lens distortion speeds up the reconstruction. Finally, local localization is based on comparing the 2D projection of the computed 3D points with respect to the ones acquired by the camera. Again, this comparison can be straight forward carried out if the lens distortion has been previously removed.

Points of interest have been obtained by means of a corner detector. Corners might not be the best way to find points of interest since it is more suitable to apply other techniques such as geometric invariants or regions. Nevertheless, corner detection is a fast technique and reliable enough for applications of mobile robot navigation. However, further research should be done in the computation of interest points. Better results might be obtained considering a detector based on geometric invariants which is theoretically more robust.

Spatial cross correlation considers the use of the epipolar geometry computed by calibration with the aim of reducing the number of potential correspondences, so that only the image points near the epipolar lines are considered as matching candidates.

The proposed method benefits from two advantages with respect to traditional cross correlation: the number of outliers is reduced, and the computation time decreases considerably.

Temporal cross correlation is not based on epipolar geometry because of the dependence of this geometry on the unknown movement of the mobile robot. Nevertheless, this unknown is not critical because the candidates have been previously filtered by the spatial cross correlation. Overall, pure translation movements are easier to process because of the similarity of every two consecutive stereo frames which considerably simplifies the image processing. On the other hand, rotational movements suffer from significant point displacements in the stereo frames which complicates cross correlation.

The tracking task is directly related to temporal cross correlation. The pursuit of image points through time permits distinguishing which interest object points are to be considered in map building. The points which are tracked through only a few sets of frames are not considered in the mapping. Also, the points that persist in several frames but are incorrectly correlated should be considered by the following task of outlier detection.

Outlier detection has the purpose of detecting inconsistencies in the 3D movement which drastically reduce the number of outliers. Nevertheless, the presence of a few outliers is negligible in mobile robot navigation. However, the number of outliers should be reduced to zero if a third camera is mounted in the stereo vision system. The crossing of both epipolar lines produced by two image points forming a correspondence might cross in the third image in a single point which might correlate with the other two correspondence points. This is a robust technique to reduce the number of potential outliers.

In summary, the proposed system follows one of the objectives of this thesis of obtaining a navigation system based entirely on computer vision. However, it is important to note that the application of data fusion techniques which takes advantage of other sensors such as sonar, encoders and gyroscopes will increase the reliability of the system. Overall, the proposed algorithm of localization and map building is robust enough for most applications in mobile robot navigation.

Chapter 7

Conclusions and Further Work

This chapter presents the conclusions of this work. Further work is also analyzed suggesting various ways through which the research may continue. Finally, this chapter lists the articles related to this thesis and other research contributions which have been published.

7.1 Conclusions

This thesis has focused on the study of modelling of stereo vision systems for robotic applications. Usually, the first step is the modelling and calibration of a single camera. To this end this thesis has presented a detailed survey covering a wide variety of pinhole camera models and their corresponding calibration methods. Then, the coupling of the two cameras which form the stereo vision system are related one to the other by the epipolar geometry using the fundamental matrix. In this way, this thesis contributes one of the most extended surveys on fundamental matrix estimation. The study of the motion estimation is also an important subject which has been considered. In this thesis, therefore, a third survey on egomotion estimation by means of the differential epipolar constraint is included. At this point, in depth knowledge of the geometry involved in fixed and moving cameras is brought in. Finally, this thesis amalgamates this knowledge and proposes an algorithm for mobile robot navigation considering vehicle localization and map building in unknown environments.

The principal contributions and conclusions of this work are summarized in the following paragraphs. Note that at the end of every chapter an extended summary is included and a discussion of the experimental results from the previous chapter is also detailed.

A comparative review of camera calibration methods, including accuracy evaluation, is presented. One of the main contributions in this case is the unification of the mathematical notation which allows the reader to easily follow the chapter through the different methods. In addition, once the desired calibrating method for a given application has been chosen, the reader can take the equations directly from this thesis or the software included. Besides, the survey concludes that the modelling of radial lens distortion is enough in most of the applications, so that the use of more complicated models does not improve accuracy significantly. Finally, the survey analyses the proposed methods to demonstrate the accuracy of the calibration method, concluding that the NSCE ratio is one of the most reliable.

Second, an overall view of fundamental matrix estimation is presented, in which the surveyed estimators have been classified. The main contribution in this aspect is the inclusion of a schema detailing the algorithmic viewpoint of each estimator. This is an important contribution which facilitates the comparison of the estimators. Moreover, the survey may be one of the most extended works on fundamental matrix estimation to date due to the number of methods studied. Chapter 3 concludes that robust methods are the most reliable because they can cope with both discrepancy in the localization of points as well as outliers. In addition, the orthogonal least-squares gives better results than the classic least-squares in function minimization. Moreover, corresponding points have to be normalized to improve minimization without ill-conditioning the convergence. Finally, a rank-2 matrix is preferred only to achieve good epipolar geometry, otherwise rank-3 matrices minimize the distance between points and epipolar lines more accurately.

Third, a review of egomotion by means of differential epipolar geometry applied to the motion of a mobile robot is detailed, including a state-of-the-art method based on differential epipolar geometry. Two main contributions should be highlighted in relation to this chapter. First, an extended classification is proposed, organizing the methods into linear versus non-linear against discrete versus differential. Furthermore, the main contribution deals with the adaptation of these methods to the common case of a mobile robot moving on a plane. This adaptation reduces the degrees of freedom, a minimization which increases the accuracy of the obtained results significantly and allows the estimation of motion at high image rates.

Finally, a mobile robot application of vehicle localization and mapping of unknown environments is proposed. A stereo system was mounted on a mobile robot and the proposed algorithm has been described in detail. This algorithm benefits from the knowledge described in the previous chapter whose principal main contributions are the following:

- Spatial cross correlation benefits from the knowledge of the epipolar geometry which reduces the computing time and the potential number of matching candidates.
- The potential outliers obtained in solving the correspondence problem can be completely removed before estimating vehicle localization. Thus, the candidates are followed through time on the image plane, considering only the points of interest which continuously appear throughout a sequence of images.
- Although a significant amount of outliers are eliminated in the previous step, the algorithm continues analyzing the consistency of the 3D motion. The 3D motion of each point in space is compared to the global motion of the system, detecting and removing those points which do not follow the overall motion.
- Vehicle localization is directly computed from the mapping acquired by the vision system without modifying the environment with landmarks and without having a previous map available. The map of the environment is updated dynamically while the robot is moving.

7.2 Further Work

The work presented in this thesis proposes the following subjects for further research:

- Completion of the camera calibration survey, detailing the last methods published in terms of accuracy and reliability. However, it should be considered that calibration can also obtain great benefits by using a different sort of image segmentation instead of interest points such as straight lines or geometric invariants.

- Further study of the fundamental matrix uncertainty to compute the accuracy instead of using the distance between points and epipolar lines. Also, the study of Kruppa equations increases the knowledge of the epipolar geometry in uncalibrated stereo.

- The use of a third camera in the vision system is a very easy way to remove the false matchings given by the cross correlation, even when considering epipolar lines. This can be done by using three fundamental matrices, but the study of the trifocal tensor which models the whole geometry more accurately should also be considered.

- Considering image segmentation, the use of corners as interest points is quite noisy and produces a significant amount of outliers. Therefore, the study of other image features such as segments, color and texture or even geometric invariants should be considered.

- The SLAM (Simultaneous Localization and Map Building) research subject should be studied with the aim of improving the proposed algorithm considering statistics and probability. Data fusion and the EKF (Extended Kalman Filter) should be included in order to improve the vehicle localization and the computation of its trajectory.

7.3 Related Publications and other Contributions

The articles published in international journals from the work of this thesis are the following:

- X. Armangué, H. Araújo and J. Salvi. A Review on Egomotion by Means of Differential Epipolar Geometry Applied to the Movement of a Mobile Robot. Accepted in April 2003 to be published in Pattern Recognition, **PR**.

- X. Armangué and J. Salvi. Overall View Regarding Fundamental Matrix Estimation. Image and Vision Computing, **IVC**, pp. 205-220, Vol. 21, Issue 2, Februay 2003.

- J. Salvi, X. Armangué and J. Batlle. A Comparative Review of Camera Calibrating Methods with Accuracy Evaluation. Pattern Recognition, **PR**, pp

1617-1635, Vol. 35, Issue 7, July 2002.

The articles published in national and international conferences are the following:

- X. Armangué, H. Araújo and J. Salvi. *Differential Epipolar Constraint in Mobile Robot Egomotion Estimation.* International Conference on Pattern Recognition, **ICPR 2002**, vol. 3, pp 599-602, Québec, Canada, August 2002.
- J. Salvi, X. Armangué, J. Pagès. *A Survey Addressing the Fundamental Matrix Estimation Problem.* IEEE International Conference on Image Processing, **ICIP 2001**, vol. 2, pp 209 -212 Thessaloniki, Greece, October 2001.
- J. Pagès, X. Armangué, J.Salvi, J. Freixenet, J. Martí. *A Computer Vision System for Autonomous Forklift Vehicles in Industrial Environments.* The 9th. Mediterranean Conference on Control and Automation, **MED 2001**, Dubrovnik, Croatia, June 2001.
- X. Armangué, J. Pagès, J. Salvi and J. Batlle. *Comparative Survey on Estimating the Fundamental Matrix.* IX Simposium Nacional de Reconocimiento de Formas y Análisis de Imágenes, **SNRFAI 2001**, pp 227-232, Castelló, Spain, May 2001.
- X. Armangué, J. Salvi and J. Batlle. *A Comparative Review of Camera Calibrating Methods with Accuracy Evaluation.* Proceedings of 5th Ibero-American Symposium on Pattern Recognition, **SIARP 2000**, pp 183-194, Lisboa, Portugal, 11-13 September 2000.
- X. Armangué, R. García, J. Batlle, X. Cufí, J.Ll. De La Rosa, P. Ridao. *Implementació d'un algorisme paral·lel de seguiment de múltiples microrobots mitjançant VHDL.* Proceedings del 2on Congrés Català d'intel·ligència Artificial, **CCIA 1999**, pp 356-364, Girona, Spain, 25-27 Octubre 1999.

Finally, the following applications of the programmed algorithms are also available at `http://eia.udg.es/~armangue/research`:

- Camera calibration toolbox in Matlab.
- Fundamental matrix toolbox in Matlab.

- Egomotion estimation toolbox in Matlab.
- User-friendly application of camera calibration for Windows (source code in C++).
- User-friendly application of fundamental matrix estimation application for Windows (source code in C++).
- Application of vehicle localization and mapping for Windows (source code in C++).

Appendix A

Notation

A.1 Mathematics Convention

This appendix synthesizes the nomenclature used to express coordinate systems and camera parameters in the research project.

$\{H\}$ defines a coordinate system H, which is composed of an origin O_H and either two $\{X_H, Y_H\}$ or three $\{X_H, Y_H, Z_H\}$ axis, depending on the number of dimensions defined.

The project defines the following coordinate systems:

- $\{W\}$=$\{O_W$,X_W,Y_W,$Z_W\}$ defines the world coordinate system.
- $\{C\}$=$\{O_C$,X_C,Y_C,$Z_C\}$ defines the camera coordinate system located at the focal point O_C.
- $\{R\}$=$\{O_R$,X_R,$Y_R\}$ defines the retinal coordinate system located at the principal point $O_R = (u_0, v_0)$.
- $\{I\}$=$\{O_I$,X_I,$Y_I\}$ defines the computer image coordinate system located in the upper-left corner of the image plane.

Each point P is always related to a coordinate system. Hence, ${}^HP$ relates the point P with respect to $\{H\}$, where ${}^HP = ({}^HX, {}^HY, {}^HZ)$. Each point can be related to any coordinate system. However, the following notations are the only ones used:

- ${}^{W}P_w$=(${}^{W}X_w$,${}^{W}Y_w$,${}^{W}Z_w$) expresses a 3D test point from the world (scene) expressed with respect to $\{W\}$.
- ${}^{C}P_w$=(${}^{C}X_w$,${}^{C}Y_w$,${}^{C}Z_w$) expresses a 3D test point from the world (scene) expressed with respect to $\{C\}$.
- ${}^{C}P_u$=(${}^{C}X_u$,${}^{C}Y_u$,f)=(${}^{C}X_u$,${}^{C}Y_u$) expresses the linear projection of a point ${}^{C}P_w$ on the image plane related to $\{C\}$, without including lens distortion.
- ${}^{C}P_d$=(${}^{C}X_d$,${}^{C}Y_d$,f)=(${}^{C}X_d$,${}^{C}Y_d$) expresses a 2D image point, including lens distortion, related to $\{C\}$.
- ${}^{I}P_d$=(${}^{I}X_d$,${}^{I}Y_d$) expresses a 2D image point related to the image coordinate system $\{I\}$, in pixels. This point is the observable point from image acquisition.

In order to distinguish a single point from a set, i.e. the set of test points, a second sub-index is used. Then, P_{ui} indicates the i-th point on a set, where $i = 1...n$.

A rigid transformation between a two coordinate system is expressed by a transformation matrix, i.e. ${}^{J}K_H$ expresses the coordinate system $\{H\}$ with respect to $\{J\}$. Moreover,

$$
{}^{J}K_H = \begin{pmatrix} {}^{J}R_H & {}^{J}T_H \\ 0_{1\times 3} & 1 \end{pmatrix}
$$

where $R = (r_1, r_2, r_3)^{\mathrm{T}}$ expresses the orientation of $\{H\}$ measured with respect to the axis of $\{J\}$. R can also be given related to the three rotation angles, i.e. α, β and γ. Moreover, $T = (t_x, t_y, t_z)^{\mathrm{T}}$ expresses the position of the origin of $\{H\}$ with respect to $\{J\}$.

Then, the following camera parameters are used:

- k_1 is the first coefficient of a series which models the radial lens distortion.
- g_1 up to g_4 are the coefficients which model the decentering and thin prism lens distortion.
- f is the focal distance, i.e the distance from the focal point O_C to the image plane.

- (u_0,v_0) are the two components of the principal point, i.e. the projection of O_C on the image plane.
- k_u, k_v are the two components which permit to transform a point from metric coordinates to pixels.
- α_u, α_v are defined as $\alpha_u = f\,k_u$ and $\alpha_v = f\,k_v$.
- s_x is the scale factor.
- $d'_x = d_x \frac{N_{cx}}{N_{fx}}$
- d_x, d_y are the center to center distances between adjacent sensor elements with respect to X direction and Y direction of the CCD sensor, respectively.
- N_{cx} is the number of sensor elements in the X direction of the CCD sensor.
- N_{fx} is the number of pixels in an image row as sampled by the computer.

Finally, nomenclature used in the differential epipolar constraint:

- Linear velocity: $v = (v_1, v_2, v_3)^{\mathrm{T}} \in \mathbb{R}^3$
- Angular velocity: $\omega = (\omega_1, \omega_2, \omega_3)^{\mathrm{T}} \in \mathbb{R}^3$
- Symmetric matrix: $\mathbf{S} = \begin{pmatrix} s_{11} & s_{12} & s_{13} \\ s_{12} & s_{22} & s_{23} \\ s_{13} & s_{23} & s_{33} \end{pmatrix} \in \mathbb{R}^{3\times3}$ where $\mathbf{S} = \mathbf{S}^{\mathrm{T}}$
- Skew symmetric matrix associated with $\boldsymbol{v}$: $\hat{\boldsymbol{v}} = \begin{pmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{pmatrix}$ where $v \times u = \hat{v}u$

Appendix B

Software

This appendix presents the software referenced throughout this document and utilized to obtain the experimental results. The software was implemented in Matlab$^{©}$ and in C++. All the code is available in `http://eia.udg.es/~armangue/research`.

B.1 Toolbox in Matlab$^{©}$

Matlab$^{©}$ was chosen to implement the different surveyed methods in this thesis. Note that Matlab$^{©}$ is an excellent environment to simulate proposed algorithms because of the available set of mathematical functions already implemented and the ease of adding new functions. Maltab$^{©}$ has become a standard language in developing and testing mathematical methods.

Three toolboxes were implemented: a camera calibration toolbox, a fundamental matrix toolbox and a motion estimation toolbox. All these toolboxes are described in the following sections.

B.1.1 Camera Calibration Toolbox

This toolbox implements the camera calibration methods presented in chapter 2 and utilized to obtain the experimental results presented in section 6.2. Table B.1 relates the list of files implemented in this toolbox which have been classified into two sublists. The first sublist details the camera calibration surveyed methods

(Hall, Faugeras, non-linear Faugeras, Tsai and Weng). The second sublist relates to the functions implemented which compute the accuracy of the calibration surveyed methods (3D error, NSCE coefficient, 2D error in the distorted plane and 2D error in the undistorted image plane).

Table B.1: File list of the camera calibration toolbox.

Filename	**Description**
`CalDemo.m`	Demonstration of the camera calibration toolbox.
`CalHall.m`	Calibrates the camera by using the Hall method.
`CalFaugeras.m`	Calibrates the camera by using the Faugeras method.
`CalFaugerasIter.m`	Calibrates the camera by using the Faugeras method including lens distortion.
`CalTsaiLM.m`	Calibrates the camera by using the Tsai method with an iterative optimization of Lenvenberg-Marquardt.
`CalTsaiNR.m`	Calibrates the camera by using the Tsai method with an iterative optimization of Newton-Raphson.
`CalWeng.m`	Calibrates the camera by using the Weng method.
`CalError3DHall.m`	Computes the 3D error given by the Hall method.
`CalError2DHall.m`	Computes the 2D error given by the Hall method.
`CalError3DFaugeras.m`	Computes the 3D error given by the Faugeras method.
`CalErrorNSCEFaugeras.m`	Computes the NSCE coefficient given by the Faugeras method.
`CalError2DFaugeras.m`	Computes the 2D error given by the Faugeras method.
`CalError3DFaugerasIter.m`	Computes the 3D error given by the Faugeras method including lens distortion.
`CalErrorNSCEFaugerasIter.m`	Computes the NSCE coefficient given by the Faugeras method including lens distortion.
`CalError2DdFaugerasIter.m`	Computes the 2D error given by the Faugeras method including lens distortion in the distorted image plane.
`CalError2DuFaugerasIter.m`	Computes the 2D error given by the Faugeras method including lens distortion in the undistorted image plane.
`CalError3DTsai.m`	Computes the 3D error given by the Tsai method.
`CalErrorNSCETsai.m`	Computes the NSCE coefficient given by the Tsai method.
`CalError2DdTsai.m`	Computes the 2D error given by the Tsai method in the distorted image plane.
`CalError2DuTsai.m`	Computes the 2D error given by the Tsai method in the undistorted image plane.
`CalError3DWeng.m`	Computes the 3D error given by the Weng method.
`CalErrorNSCEWeng.m`	Computes the NSCE coefficient given by the Weng method.
`CalError2DdWeng.m`	Computes the 2D error given by the Weng method in the distorted image plane.
`CalError2DuWeng.m`	Computes the 2D error given by the Weng method in the undistorted image plane.

B.1.2 Fundamental Matrix Estimation Toolbox

The methods implemented in this toolbox are described in section 3.4 and the results obtained analyzed in section 6.3. The FNS, CFNS, MLESAC and MAPSAC methods are not included in this toolbox because their implementations have been provided by the original authors (FNS, CFNS are available in
`http://www.cs.adelaide.edu.au/~hengel/Vision/Code/`
while MLESAC and MAPSAC are available in
`http://research.microsoft.com/~philtorr/`).

Table B.2 shows the file list in which the the different **F** estimators and the related functions described in section 3.4.5 are detailed.

Table B.2: File list of the fundamental matrix estimation toolbox.

Filename	**Description**
`FunMatDemo.m`	Demonstration of the fundamental matrix estimation toolbox.
`FunMat7p.m`	Estimates the fundamental matrix using the seven points method.
`FunMatLS.m`	Estimates the fundamental matrix using the least-squares technique.
`FunMatEig.m`	Estimates the fundamental matrix using the eigen analysis.
`FunMatFaugeras.m`	Estimates the fundamental matrix using the analytic method with rank-2 constraint.
`FunMatNR.m`	Estimates the fundamental matrix using the Newton-Raphson technique.
`FunMatinmmdpel.m`	Estimates the fundamental matrix using the Iterative Linear Method Minimizing Distances of Points to Epipolar Lines.
`FunMatilmmdpel.m`	Estimates the fundamental matrix using the Iterative Nonlinear Method Minimizing Distances of Points to Epipolar Lines.
`FunMatGradLS.m`	Estimates the fundamental matrix using the gradient method with the least-squares technique.
`FunMatGradEig.m`	Estimates the fundamental matrix using the gradient method with the eigen analysis.
`FunMatMEstLS.m`	Estimates the fundamental matrix using the M-Estimators with the least-squares technique.
`FunMatMEstEig.m`	Estimates the fundamental matrix using the M-Estimators with the eigen analysis.
`FunMatMEstTorr.m`	Estimates the fundamental matrix using the M-Estimators proposed by Torr.
`FunMatLMedSLS.m`	Estimates the fundamental matrix using the Least Median of Squares with the least-squares technique.
`FunMatLMedSEig.m`	Estimates the fundamental matrix using the Least Median of Squares with the eigen analysis.
`FunMatRANSAC.m`	Estimates the fundamental matrix using the RANndom SAmpling Consensus.
`NormalOneToOne.m`	Data normalization between $[-1, 1]$.
`NormalHartley.m`	Data normalization using the method proposed by Hartley.
`FunMatRank2.m`	Transforms a rank-3 3 matrix to a rank-2 matrix

B.1.3 Motion Estimation Toolbox

The last toolbox implemented concerns the motion estimation. The methods implemented are explained in chapter 4 and the results obtained compared in section 6.4.

Table B.3 presents the file list of this toolbox. The first sublist deals with the methods implemented based on the differential epipolar constraint. The second sublist consists of the files whose motion recovery is based on the optical flow. These methods are obtained from
`http://www.cns.nyu.edu/~david/publications.html#tian-cvpr96`
and adapted for comparison with those previously listed. The third sublist of files concerns the adaptation of 6-DOF general methods to the 2-DOF movement of a mobile robot moving on a planar surface.

B.2 Applications Implemented in C++

The use of Matlab© eases the developing of mathematical applications but suffers from two important drawbacks:

- Matlab© language is interpreted, consequently its execution is rather slow compared to compiled languages. This problem is well known and Matlab© developers provide different tools such as P-Funtions, MEX-Function and Matlab© compilers with the aim of decreasing the computing time execution. These tools are very useful in many cases but are not supported in all toolboxes (in particular the compiler).
- Some problems arise when building a self-executed application. First, it is interesting to implement a user-friendly application especially for non Matlab© programmers. Second, Matlab© does not support connectivity to external devices such as frame grabbers.

In order to overcome these drawbacks, camera calibration methods and fundamental matrix estimators have been implemented in C++ using the object oriented paradigm [Robson 00, Stroustrup 97]. This language improves the computing time execution considerably, building a user-friendly application. Moreover, an introduction to the application developed to localize the mobile robot while the mapping of

Table B.3: File list of the motion estimation toolbox.

Filename	Description
`MotionDemo.m`	Demonstration of the motion estimation toolbox.
`Motion7p.m`	Estimates the motion using the seven points method.
`MotionLSeig.m`	Estimates the motion using the least-squares technique with the eigen analysis.
`MotionIRLS.m`	Estimates the motion using the Iterative Reweighted Least Squares estimator.
`MotionMIRLS.m`	Estimates the motion using the Modified Iterative Reweighted Least Squares estimator.
`MotionLMedS.m`	Estimates the motion using the Least Median of Squares regression.
`MotionMa.m`	Estimates the motion using the Liner Differential Method from Y. Ma, J. Košecká and S. Sastry.
`MotionBaumela.m`	Estimates the motion using the Liner Differential Method from L. Baumela, L. Agapito, P. Bustos and I. Reid.
`MotionvBruss.m`	Estimates the translation motion using the Bruss and Horn method.
`MotionwBruss.m`	Estimates the angular motion using the Bruss and Horn method.
`MotionvPrazdny.m`	Estimates the translation motion using the Prazdny method.
`MotionwPrazdny.m`	Estimates the angular motion using the Prazdny method.
`MotionvHeeger.m`	Estimates the translation motion using the Heeger and Jepson method.
`MotionwHeeger.m`	Estimates the angular motion using the Heeger and Jepson method.
`MotionvTomasi.m`	Estimates the translation motion using the Tomasi and Shi method.
`MotionwTomasi.m`	Estimates the angular motion using the Tomasi and Shi method.
`MotionKanataniA.m`	Estimates the motion using the Kanatani method without renormalization.
`MotionKanataniB.m`	Estimates the motion using the Kanatani method with renormalization.
`MotionRobAlphaLDeig.m`	Estimates the robot motion using the least-squares technique with the eigen analysis.
`MotionRobAlphaIRLS.m`	Estimates the robot motion using the Iterative Reweighted Least Squares estimator.
`MotionRobAlphaMIRLS.m`	Estimates the robot motion using the Modified Iterative Reweighted Least Squares estimator.
`MotionRobAlphaLMedS.m`	Estimates the robot motion using the Least Median of Squares regression.
`MotionRobAlphaMa.m`	Estimates the robot motion using the Liner Differential Method from Y. Ma, J. Košecká and S. Sastry.
`MotionRobAlphaBaumela.m`	Estimates the robot motion using the Liner Differential Method from L. Baumela, L. Agapito, P. Bustos and I. Reid.

its surrounding is built is briefly detailed at the end of this appendix.

B.2.1 Camera Calibration Application

This application calibrates a camera with the methods described in chapter 2 by means of a set of 3D object points and their 2D image projections. Moreover, the application computes the accuracy given by all the methods and saves the calibrating parameters obtained for further use.

Figure B.1 shows the main window of this application which is composed of seven parts:

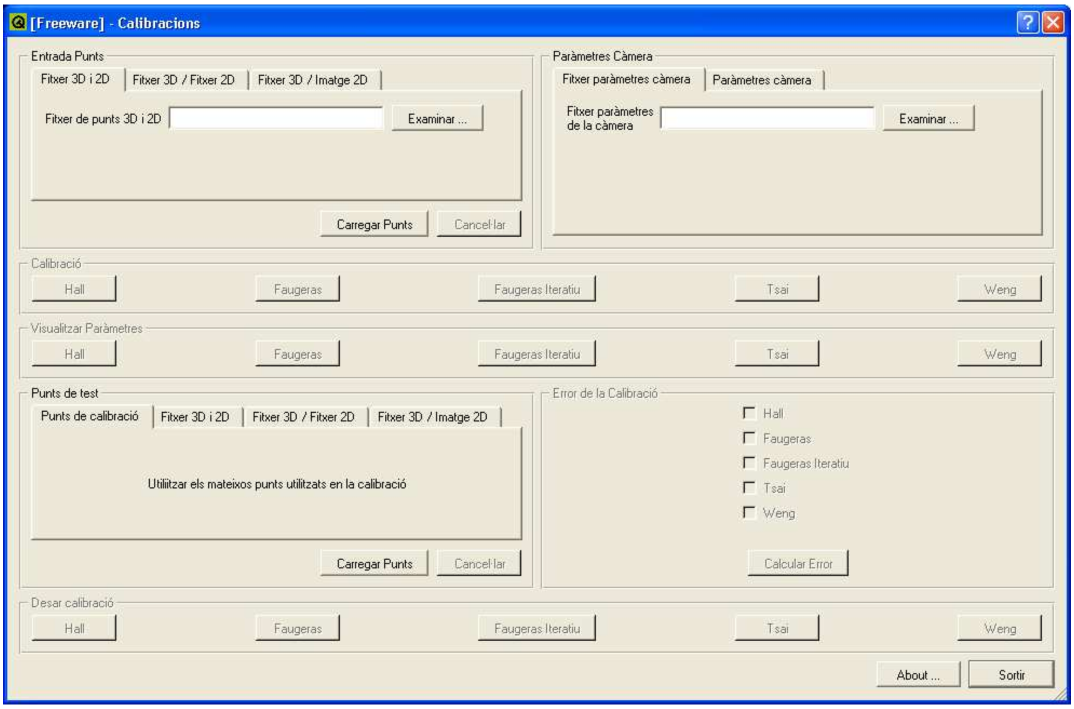
1. The input box of the calibration points.
2. The input box to introduce the camera parameters used by Tsai's method.
3. The set of buttons to execute the calibration methods.
4. The display area to show the calibration parameters obtained.
5. The input box to introduce the test point.
6. The display area to show the accuracy for every method obtained.
7. The set of buttons to save the camera calibration parameters obtained.

The reader is directed to the online help for the application and for further details which include a complete programming manual. However, we have considered it necessary to highlight the camera calibration hierarchy composed of five classes (see Figure B.1). The base class is an abstract class and is derived by different calibration methods. Table B.4 shows the file list of the camera calibration hierarchy class.

B.2.2 Fundamental Matrix Estimation Application

This application is equivalent to the camera calibration application but applied to the fundamental matrix estimation. The application estimates the fundamental matrix from all the surveyed methods and explained in section 3.4 by just providing the image point correspondences.

The main window of this application (see Figure B.3) is divided into seven parts:



Utilitzar els mateixos punts utilitzats en la calibració

Figure B.1: Camera calibration application.

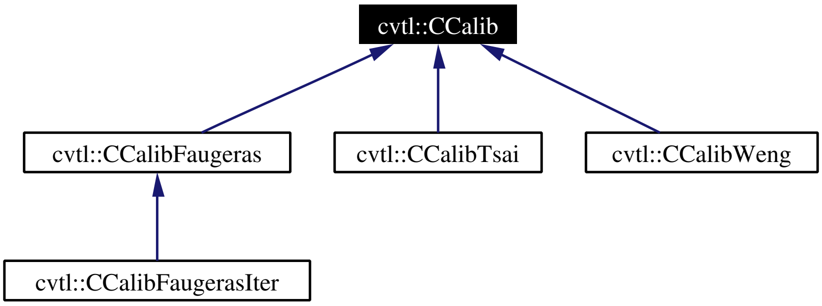


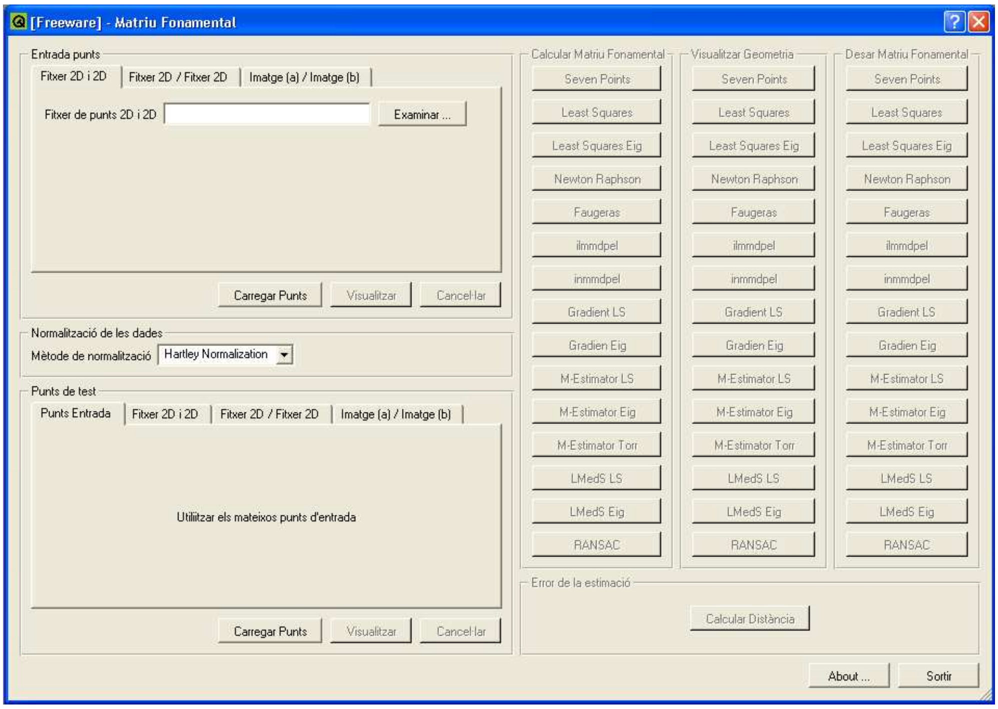
Figure B.2: Camera calibration hierarchy class.

Table B.4: File list of the camera calibration hierarchy class.

Filename	**Description**
`CCalib.h`	CCalib abstract class definition.
`CCalib.cpp`	CCalib class implementation.
`CCalibHall.h`	CCalibHall class definition.
`CCalibHall.cpp`	CCalibHall class implementation.
`CCalibFaugeras.h`	CCalibFaugeras class definition.
`CCalibFaugeras.cpp`	CCalibFaugeras class implementation.
`CCalibFaugerasIter.h`	CCalibFaugerasIter class definition.
`CCalibFaugerasIter.cpp`	CCalibFaugerasIter class implementation.
`CCalibTsai.h`	CCalibTsai class definition.
`CCalibTsai.cpp`	CCalibTsai class implementation.
`CCalibWeng.h`	CCalibWeng class definition.
`CCalibWeng.cpp`	CCalibWeng class implementation.

1. The input box to introduce the image points and matchings used in **F** estimation.
2. The selection criteria to normalize the data points.
3. The input box to introduce the image points and matchings used in accuracy evaluation.
4. The set of buttons to estimate the fundamental matrix.
5. The display area to show the image points and epipolar lines computed.
6. The set of buttons to save the fundamental matrix components.
7. The button to display the accuracy evaluation obtained in the computation for each method.

The reader is directed to the online help for the application and for further details which include a complete programming manual. Note: the fundamental matrix estimator hierarchy composed of a base class and a derived class for each method implemented is included in Figure B.4. The base class is an abstract class with a skeleton for a generic estimator. Table B.5 shows the file list of fundamental matrix hierarchy class.



Mètode de normalització: Hartley Normalization

Utilitzar els mateixos punts d'entrada

Figure B.3: Fundamental matrix estimation application.

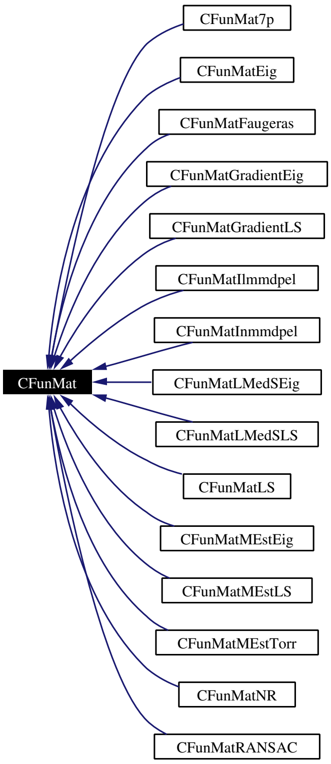


Figure B.4: Fundamental matrix estimator hierarchy class.

Table B.5: File list of the fundamental matrix estimation hierarchy class.

Filename	**Description**
`CFunMat.h`	CFunMat abstract class definition.
`CFunMat.cpp`	CFunMat class implementation.
`CFunMat7p.h`	CFunMat7p class definition.
`CFunMat7p.cpp`	CFunMat7p class implementation.
`CFunMatLS.h`	CFunMatLS class definition.
`CFunMatLS.cpp`	CFunMatLS class implementation.
`CFunMatEig.h`	CFunMatEig class definition.
`CFunMatEig.cpp`	CFunMatEig class implementation.
`CFunMatFaugeras.h`	CFunMatFaugeras class definition.
`CFunMatFaugeras.cpp`	CFunMatFaugeras class implementation.
`CFunMatNR.h`	CFunMatNR class definition.
`CFunMatNR.cpp`	CFunMatNR class implementation.
`CFunMatIlmmdpel.h`	CFunMatIlmmdpel class definition.
`CFunMatIlmmdpel.cpp`	CFunMatIlmmdpel class implementation.
`CFunMatInmmdpel.h`	CFunMatInmmdpel class definition.
`CFunMatInmmdpel.cpp`	CFunMatInmmdpel class implementation.
`CFunMatGradientLS.h`	CFunMatGradientLS class definition.
`CFunMatGradientLS.cpp`	CFunMatGradientLS class implementation.
`CFunMatGradientEig.h`	CFunMatGradientEig class definition.
`CFunMatGradientEig.cpp`	CFunMatGradientEig class implementation.
`CFunMatMEstLS.h`	CFunMatMEstLS class definition.
`CFunMatMEstLS.cpp`	CFunMatMEstLS class implementation.
`CFunMatMEstEig.h`	CFunMatMEstEig class definition.
`CFunMatMEstEig.cpp`	CFunMatMEstEig class implementation.
`CFunMatMEstTorr.h`	CFunMatMEstTorr class definition.
`CFunMatMEstTorr.cpp`	CFunMatMEstTorr class implementation.
`CFunMatLMedSLS.h`	CFunMatLMedSLS class definition.
`CFunMatLMedSLS.cpp`	CFunMatLMedSLS class implementation.
`CFunMatLMedSEig.h`	CFunMatLMedSEig class definition.
`CFunMatLMedSEig.cpp`	CFunMatLMedSEig class implementation.
`CFunMatRANSAC.h`	CFunMatRANSAC class definition.
`CFunMatRANSAC.cpp`	CFunMatRANSAC class implementation.

B.2.3 Localization and Mapping Engine

This application is composed of more than 28.000 lines of code, organized in 140 files. Nevertheless, we considered interesting the listing of at least one set of the most important objects which are the following:

Table B.6: Class list of localization and mapping.

Class name	**Description**
GrillEngine	Main class which contains the whole localization and mapping engine.
FuncMode	Class which determines the functioning mode.
Configuration	Class which determines the configuration parameters.
Heuristics	Class which determines the heuristics parameters.
Exception	Exception class to control error .
StereoCamera	Class which acquires stereo images.
Image	Class to store an image frame.
Point2D	Class to store a 2D point (x, y).
Point3D	Class to store a 3D point (x, y, z).
Position	Class to store a robot position (x, y, α).
RGBtoI	Class which implements the image conversion from color to intensity.
EraseDistortion	Class which implements the erase distortion task.
Corners	Class which implements the corner detector task.
CrossCorrelation	Class which implements the temporal cross correlation task.
CrossCorrelationFM	Class which implements the spatial cross correlation task.
StereoReconstruction	Class which implements the stereo reconstruction task.
Tracker	Class which implements the tracker task and stores 2D and 3D points.
DetectOutliers	Class which implements the outlier detector task.
LocalLocalization	Class which implements the local localization task.
GlobalLocalization	Class which implements the global localization task.
Map3D	Class which implements the map task.

The reader is directed to the online help for the application and for further details which include a complete programming manual.

Bibliography

[Act 00] ActivMedia Robotics. *Pioneer 2 Mobile Robots with Pioneer 2 Operating System Servers*, May 2000. version 6.

[Ahlers 89] R. Ahlers and J. Lu. *Stereoscopic Vision - An Application Oriented Overview.* SPIE - Optics, Illumination, and Image Sensing for Machine Vision IV, vol. 1194, pages 298–307, 1989.

[Armangué 00] X. Armangué, J. Salvi and J. Batlle. *A Comparative Review of Camera Calibrating Methods with Accuracy Evaluation.* In Proceedings of the 5th Ibero-American Symposium on Pattern Recognition, Lisboa, Portugal, September 11-13 2000.

[Armangué 02] X. Armangué, H. Araújo and J. Salvi. *Differential Epipolar Constraint in Mobile Robot Egomotion Estimation.* In Proceedings of the IEEE International Conference on Pattern Recognition, volume 2, pages 599–602, Québec, Canada, 2002.

[Armangué 03a] X. Armangué, H. Araújo and J. Salvi. *A Review on Egomotion by Means of Differential Epipolar Geomety Applied to the Movement of a Mobile Robot.* Accepted to be published in Pattern Recognition, 2003.

[Armangué 03b] X. Armangué and J. Salvi. *Overall View Regarding Fundamental Matrix Estimation.* Image and Vision Computing, vol. 21, no. 2, pages 205–220, 2003.

[Barron 92] J. L. Barron, D. J. Fleet, S. S. Beauchemin and T. Burkitt. *Performance Of Optical Flow Techniques.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 236–242, Champaigne, Illinois, USA, 1992.

[Batista 99] J. Batista, H. Araújo and A. T. de Almeida. *Iterative Multistep Explicit Camera Calibration.* IEEE Transactions on Robotics and Automation, vol. 15, no. 5, pages 897–916, October 1999.

[Batlle 98] J. Batlle, E. Mouaddib and J. Salvi. *A Survey: Recent Progress in Coded Structured Light as a Technique to Solve the Correspondence Problem.* Pattern Recognition, vol. 31, no. 7, pages 963–982, 1998.

[Baumela 00] L. Baumela, L. Agapito, P. Bustos and I. Reid. *Motion Estimation Using the Differential Epipolar Equation.* In 15th International Conference on Pattern Recognition, volume 3, pages 848–851, Barcelona, September 3-7 2000.

[Bober 98] M. Bober, N. Greorgis and J. Kittler. *On Accurate and Robust Estimation of Fundamental Matrix.* Computer Vision and Image Understanding, vol. 72, no. 1, pages 39–53, October 1998.

[Borenstein 96] J. Borenstein, B. Everett and L. Feng. Navigating mobile robots: Systems and techniques. A.K. Petters, Ltd, Wellesley, MA, 1996.

[Broggi 98] A. Broggi. *Vision-Based Driving Assistance in Vehicles of the Future.* IEEE Intelligent Systems, vol. 13, no. 6, pages 22–23, Nov-Dec 1998.

[Brooks 96] M. J. Brooks, L. Agapito, D. Q. Huynh and L. Baumela. *Direct Methods for Self-Calibration of a Moving Stereo Head.* In Proceedings of the 4th European Conference on Computer Vision, volume 2, pages 415–426, Cambridge, UK, April 1996.

[Brooks 97] M. J. Brooks, W. Chojnacki and L. Baumela. *Determining the Ego-Motion of an Uncalibrated Camera from Instantaneous Optical Flow.* Journal of the Optical society of America, vol. 10, pages 2670–2677, 1997.

[Brooks 98] M. J. Brooks, W. Chojnacki, A. Van Den Hengel and L. Baumela. *Robust Techniques for the Estimation of Structure from Motion in the Uncalibrated Case.* In Proceedings of the 5th European Conference on Computer Vision, volume 1, pages 281–295, Freiburg, Germany, June 1998.

[Brooks 01] M. J. Brooks, W. Chojnacki, D. Gawley and A. van den Hengel. *What Value Covariance Information in Estimating Vision Parameters?* In I. C. Society, editor, Proceedings of the 8th IEEE International Conference on Computer Vision, volume 1, pages 302 –308, July 7-14 2001.

[Bruss 83] A. R. Bruss and B. K. Horn. *Passive Navigation.* Computer Vision, Graphics and Image Processing, vol. 21, pages 3–20, 1983.

[Casals 89] A. Casals. Sensor devices and systems for robotics, volume 52. Springer-Verlag. NATO ASI Series, Berlin Heidlberg, 1989.

[Chantler 94] M. Chantler. *The Effect of Variation in Illuminant Direction on Texture Classification.* PhD thesis, Dept. Computing and Electrical Engineering, Heriot-Watt University, 1994.

[Charbonnier 95] L. Charbonnier and A. Fournier. *Heading Guidance and Obstacles Localization for an Indoor Mobile Robot.* IEEE International Conference on Advanced Robotics, pages 507–513, 1995.

[Chen 90] S. Chen and W. Tsai. *A Systematic Approach to Analytic Determination of Camera Parameters by Line Features.* Pattern Recognition, vol. 23, no. 8, pages 859–877, 1990.

[Chojnacki 00] W. Chojnacki, M. J. Brooks, A. van den Hengel and D. Gawley. *On the Fitting of Surfaces to Data with Covariances.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 11, pages 1294 –1303, 2000.

[Chojnacki 02] W. Chojnacki, M. J. Brooks, D. Gawley and A. van den Hengel. *A New Approch to Constrained Parameter Estimator*

Applicable to some Computer Vision Problems. In Proceedings of Statistical Methods in Video Processing Workshop held in conjunction with ECCV 02, Copenhagen, Denmark, June 2002.

[Csurka 97] G. Csurka, C. Zeller, Z. Zhang and O. D. Faugeras. *Characterizing the Uncertainty of the Fundamental Matrix.* Computer vision and Image Understanding, vol. 68, no. 1, pages 18–36, October 1997.

[Deriche 94] R. Deriche, Z. Zhang, Q.-T. Luong and O. Faugeras. *Robust Recovery of the Epipolar Geometry for an Uncalibrated Stereo Ring.* European Conference on Computer Vision, vol. 800, pages 567–576, 1994.

[DeSouza 02] G. N. DeSouza and A. C. Kak. *Vision for Mobile Robot Navigation: A Survey.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 2, pages 237–267, February 2002.

[Echigo 90] T. Echigo. *A Camera Calibration Technique Using Three Sets of Parallel Lines.* Machine Vision and Applications, vol. 3, pages 159–167, 1990.

[Faugeras 86] O. D. Faugeras and G. Toscani. *The Calibration Problem for Stereo.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 15–20, Los Alamitos, CA, June 1986.

[Faugeras 92a] O. D. Faugeras, Q. T. Luong and S. J. Maybank. *Camera Self-Calibration: Theory and Experiments.* European Conference on Computer Vision, pages 321–334, 1992.

[Faugeras 92b] O. D. Faugeras. *What Can Be Seen in Three Dimensions with an Uncalibrated Stereo Ring?* European Conference on Computer Vision, vol. 588, pages 563–578, 1992.

[Faugeras 93] O. D. Faugeras. Three-dimensional computer vision. The MIT Press, Cambridge, Massachusetts, 1993.

[Favaro 02] P. Favaro and S. Soatto. *Learning Shape from Defocus.* In Proceedings of European Conference on Computer Vision, volume 2 of *Lecture Notes in Computer Science*, pages 735–745, Copenhagen, Denmark, 2002. Springer.

[García 01a] R. García. *A Proposal to Estimate the Motion of an Underwater Vehicle Through Visual Mosaicking.* PhD thesis, Universitat de Girona, 2001.

[García 01b] R. García, X. Cufí and J. Batlle. *Detection of Matchings in a Sequence of Underwater Images Through Texture Analysis.* In Proceedings of IEEE International Conference on Image Processing, pages 361–364, Thessaloniki, Greece, 2001.

[Gibbins 94] D. Gibbins. *Estimating Illumination Conditions for Shape from Shading.* PhD thesis, The Flinders University of South Australia Bedford Park, Adelaide, Australia, 1994.

[Grossmann 02] E. G. Grossmann. *Maximum Likelihood 3D Reconstruction From One or More Uncalibrated Views Under Geometric Constraints.* PhD thesis, Instituto Superior Técnico, Portugal, 2002.

[Hall 82] E. L. Hall, J. B. K. Tio, C. A. McPherson and F. A. Sadjadi. *Measuring Curved Surfaces for Robot Vision.* Computer Journal, vol. 15, no. 12, pages 42–54, December 1982.

[Haralick 92a] R. M. Haralick and L. G. Shapiro. Computer and robot vision, volume 1. Addison-Wesley Publishing Company, 1992.

[Haralick 92b] R. M. Haralick and L. G. Shapiro. Computer and robot vision, volume 2. Addison-Wesley Publishing Company, 1992.

[Harris 88] C. J. Harris and M. Stephens. *A Combined Corner and Edge Detector.* In Proceedings of 4th Alvey Vision Conference, pages 147–151, Manchester, 1988.

[Hartley 92a] R. Hartley, R. Gupta and T. Chang. *Stereo From Uncalibrated Cameras.* Computer Vision and Pattern Recognition, pages 761–764, 1992.

[Hartley 92b] R. I. Hartley. *Estimation of Relative Camera Position for Uncalibrated Cameras.* European Conference on Computer Vision, pages 579–587, 1992.

[Hartley 93] R. I. Hartley. *Euclidean Reconstruction from Uncalibrated Views.* Second European Workshop on Applications of Invariance in Computer Vision, pages 237–257, 1993.

[Hartley 94] R. I. Hartley. *Self-Calibration from Multiple Views with a Rotating Camera.* European Conference on Computer Vision, vol. 800, pages 471–478, 1994.

[Hartley 95] R. Hartley. *In Defence of the 8-Point Algorithm.* In Proceedings of the 5th International Conference on Computer Vision, pages 1064–1070, Boston, 1995. IEEE Computer Society Press.

[Hartley 97] R. I. Hartley. *Kruppa's Equations Derived from the Fundamental Matrix.* Pattern Analysis and Machine Intelligence, vol. 19, no. 2, pages 133–135, February 1997.

[Hartley 00] R. Hartley and A. Zisserman. Multiple view geometry in computer vision. Cambridge University Press, 2000.

[Heeger 92] D. J. Heeger and A. D. Jepson. *Subspace Methods for Recovering Rigid Motion I: Algorithm and Implementation.* International Journal of Computer Vision, vol. 7, no. 2, pages 95–318, 1992.

[Hong 93] Z. Hong and J. Yang. *An Algorithm for Camera Calibration Using a Three-Dimensional Reference Point.* Pattern Recognition, vol. 26, no. 11, pages 1655–1660, 1993.

[Horn 90] B. K. P. Horn. *Relative Orientation.* International Journal of Computer Vision, vol. 4, pages 59–78, 1990.

[Huang 89] T. S. Huang and O. D. Faugeras. *Some Propieties of the E Matrix in Two-View Motion Estimation.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, no. 12, pages 1310–1312, December 1989.

[Huber 81] P. J. Huber. Robust statistics. John Wiley and Sons, 1981.

[Hutchinson 96] S. Hutchinson, G. D. Hager and P. I. Corke. *A Tutorial on Visual Servo Control.* IEEE Transactions on Robotics and Automation, vol. 12, no. 5, pages 651–670, October 1996.

[Iocchi 98] L. Iocchi and K. Konolige. *A Multiresolution Stereo Vision System for Mobile Robots.* In Proceedings of the Italian AI Association Workshop on New Trends in Robotics Research, Padova, Italy, 1998.

[Ito 91] M. Ito. *Robot Vision Modelling - Camera Modelling and Camera Calibration.* Advanced Robotics, vol. 5, no. 3, pages 321–335, 1991.

[Jang 96] J.-H. Jang and H. Ki-Sang. *Self-Calibration of a Stereo-Camera by Pure Translational Motion.* In Proceedings of the IEEE International Conference on Image Processing, volume 2, pages 297–300, Laussane, Switzerland, September 16-19 1996.

[Jarvis 83] R. A. Jarvis. *A Perspective on Range Finding Techniques for Computer Vision.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 5, no. 2, pages 122–139, 1983.

[Jepson 91] A. D. Jepson and D. J. Heeger. *A Fast Subspace Algorithm for Recovering Rigid Motion.* In Proccedings of the IEEE Workshop on Visual Motion, pages 124–131, Princeton, NJ, 1991.

[Jepson 93] A. D. Jepson and D. J. Heeger. *Linear Subspace Methods for Recovering Translation Direction.* In L. Harris and M. Jenkin, editors, Spatial Vision in Humans and Robots, pages 39–62. Cambridge University Press, 1993.

[Kamata 92] S. Kamata, R. O. Eason, M. Tsuji and E. Kawaguchi. *A Camera Calibration Using 4 Points Targets.* International Conference on Pattern Recognition, vol. 1, pages 550–553, 1992.

[Kanatani 93a] K. Kanatani. *3-D Interpretation of Optical Flow by Renormalization.* International Journal of Computer Vision, vol. 11, no. 3, pages 267–282, 1993.

[Kanatani 93b] K. Kanatani. *Unbiased Estimation and Statistical Analysis of 3-D Rigid Form Two Views.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, no. 1, pages 37–50, January 1993.

[Khadraoui 96] D. Khadraoui, G. Motyl, P. Martinet, J. Gallice and F. Chaumette. *Visual Servoing in Robotics Scheme Using a Camera/Laser-Stripe Sensor.* IEEE International Journal on Robotics and Automation, vol. 12, no. 5, pages 743–750, 1996.

[Lai 93] J. Z. C. Lai. *On the Sensitivity of Camera Calibration.* Journal of Image and Vision Computing, vol. 11, no. 10, pages 656–664, 1993.

[Lauterbach 93] B. Lauterbach, E. Forgber, G. Urban and W. Anheier. *Vision-Based Navigation Assistance for Autonomous Factory Transport Vehicles.* In Proceedings of International Conference on Intelligent Vehicles, pages 329–334, Tokyo, Japan, 1993. IEEE Press.

[Lenz 88] R. K. Lenz and R. Y. Tsai. *Techniques for Calibration of the Scale Factor and Image Center for High Accuracy 3D Machine Vision Metrology.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 10, no. 5, pages 713–720, September 1988.

[Lenz 89] R. K. Lenz and R. Y. Tsai. *Calibrating a Cartesian Robot with Eye-on-Hand Configuration Independent of Eye-to-Hand Relationship.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, no. 9, pages 916–928, 1989.

[Li 94] M. Li. *Camera Calibration of a Head-Eye System for Active Vision.* European Conference on Computer Vision, vol. 800, pages 543–554, 1994.

[Li 96] F. Li, M. Brady and C. Wiles. *Fast Computation Of the Fundamental Matrix for Active Stereo Vision System.* In Proceedings of the 4th European Conference on Computer Vision, volume 1, pages 157–166, Cambridge, UK, April 1996.

[Liu 90] Y. Liu, T. S. Huang and O. D. Faugeras. *Determination of Camera Location from 2-D to 3-D Line and Point Correspondences.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 1, pages 28–37, 1990.

[Longuet-Higgins 81] H. C. Longuet-Higgins. *A Computer Algorithm for Reconstructing a Scene from Two Projections.* Nature, vol. 293, pages 133–135, 1981.

[Lowe 99] D. G. Lowe. *Object Recognition from Local Scale-Invariant Features.* In Proceedings of the International Conference on Computer Vision, pages 1150–1157, Corfu, Greece, 1999.

[Luong 94a] Q.-T. Luong and O. D. Faugeras. *A Stability Analysis of the Fundamental Matrix.* European Conference on Computer Vision, vol. 800, pages 577–588, 1994.

[Luong 94b] Q.-T. Luong and T. Viéville. *Canonic Representation for Geometries of Multiple Projective Views.* European Conference on Computer Vision, vol. 800, pages 589–599, 1994.

[Luong 96] Q.-T. Luong and O. D. Faugeras. *The Fundamental Matrix: Theory, Algorithms, and Stability Analysis.* International Journal of Computer Vision, vol. 17, no. 1, pages 43–75, January 1996.

[Ma 98a] Y. Ma, J. Košecká and S. Sastry. *Motion Recovery Form Images Sequences: Discrete Viewpoint vs. Differential Viewpoint.* In Proceedings of the 5th European Conference on Computer Vision, Freiburg, Germany, June 1998.

[Ma 98b] Y. Ma, J. Košecká and S. Sastry. *Optimal Motion from Image Sequences: A Riemannian Viewpoint.* In Proceeding of the Conference on Mathematical Theory of Networks and Systems, Padova, Italy, 1998.

[Ma 00] Y. Ma, J. Košecká and S. Sastry. *Linear Differential Algorithm for Motion Recovery: A Geometric Approach.* International Journal of Computer Vision, vol. 36, no. 1, pages 71–89, 2000.

[Maybank 93] S. J. Maybank. Theory of reconstruction from image motion, volume 28. Springer-Verlag, 1993.

[McCane 02] B. McCane, B. Galvin and K. Novins. *Algorithmic Fusion for More Robust Feature Tracking.* International Journal of Computer Vision, vol. 49, no. 1, pages 79–89, 2002.

[Mosteller 77] F. Mosteller and J. Tukey. Data and analysis and regression. Addison-Wesley, 1977.

[Newman 95] T. S. Newman. *A Survey of Automated Visual Inspection.* Image Understanding, vol. 61, no. 2, pages 231–262, 1995.

[Pagès 03] J. Pagès, J. Salvi, R. García and C. Matabosch. *Overview of Coded Light Projection Techniques for Automatic 3D Profiling.* In Proceedings of IEEE International Conference on Robotics and Automation, Taipei, Taiwan, September 2003.

[Penna 91] M. Penna. *Camera Calibration: A Quick and Easy Way to Detection the Scale Factor.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 13, no. 12, pages 1240–1245, 1991.

[Prazdny 80] K. Prazdny. *Egomotion and Relative Depth Map from Optical Flow.* Biological Cybernetics, vol. 36, pages 87–102, 1980.

[Prazdny 81] K. Prazdny. *Determining the Instantaneous Direction of Motion from Optical Flow Generated by a Curvilinearly Moving Observer.* Computer Graphics and Image Processing, vol. 17, pages 238–248, 1981.

[Robson 00] R. Robson. Usin the STL. the C++ standard template library. Springer, New York, 2000.

[Rousseeuw 87] P. J. Rousseeuw and A. M. Leroy. Robust regression and outlier detection. John Wiley and Sons, New York, 1987.

[Ruhe 80] A. Ruhe and P. A. Wedin. *Algorithms for Separable Nonlinear Least Squares Problems.* SIAM Review, vol. 22, no. 3, pages 318–337, 1980.

[Salvi 97] J. Salvi. *An Approach to Coded Structured Light to Obtain Three Dimensional Information.* PhD thesis, Universitat de Girona, Departament d'Electrònica, Informàtica i Automàtica, 1997.

[Salvi 98] J. Salvi, J. Batlle and E. M. Mouaddib. *A Robust-Coded Pattern Projection for Dynamic 3D Scene Measurement.* Pattern Recognition Letters, vol. 19, no. 11, pages 1055–1065, September 1998.

[Salvi 01] J. Salvi, X. Armangué and J. Pagès. *A Survey Addressing the Fundamental Matrix Estimation Problem.* In Proceedings of the IEEE International Conference on Image Processing, volume 2, pages 209 –212, October 2001.

[Salvi 02] J. Salvi, X. Armangué and J. Batlle. *A Comparative Review of Camera Calibrating Methods with Accuracy Evaluation.* Pattern Recognition, vol. 35, no. 7, pages 1617–1635, 2002.

[Se 02] S. Se, D. Lowe and J. Little. *Mobile Robot Localization and Mapping with Uncertainty Using Scale-Invariant Visual Landmarks.* The International Journal of Robotics Research, vol. 21, no. 8, pages 735–758, August 2002.

[Shi 94] J. Shi and C. Tomasi. *Good Features to Track.* In IEEE Conference on Computer Vision and Pattern Recognition, Seattle, June 1994.

[Slama 80] C. C. Slama, C. Theurer and S. W. Henriksen. Manual of photogrammetry. American Society of Photogrammetry, Falls Church, VA, 4 edition, 1980.

[Soatto 98] S. Soatto and R. Brockett. *Optimal and Suboptimal Structure From Motion.* In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, pages 282–288, Santa Barbara, CA, USA, 1998.

[Solomon 96] F. Solomon and K. Ikeuchi. *Extracting the Shape and Roughness of Specular Lobe Objects Using Four Light Photometric Stereo.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, pages 449–454, November 1996.

[Stewart 95] C. V. Stewart. *MINPRAN: A New Robust Estimator from Computer Vision.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, no. 10, pages 925–938, 1995.

[Stoer 80] J. Stoer and R. Bulirsch. Introduction to numerical analysis. Springer-Verlag, 1980.

[Stroustrup 97] B. Stroustrup. The C++ programming language. Addison-Wesley, Reading, 1997.

[Taylor 95] C. J. Taylor and D. J. Kriegman. *Structure and Motion from Line Segments in Multiple Images.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, no. 11, pages 1021–1032, November 1995.

[Tian 96] T. Y. Tian, C. Tomasi and D. J. Heeger. *Comparison of Approches to Egomotion Computation.* In Proceedings of Computer Vision and Pattern Recognition, pages 315–320, San Francisco, CA, June 18 - 20 1996.

[Tomasi 92] C. Tomasi and T. Kanade. *Shape and Motion from Image Streams under Orthography.* International Journal of Computer Vision, vol. 9, no. 2, pages 137–154, 1992.

[Tomasi 93] C. Tomasi and J. Shi. *Direction of Heading from Image Deformations.* In Proceedings of Computer Vision and Pattern Recognition, pages 422–427, New York City, NY, June 15-17 1993.

[Torr 97] P. H. S. Torr and D. W. Murray. *The Developement and Comparision of Robust Methods for Estimating the Fundamental Matrix.* International Journal Computer Vision, vol. 24, no. 3, pages 271–300, September 1997.

[Torr 98] P. H. S. Torr and A. Zisserman. *Robust Detection of Degenerate Configurations While Estimating the Fundamental Matrix.* Computer Vision and Image Understanding, vol. 71, no. 3, pages 312–333, September 1998.

[Torr 00] P. H. S. Torr and A. Zisserman. *MLESAC: A New Robust Estimator with Application to Estimating Image Geometry.* Computer Vision and Image Understanding, vol. 78, pages 138–156, 2000.

[Torr 02] P. H. S. Torr. *Bayesian Model Estimation and Selection for Epipolar Geometry and Generic Manifold Fitting.* International Journal of Computer Vision, vol. 50, no. 1, pages 35–61, 2002.

[Toscani 86] G. Toscani and O. D. Faugeras. *Structure and Motion from Two Noisy Prespective Images.* In Proceedings of the IEEE Conference on Robotics and Automation, pages 221–227, San Francisco, CA, April 7-10 1986.

[Toscani 87] G. Toscani. *Systèmes de Calibration et Perception du Movement en Vision Artificielle.* PhD thesis, Université Paris Sud, 1987.

[Trucco 98] E. Trucco and A. Verri. Introductory techniques for 3D computer vision. Prentice-Hall, Inc., 1998.

[Tsai 84] R. Y. Tsai and T. S. Huang. *Uniqueness and Estimation of Three-Dimensional Motion Parameters of Rigid Objects with*

Curved Surfaces. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 6, no. 1, pages 13–27, 1984.

[Tsai 87] R. Y. Tsai. *A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses.* IEEE International Journal on Robotics and Automation, vol. RA-3, no. 4, pages 323–344, August 1987.

[Vanderkooy 96] G. Vanderkooy and G. F. McLean. *Projective Invariants and the Correspondence Problem.* In Proceedings of IEEE International Conference on Image Processing, volume 2, pages 317–320, Lausane, Switzerland, September 1996.

[Viéville 96] T. Viéville and O. D. Faugeras. *The First Order Expansion of Motion Equations in the Uncalibrated Case.* Computer Vision, Graphics, and Image Processing: Image Understanding, vol. 64, no. 1, pages 128–146, 1996.

[Wang 91] L. L. Wang and W. Tsai. *Camera Calibration by Vanishing Lines for 3-D Computer Vision.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 13, no. 4, pages 370–376, 1991.

[Wang 92] C. C. Wang. *Extrinsic Calibration of a Vision Sensor Mounted on a Robot.* IEEE Int. Journal on Robotics and Automation, vol. 8, no. 2, pages 161–175, 1992.

[Weckesser 96] W. Weckesser, G. Appenzeller, A. Von Essen and R. Dillmann. *Exploration of the Environment with an Active and Intelligent Opticla Sensor System.* In Proceedings of IEEE International Conference on Intelligent Robots and Systems, pages 664–671, Osaka, Japan, November 1996.

[Wei 94] G.-Q. Wei and S. De Ma. *Implicit and Explicit Camera Calibration: Theory and Experiments.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16, no. 5, pages 469–480, May 1994.

[Weng 92a] J. Weng, N. Ahuja and T. S. Huang. *Matching Two Prespective Views.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 8, pages 806–825, August 1992.

[Weng 92b] J. Weng, P. Cohen and M. Herniou. *Camera Calibration with Distortion Models and Accuracy Evaluation.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 10, pages 965–980, October 1992.

[Woodham 78] R. Woodham. *Reflectance Map Techniques for Analyzing Surface Defects in Metal Castings.* Rapport technique AI-TR-457, MIT A.I. Laboratory, June 1978.

[Zhang 93] Z. Zhang. *The Matching Problem: The State of the Art.* Rapport technique 2146, Institut National de Recherche en Informatique et en Automatique, December 1993.

[Zhang 94] Z. Zhang, R. Deriche, O. Faugeras and Q.-T. Luong. *A Robust Technique for Matching Two Uncalibred Images Through the Recovery of the Unknown Epipolar Geometry.* Rapport technique 2273, Institut National de Recherche en Informatique et Automatique, May 1994.

[Zhang 96a] Z. Zhang. *On the Epipolar Geometry Between Two Images With Lens Distortion.* In Proceedings of the International Conference on Pattern Recognition, volume 1, pages 407–411, 1996.

[Zhang 96b] Z. Zhang, Q.-T. Luong and O. Faugeras. *Motion of an Uncalibrated Stereo Ring: Self-Calibration and Metric Reconstruction.* IEEE Transactions on Robotics and Automation, vol. 12, no. 1, pages 103–113, February 1996.

[Zhang 98] Z. Zhang. *Determining the Epipolar Geometry and its Uncertainty: A Review.* International Journal of Computer Vision, vol. 27, no. 2, pages 161–198, 1998.

[Zhang 99] T. Zhang and C. Tomasi. *Fast, Robust, and Consistent Camera Motion Estimation.* In Proceedings of the IEEE Computer Vision and Pattern Recognition, volume 1, pages 164–170, Ft. Collins, CO, USA, 1999.

[Zhuang 84] X. Zhuang and R. M. Haralick. *Rigid Body Motion on Optical Flow Image.* In Proceedings of the 1st International Conference on Artificial Intelligence Applications, pages 366–375, Denver, CO, September 17-20 1984.

[Zhuang 88] X. Zhuang, T. S. Huang, N. Ahuja and R. M. Haralick. *A Simplified Linear Optic Flow-Motion Algorithm.* Computer Vision, Graphics, and Image Processing, vol. 42, pages 334–344, 1988.