Universitat de Girona

AN APPROACH TO CODED STRUCTURED LIGHT TO OBTAIN THREE DIMENSIONAL INFORMATION

Joaquim SALVI MAS

ISBN: 84-8458-107-1
Dipòsit legal: GI-1228-2001

PhD. Thesis

UNIVERSITY OF GIRONA

Department of Electronics, Informatics and Automation

to obtain the degree of:

European PhD in Industrial Engineering

by:

Joaquim Salvi

Title of the thesis:

An Approach to Coded Structured Light to Obtain Three Dimensional Information

Submitted and presented in Girona, december 1997 and february 1998 respectively:

Supervisor:	Dr. Joan Batlle (Universitat de Girona – Spain)
Supervisor:	Dr. El Mustapha Mouaddib (Université de Picardie Jules Verne – France)
Referee:	Professor Bernard Dubuisson (Université de Compiègne – France)
Referee:	Dr. Geoff Roberts (University of Wales College Newport – Great Britain)
President:	Professor Josep Amat (Universitat Politècnica de Catalunya – Spain)
Examiner:	Professor Juan Jose Villanueva (Universitat Autònoma de Barcelona – Spain)
Examiner:	Professor Antonio Perez (Universidad Politécnica de Madrid – Spain)
Examiner:	Professor Emilio Zapata (Universidad de Málaga – Spain)
Examiner:	Dr. Geoff Roberts (University of Wales College Newport – Great Britain)

Gratitudes

In the past few years I have been lucky to work with a lot of people which have helped me, in one or another way, with the hard and stimulating work of researching. This sheet of paper tries to thank them. Surely, it is rather difficult to explain the gratitude I feel saying just words, but I will try...

Firstly, I have to thank Joan Batlle, the director of this thesis and also one of my best friends. The help he gave me in a lot of different ways: like writing down that article with a deadline in five days, like searching stays in foreign universities, and the several hours he spent to convince me of some dark aspects of this work, has surely led me to present today this thesis.

I would like to thank the great help given by my friend and co-director of this thesis El Mustapha Mouaddib, who welcomed me in his research group *Equipe Perception en Robotique* at the University of Picardie in Amiens (France). There, I learnt all those aspects of calibration during two three-month stays. I remember the beers we had at the *Bissap Cafe* discussing those complicated equations, and his patience when I monopolised his computer day after day. I also want to mention his wife Valerie and daughter Nissa.

I want to thank all the friends of the research group of Computer Vision and Robotics at the University of Girona, for creating that atmosphere which allows anyone to work without interruptions. For all of them, and for their friendly relationship, thanks to: Xevi Cufí, Joan Martí, Rafa Garcia, Pep Forest, Jordi Freixenet, Pere Ridao, Lluis Pacheco, Toni Fernandez and Pep Cortada. I also want to mention Jordi Regincós and Enrique Torres for their friendly relationship.

I would like to thank Josep Tomas, Salvador Salanova and Xavi Manyer for their technical support, and Olga Moreno for her administrative assistance.

A very special gratitude to all my friends of the *Equipe Perception en Robotique* in Amiens who welcomed me in their lab. Thanks for all the help I received, and for their patiente trying to understand my awful french. I will

never forget the moments we shared. Thanks to: Laurent Delahoche, Pascal Vasseur, Bruno Marhic, Claude Pegard, Eric Brassart, and in general every one of the *Equipe Perception en Robotique*.

Especially, I want to thank Patricia Eyskens for her friendly help with my english, revising the published articles and even this thesis.

I would like to thank Pep Vehí and Joan Martí for the time we spent talking about how to solve some of the problems involved with LaTeX and *Scientific WorkPlace*.

I want to thank the people of the Computer Vision Centre of the *Universitat Autonoma de Barcelona*, who allows me to use their library.

I am also grateful to all the people who gave me their advices and recommendations about the contents of this thesis. Especially, I want to thank Alicia Casals of the *Universitat Politècnica de Catalunya* for her worthy recommendations. I would also like to express my gratitude to the reviewers of this thesis: Geoff Roberts of the University of Wales College and Bernard Dubuisson of the *Université de Tecnologie de Compiègne*.

I have to thank all the people of the Electronics, Informatics and Automation Department and the Institute of Informatics and Applications of the University of Girona as well, and all the people of the *Laboratoire des Systèmes Automatiques* of the University of Picardie.

At last, but for me is the most important, I want to thank the support received from my whole family, especially from my parents Josep Salvi and Pilar Mas and my brother David. I would also like to express my gratitude and love to Alexandra for her patience when a weekend after another I had to work, and we could not go out. I also have to mention her great help revising my english.

I hope, I have not forgotten anybody. If there is someone who can not find his/her name in this list, I apologize and I invite him/her to write it down in the free space which follows this paragraph.

Joaquim Salvi

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Describing the interest of this thesis

The human visual ability to perceive depth looks like a puzzle. We perceive three-dimensional spatial information quickly and efficiently by using the binocular stereopsis of our eyes and, what is more important, the learning of the most common objects which we achieved through living. Nowadays, modelling the behaviour of our brain is a fiction, that is why the huge problem of 3D perception and further interpretation is split into a sequence of easier problems. A lot of research is involved in robot vision in order to obtain 3D information of the surrounded scene. Most of this research is based on modelling the stereopsis of humans by using two cameras as if they were two eyes. This method is known as stereo vision and has been widely studied in the past and is being studied at present, and a lot of work will be surely done in the future. This fact allows us to affirm that this topic is one of the most interesting ones in computer vision.

The stereo vision principle is based on obtaining the three dimensional position of an object point from the position of its projective points in both camera image planes. However, before inferring 3D information, the mathematical models of both cameras have to be known. This step is known as camera calibration and will be broadly described in the following chapter. Perhaps the most important problem in stereo vision is the determination of the pair of homologue points in the two images, known as the correspondence problem, and it is also one of the most difficult problems to be solved which is currently investigated by a lot of researchers. The epipolar geometry allows us to reduce the correspondence problem. An approach to the epipolar geometry is described in chapter 3. Nevertheless, it does not solve it at all as a lot of considerations have to be taken into account. As an exemple we

have to consider points without correspondence due to a surface occlusion or simply due to a projection out of the camera scope.

The interest of this thesis is focused on structured light which has been considered as one of the most frequently used techniques in order to reduce the problems related to stereo vision. Structured light is based on the relationship between a projected light pattern its projection and an image sensor. The deformations between the pattern projected into the scene and the one captured by the camera, permits to obtain three dimensional information of the illuminated scene. This technique has been widely used in such applications as: 3D object reconstruction, robot navigation, quality control, and so on. Although the projection of regular patterns solve the problem of points without match, it does not solve the problem of multiple matching, which leads us to use hard computing algorithms in order to search the correct matches.

In recent years, another structured light technique has increased in importance. This technique is based on the codification of the light projected on the scene in order to be used as a tool to obtain an unique match. Each token of light projected on the scene carries a label indicating from which position in the projector frame it comes. When the token is imaged by the camera, we have to read the label (decode the pattern) in order to solve the correspondence problem. The advantages and disadvantages of stereo vision against structured light and a survey on coded structured light are related in chapter 4. The work carried out in the frame of this thesis has permitted to present a new coded structured light pattern which solves the correspondence problem uniquely and robustly. Unique, as each token of light is coded by a different word which removes the problem of multiple matching. Robust, since the pattern has been coded using the position of each token of light with respect to both co-ordinate axis X and Y. Algorithms and experimental results are included in the thesis. The reader can see examples of 3D measurement of static objects, and the more complicated measurement of moving objects. The technique can be used in both cases as the pattern is coded by a single projection shot. Then it can be used in several applications of robot vision.

Our interest is focused on the mathematical study of the camera and pattern projector models. We are also interested in how these models can be obtained by calibration, and how they can be used to obtained three dimensional information from two correspondence points. Furthermore, we have studied structured light and coded structured light, and we have presented a new coded structured light pattern. However, in this thesis we started from the assumption that the correspondence points could be well-segmented from the captured image. Computer vision constitutes a huge problem and a lot

of work is being done at all levels of human vision modelling, starting from a) image acquisition; b) further image enhancement, filtering and processing, c) image segmentation which involves thresholding, thinning, contour detection, texture and colour analysis, and so on. The interest of this thesis starts in the next step, usually known as depth perception or 3D measurement. So, we did not take into account the lower levels.

1.2 Schema of the thesis

This thesis is focused on the study of structured light and, in particular, on how light can be coded in order to solve the correspondence problem. Furthermore, a new coded structured light pattern is presented, which permits a reconstruction of static and moving objects. The system can be used as a sensor allowing us to measure 3D dynamic scenes.

In chapter 2 the reader is engaged in the study of camera and pattern calibration. The chapter describes the intrinsic parameters which model the optical characteristics and the internal geometry of the sensor, and the extrinsic parameters which model the position and orientation of the sensor with respect to a world co-ordinated system. Furthermore, two algorithms of camera calibration are described in detail. The first one is based on a linear calibration, that is without considering lens distortion. The second one is based on a non-linear calibration of a whole camera model, including lens distortion which leads to a function minimization by iteration. Finally, the way how 3D information of an object point can be obtained knowing only its two correspondence points is described.

Chapter 3 deals with the study of the fundamental matrix. The fundamental matrix is computed by the relationship between two pinhole models, and includes the intrinsic parameters of both pinhole models and the relative position and orientation from one to the other. This matrix has been widely used in order to obtain the epipolar geometry of the system that simplifies considerably the search of the correspondence points.

Chapter 4 is focused on the study of structured light. Structured light reduces considerably the problem of matching, present in stereo vision systems. However, the projection of regular patterns does not solve the problem of multiple matching. The correspondence problem can be solved by coding the light projected on the scene. That is, each token of light projected on the scene carries with it a label indicating where it comes from. When the token is imaged by the camera we only have to read the label to solve the correspondence problem. This chapter surveys the most interesting techniques based on coded structured light projection.

Chapter 5 presents a new coded structured light pattern. The pattern is uniquely coded and consequently it is more robust than periodical coding, and it is coded with respect to both co-ordinate axes. This double coding has been demonstrated to be more accurate than single axe codification. Calibration is explained and the intrinsic and extrinsic parameters are shown. Furtermore, some experimental examples of 3D object measurement are shown and discussed.

Finally, chapter 6 presents the conclusions of the thesis. Further work is also included in some subjects where the research of this thesis can continue.

Chapter 2

Calibration

Inferring three dimensional information leads us to a previous step of system calibration. This chapter describes the two majors matters of calibration. First, how the model of a system is defined by a set of parameters in order to approximate the physical behaviour of the system to a geometrical one. Secondly, how these parameters are obtained by means of a calibration algorithm. Finally, the chapter will describe how 3D reconstruction can be achieved by means of the relationship between two calibrated models.

2.1 Considerations

Since our 3D imaging system is based on a camera sensor and a pattern projector system, two different sets of parameters have to be computed by calibration. Firstly, we have to define the model of a camera sensor and, secondly, the model of the pattern projector system. However, note that, both devices are defined using the same model. That is, the camera and the projector pattern are based both on the pinhole principle. Then, the whole chapter will be focused on the major effort of modelling and calibrating a camera. The reader might note that all the mathematical equations introduced in this chapter are also used to model and calibrate the pattern projector system.

2.2 Modelling a camera

Modelling a camera is based on the definition of the set of parameters that approximate the behaviour of the physical sensor to a geometrical model, known as the pinhole model [Faug 93] [Ayac 91b]. The geometrical model of the pinhole consists of a plane Π called the retinal plane in which the

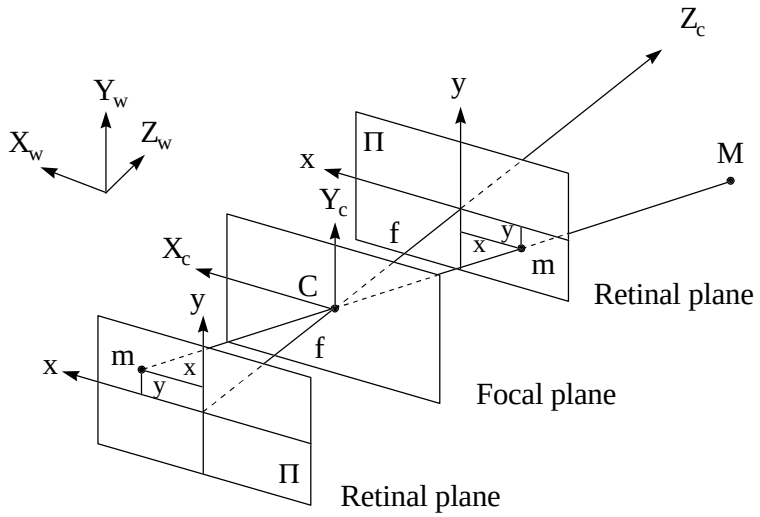

Figure 2.1: The camera co-ordinate system with respect to a world co-ordinate system.

image is formed through a perspective projection. A point C is placed at a fixed distance from the retinal plane. This point is known as the optical centre (also focal point) and it is defined as the pinhole through where some of the light beams reflected by the object pass, forming a perspective image of the scene in the retinal plane. At this point, two considerations must be taken into account. Of course, the optical centre can be placed in front of or behind the camera (see figure 2.1). In the first case, a projective inverted image of the scene is formed in the retinal plane. In the second one, a single projective image is presented. This section describes the model of a camera sensor with a back-placed optical centre. However, it will be shown that both camera configurations are modelled by the same parameters but considering a different sign of the focal distance.

There are two kinds of parameter sets which model the given camera as a pinhole model [Beye 92] [Ito 91]. One is the intrinsic parameter set, as the pinhole point, which models the internal geometry and optical characteristics of the image sensor. The other is the extrinsic parameter set, which models the position and orientation of the sensor with respect to the world co-ordinate system.

The intrinsic parameter set contains: a) the focal distance f, which is

the distance (in mm.) from the retinal plane Π to the optical centre C; b) the conversion parameters k_u and k_v which relate the horizontal and vertical adjustment to go from the camera co-ordinate system (in mm.) located in the retinal plane to the image co-ordinate system located in the image plane and expressed in pixels; and c) the position $(u_0,\ v_0)$ of the principal point (also image centre) defined by the projection of the principal point C on the image plane, expressed in pixels.

The extrinsic parameters set defines the relation between the camera co-ordinate system located on the retinal plane with respect to the world co-ordinate system located at any point in the scene. This relationship is defined by the position and the orientation of the camera co-ordinate system with respect to the world co-ordinate system. The position is given by a 3x1 translation vector T and the orientation is given by the 3x3 rotation matrix $\mathbf{R}$.

Then, up to 11 parameters have to be used to model the camera sensor. This section goes through each parameter in order to obtain the mathematical equations which relate an object point with its projective point. Finally, this linear relationship will be modelled by a 3x4 transformation matrix. However, as a result of some types of imperfections in the design and assembly of the lens composing the optical system, a linear relationship does not hold true. These kinds of imperfections, known as lens distortion, will be discussed. The non-linear mathematical relation between both points is also presented.

2.2.1 The Extrinsic Parameters

The surface points of the scene are referred to a world co-ordinate system, with an origin O_w, and co-ordinate axis (X_w, Y_w, Z_w). Given an object point P, this point is expressed with respect to O_w, that is with the co-ordinates (Px_w, Py_w, Pz_w). Consider a second co-ordinate system, with an origin $C = O_c$, and co-ordinate axis (X_c, Y_c, Z_c), placed in the optical centre of the camera sensor as shown in figure 2.2.

This geometrical relationship permits to define which are the extrinsic parameters of the model, as these parameters will determine the position and orientation of the camera co-ordinate system with respect to the world co-ordinate system. The position of the optical centre $O_c = C$ with respect to O_w is given by the translation vector T shown in equation 2.1.

$$T = \begin{bmatrix} t_X \\ t_Y \\ t_Z \end{bmatrix} \tag{2.1}$$

The orientation of the co-ordinate axis of the camera with respect to the

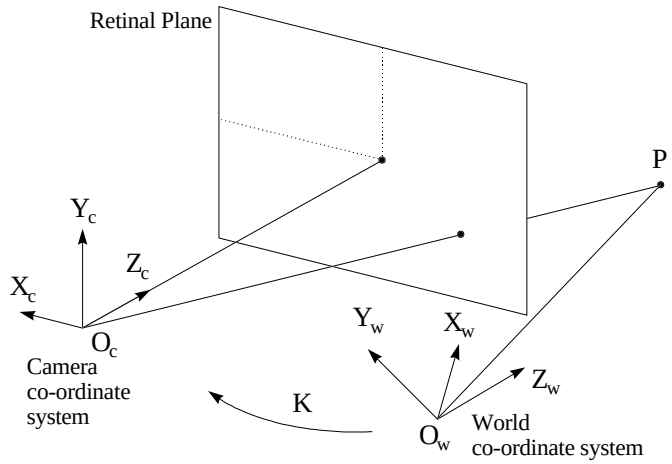

Figure 2.2: Geometric relatioship between the camera co-ordinate system and the world co-ordinate system.

world co-ordinate system is given by a 3x3 rotation matrix $\mathbf{R}$. The rotation matrix can be obtained by the matrix product of three single axe rotations. A rotation of ϕ degrees around an axe $\varkappa$ will be expressed as $Rot(\varkappa, \phi)$. Then, three single axe rotations can be expressed as shown by the following equations.

$$Rot(X, \phi_x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi_x & -s\phi_x \\ 0 & s\phi_x & c\phi_x \end{bmatrix} \tag{2.2}$$

$$Rot(Y, \phi_y) = \begin{bmatrix} c\phi_y & 0 & s\phi_y \\ 0 & 1 & 0 \\ -s\phi_y & 0 & c\phi_y \end{bmatrix} \tag{2.3}$$

$$Rot(Z, \phi_z) = \begin{bmatrix} c\phi_z & -s\phi_z & 0 \\ s\phi_z & c\phi_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

Where $c\phi$ represents the cosinus of the ϕ angle, and $s\phi$ represents the sinus. In the following, two orientation representations are presented in order to obtain the rotation matrix $\mathbf{R}$.

The RPY representation

$$\mathbf{R} = RPY(\phi_z, \phi_y, \phi_x) = Rot(Z, \phi_z) \cdot Rot(Y, \phi_y) \cdot Rot(X, \phi_x) \tag{2.5}$$

The Euler representation

$$\mathbf{R} = Euler(\phi, \theta, \psi) = Rot(Z, \phi) \cdot Rot(Y, \theta) \cdot Rot(Z, \psi) \tag{2.6}$$

Both representations are possible. However, only the Euler representation will be considered. Then the rotation matrix $\mathbf{R}$ is represented by the following matrix.

$$\begin{aligned} \mathbf{R} &= Rot(Z, \phi) \cdot Rot(Y, \theta) \cdot Rot(Z, \psi) = \\ & \begin{bmatrix} c\phi c\psi - s\phi c\theta s\psi & -c\phi s\psi - s\phi c\theta c\psi & s\phi s\theta \\ c\phi c\psi + c\phi c\theta s\psi & -s\phi s\psi + c\phi c\theta c\psi & -c\phi s\theta \\ s\theta s\psi & s\theta c\psi & c\theta \end{bmatrix} \end{aligned} \tag{2.7}$$

In order to simplify the representation of the $\mathbf{R}$ matrix, it is more interesting to represent the matrix with respect to its three orientation vector r_i, where each orientation vector is made by three components.

$$\mathbf{R} = \begin{bmatrix} r_X \\ r_Y \\ r_Z \end{bmatrix} \tag{2.8}$$

Note that, r_X, r_Y and r_Z represent the orientation of the three axis of the camera co-ordinate system with respect to the world co-ordinate system. In the figure 2.3, the relationship of the components of a vector (e.g. Xc camera co-ordinate axe) with respect to the world co-ordinate axe (given by r_X), is shown. Note that any vector could be placed with the origin at the world co-ordinate system without changing its representation, as all the vectors are space-free elements.

Then, both co-ordinate systems are related by the following equation,

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \mathbf{R} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + T \tag{2.9}$$

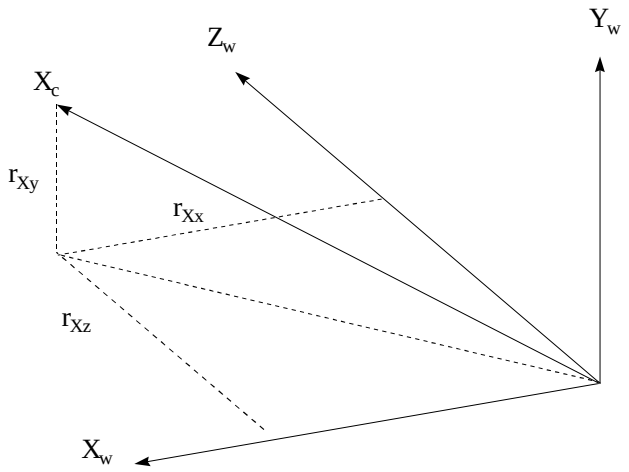

Figure 2.3: Relationship between the orientation vectors r_i and the camera co-ordinate frame. The figure shows only the relationship between the r_X vector and the X_c axis.

so that, given an object point P, with the components(P_{Xw}, P_{Yw}, P_{Zw}) with respect to O_w, one can compute its representation with respect to O_c with the components (P_{Xc}, P_{Yc}, P_{Zc}),

$$\begin{bmatrix} P_{Xc} \\ P_{Yc} \\ P_{Zc} \end{bmatrix} = \mathbf{R} \begin{bmatrix} P_{Xw} \\ P_{Yw} \\ P_{Zw} \end{bmatrix} + T \tag{2.10}$$

Equation 2.10 is usually espressed by a single matrix relationship with homgeneous co-ordinates, as shown by equation 2.11 (see also the figure 2.2),

$$\begin{bmatrix} P_c \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} P_w \\ 1 \end{bmatrix} \tag{2.11}$$

$$\mathbf{K} = \begin{bmatrix} \mathbf{R}_{3x3} & T_{3x1} \\ 0_{1x3} & 1 \end{bmatrix} \tag{2.12}$$

Summarising the section, six extrinsic parameters are obtained in order to model the translation and orientation of the camera co-ordinate system with respect to the world co-ordinate system. The six parameters are: the three components of the translation vector $T = (t_X, t_Y, t_Z)$ and the three angles (ϕ, θ, ψ) which define the rotation matrix $\mathbf{R}$.

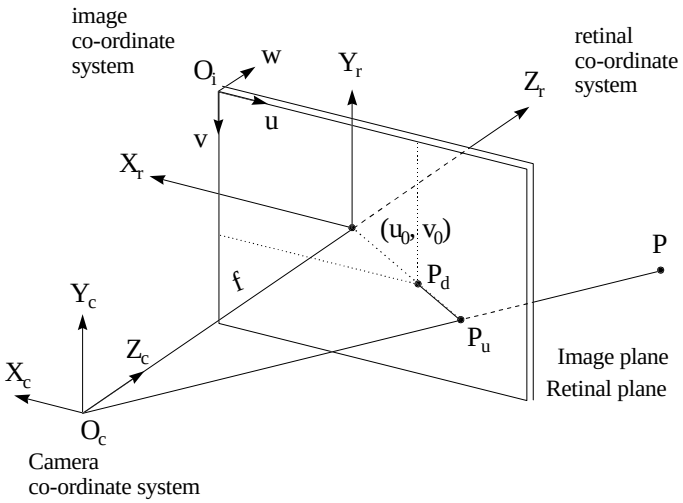

Figure 2.4: Geometrical relationship between a 3D object point and its 2D image projection.

2.2.2 The Intrinsic Parameters

At this point, the section will focus on the modelling of the intrinsic parameters of the camera sensor. That is, it will focus on the determination of the parameters which model the internal geometry and optical characteristics of the image sensor. The intrinsic camera model will relate a 3D object point with respect to its observable 2D projection on the image plane. The 3D object point must be expressed with respect to the camera co-ordinate system, and the 2D image point will be expressed with respect to the image co-ordinate system, in pixels. The geometrical model that has to be defined is illustrated in figure 2.4.

The modelling of the intrinsic parameters will be split in four steps:

Ideal projection. Models the projective relationship between a 3D object point and its 2D imaged point.

Lens distortion. Takes into account the discrepancy between the ideal 2D point and the observed 2D point.

Pixel adjustment. Transforms the 2D observed point from metric co-ordinates to pixels.

Principal point. Gives the 2D observed point co-ordinates with respect to the computer co-ordinate system.

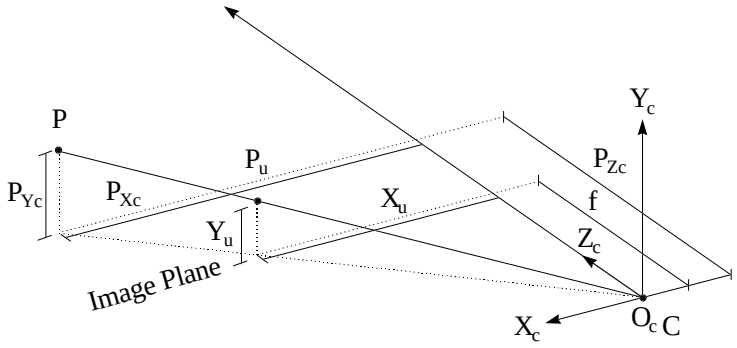

Figure 2.5: Obtaining the ideal projection from the perspective relation.

Each step is explained in detail hereafter.

Step 1. Modelling the ideal projection

Consider that any optical sensor can be modelled as a pinhole camera. Note that the camera co-ordinate system has been located at the optical centre C, with the Z_c axe along the optical axis of the image sensor. Consider that the image plane is located at a distance f from the optical centre C, and it is parallel to the plane defined by the co-ordinate axis X_c and Y_c. Starting from these assumptions, given an object point P referred to O_c, its ideal projection P_u on the image plane must lie on the line passing through the object point P and the optical centre C. In order to obtain the equations which relate the 2D point P_u with respect to P and f, the geometric law of the perspective relation will be used [Kana 91] [Hara 93]. This relation is shown in figure 2.5.

Then, the perspective relation is given by the following equations,

$$\begin{aligned} \frac{X_u}{f} &= \frac{P_{X_c}}{P_{Z_c}} \\ \frac{Y_u}{f} &= \frac{P_{Y_c}}{P_{Z_c}} \end{aligned} \tag{2.13}$$

So that,

$$X_u = f\frac{P_{X_c}}{P_{Z_c}}$$

$$Y_u = f\frac{P_{Y_c}}{P_{Z_c}} \tag{2.14}$$

Which can be expressed in a matricial way applying,

$$\begin{bmatrix} sX_u \\ sY_u \\ s \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{f} \end{bmatrix} \begin{bmatrix} P_{Xc} \\ P_{Yc} \\ P_{Zc} \end{bmatrix} \tag{2.15}$$

Summarising the step, a single intrinsic parameter has to be considered in order to model the ideal projection. This parameter is the focal distance f.

Step 2. Modelling the lens distortion

As a result of some types of imperfections in the design and assembly of the lens composing the optical system, the linear relationship of the perspective projection does not hold true [Weng 92]. These imperfections imply that the observed projection on the image plane differs from the ideal one defined in the previous step [Gosh 89]. Then, another two equations which explicitly take into account the positional error must be introduced, where (X_u, Y_u) is the ideal projection and (X_d, Y_d) are the co-ordinates of the observed point on the image plane.

$$\begin{aligned} X_u &= X_d + D_x \\ Y_u &= Y_d + D_y \end{aligned} \tag{2.16}$$

Lens distortion can be modelled by a radial and tangential approximation, shown in figure 2.6.

Radial distortion: Radial distortion causes an inward or outward displacement of a given image point from its ideal projection. This type of distortion is mainly caused by flawed radial curvature of the lens. A negative radial displacement of the image points is referred to as barrel distortion. It causes outer points to crowd increasingly together and the scale to decrease. A positive radial displacement is referred to as pincushion distortion. It causes outer points to spread and the scale to increase. This type of distortion is strictly symmetric about the optical axis. Figure 2.7 illustrates the effect of radial distortion.

The displacement given by the radial distortion can be modelled by the following equations, where r is the radial distance of the observed projection from the projection of the focal point on the image plane (known as principal point), and k_1, k_2, ... are the coefficients of the radial distortion.

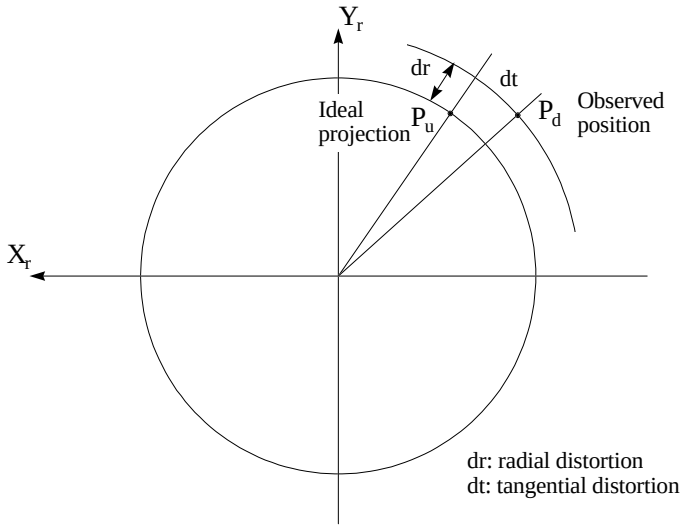

Figure 2.6: Radial and tangential distortion.

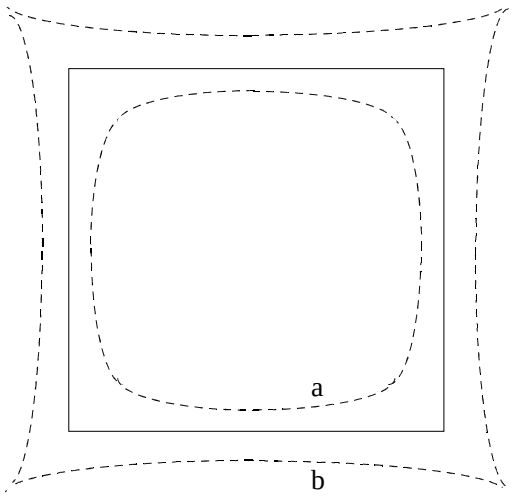

Figure 2.7: Effect of radial distortion. Solid lines: no distortion; dashed lines: radial distortion (a: negative, b: positive).

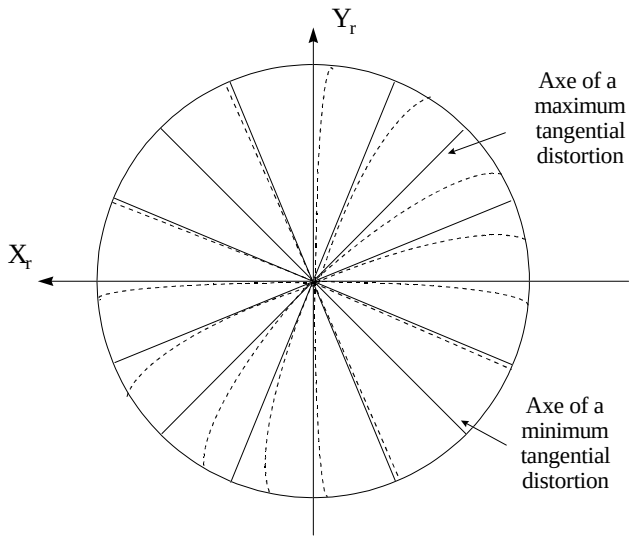

Figure 2.8: Effect of tangential distortion. Solid lines: no distortion; dashed lines: tangential distortion.

$$\begin{aligned} D_x &= X_d(k_1 r^2 + k_2 r^4 + ...) \\ D_y &= Y_d(k_1 r^2 + k_2 r^4 + ...) \\ r &= \sqrt{X_d^2 + Y_d^2} \end{aligned} \tag{2.17}$$

Tangential distortion: Tangential distortion is introduced in order to model the actual optical systems subjected to various degrees of decentering, that is, the optical centers of the lens elements are not strictly collinear. These kinds of imperfections arise from imperfections in the lens design and manufacturing as well as from camera assembly. The effect produced by tangential distortion is shown in figure 2.8.

Tangential distortion is also modelled by another infinite serie [Weng 92]. However, most of the authors have affirmed that radial distortion is the most important lens distortion and therefore, it is the only one that has to be considered. Any more elaborate modelling would not only not give more accurate results, but also would cause numerical instability in calibration [Tsai 87].

This section takes only into account the first coefficient k_1 of radial distortion, that is the coefficient modelling the major contribution of radial distortion, which has been considered the widest used model of lens distortion.

It seems that very little improvement is given modelling more coefficients. Then, equations 2.17 are simplified, obtaining,

$$\begin{aligned} D_x &= X_d k_1 r^2 \\ D_y &= Y_d k_1 r^2 \\ r &= \sqrt{X_d^2 + Y_d^2} \end{aligned} \tag{2.18}$$

Summarising the step, a single parameter k_1 as the first coefficient of the serie which models the radial distortion has to be considered.

Step 3. Modelling the pixel adjustment

This step of camera modelling is based on the transformation of the real projection on the retinal plane from metric co-ordinates to the co-ordinates of the computer buffer (image plane) in pixels. The relation of both planes is shown in figure 2.9. Actually, this transformation is based on a simple scale adjustment of both axes, shown in equation 2.19. Note that (X_d, Y_d) is the real projection with lens distortion in metric co-ordinates, and (X_p, Y_p) is the same point expressed in pixels.

$$\begin{aligned} X_p &= k_u X_d \\ Y_p &= k_v Y_d \end{aligned} \tag{2.19}$$

Which can be modelled in a matricial way applying the following matricial equation,

$$\begin{bmatrix} X_p \\ Y_p \end{bmatrix} = \begin{bmatrix} k_u & 0 \\ 0 & k_v \end{bmatrix} \begin{bmatrix} X_d \\ Y_d \end{bmatrix} \tag{2.20}$$

Summarising the step, two corrected coefficients have to be modelled, that is parameters k_u and k_v.

Step 4. Obtaining the principal point

The principal point (also image centre) is defined as a point (u_0, v_0) on the image plane given by the intersection of the optical axis of the camera and the image plane. That is, the principal point is the projection of the focal point on the image plane which is given in pixels. The principal point is considered as the origin of the retinal image plane (X_r, Y_r), and it has also been used as the origin for modelling the lens distortion of the camera. However, the

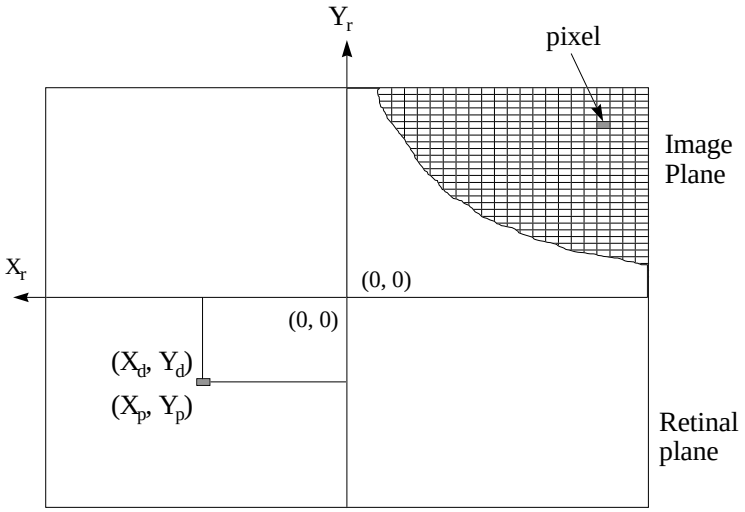

Figure 2.9: Transformation from metric co-ordinates of the retinal plane to the pixel co-ordinates of the image plane.

origin of the co-ordinate system in the computer image plane is located at the north-west corner of the image. Figure 2.10 illustrates this relation. Then, a translation is needed in order to place the principal point from the corner of the image to its real location on the image plane. This translation is given by the values (u_0, v_0), with respect to each axe of the co-ordinate system. Note also that the U axe and V axe of the computer image plane are inverted from the X_r axe and the Y_r axe of the retinal image plane.

Given a point (X_p, Y_p) expressed with respect to the camera co-ordinate system, the same point will be expressed with respect to the computer image co-ordinate system using the following equations. The resulting point representation is called (X_i, Y_i), which is the one observed by the user in pixels on the image plane of the computer.

$$\begin{aligned} X_i &= -X_p + u_0 \\ Y_i &= -Y_p + v_0 \end{aligned} \tag{2.21}$$

Which can be modelled in a matricial way applying the following equation,

$$\begin{bmatrix} X_i \\ Y_i \end{bmatrix} = \begin{bmatrix} -1 & 0 & u_0 \\ 0 & -1 & v_0 \end{bmatrix} \begin{bmatrix} X_p \\ Y_p \\ 1 \end{bmatrix} \tag{2.22}$$

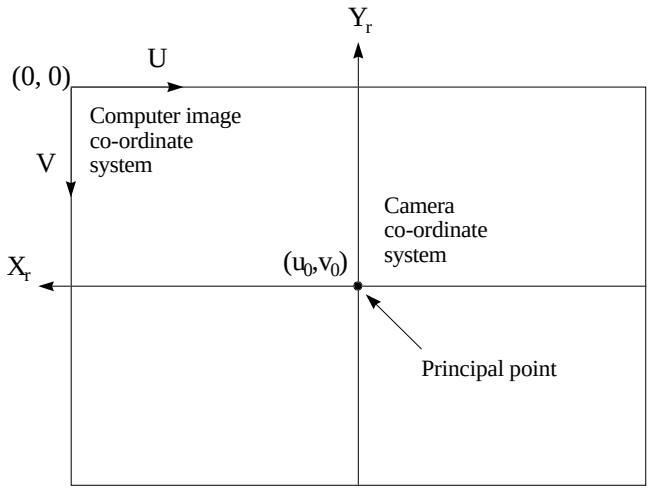

Figure 2.10: Transformation from the co-ordinate system of the camera image plane to the co-ordinate system of the computer image buffer.

Summarising the step, the position of the principal point on the image plane (u_0, v_0) has to be modelled.

2.2.3 Summary of Camera modelling

This section illustrates graphically the steps of camera modelling in order to provide a brief summary for the reader. Two different camera models will be given. Firstly, a camera model without lens distortion which allows us to express the relation between the 3D object point and the 2D observed projection using linear equations. Secondly, a camera model with radial lens distortion, which will be modelled using non-linear equations. Note that the 3D object point is referred to a world co-ordinate system and the observed 2D point on the image plane of the computer is expressed in pixels.

The whole camera model without lens distortion

In order to express the relation between the metric 3D object point referred to a world co-ordinate system and its observed 2D projection on the computer image plane, expressed in pixels, the matricial equations will be expanded. This relation is usually expressed as the product of two matrices. The first matrix expresses the entire intrinsic parameters of the camera sensor, and the

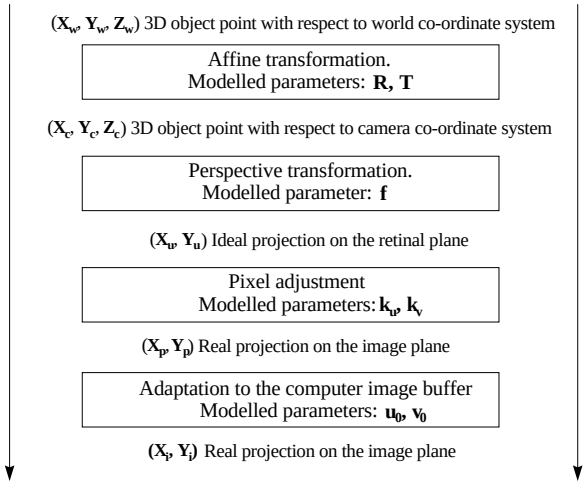

Figure 2.11: Modelling a camera without lens distortion.

second one the extrinsic parameters. So that, the intrinsic matrix is obtained substituting 2.14 into 2.19, and the result into 2.21, obtaining the following equation,

$$\begin{aligned} X_i &= -k_u f \frac{P_{Xc}}{P_{Zc}} + u_0 \\ Y_i &= -k_v f \frac{P_{Yc}}{P_{Zc}} + v_0 \end{aligned} \tag{2.23}$$

which can be expressed in a matricial way as,

$$\begin{bmatrix} sXi \\ sYi \\ s \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} P_{Xc} \\ P_{Yc} \\ P_{Zc} \\ 1 \end{bmatrix} \tag{2.24}$$

where,

$$\begin{aligned} \alpha_u &= -fk_u \\ a_v &= -fk_v \end{aligned} \tag{2.25}$$

Then, applying 2.11 in order to give the object point with respect to the world co-ordinate system, the following matricial relation is obtained,

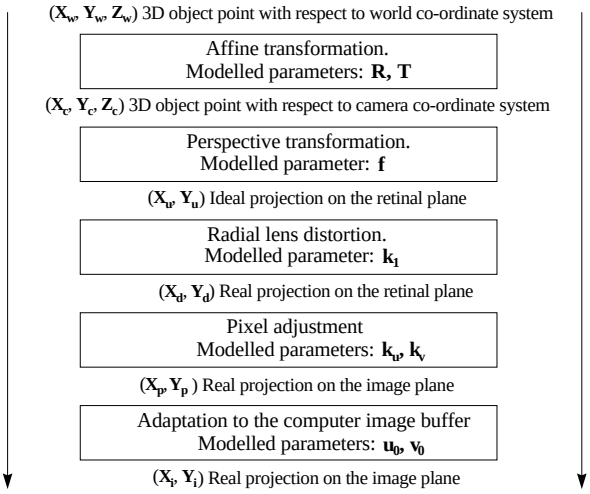

Figure 2.12: Modelling a camera with radial lens distortion.

$$\begin{bmatrix} sXi \\ sYi \\ s \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_{Xw} \\ P_{Yw} \\ P_{Zw} \\ 1 \end{bmatrix} \tag{2.26}$$

The whole camera model with radial lens distortion

In this case, a linear relation does not hold true. Actually, the equations introduced by the modelling of lens distortion does not permit the use of matrices in order to express the 2D perspective point with respect to the 3D object point. Thus, computing and arranging the equations presented in the steps of camera modelling, the following relation is obtained.

$$f\frac{P_{Xc}}{P_{Zc}} = X_d + k_1 r^2 X_d \tag{2.27}$$

$$f\frac{P_{Yc}}{P_{Zc}} = Y_d + k_1 r^2 Y_d \tag{2.28}$$

where,

$$X_d = \frac{(X_i - u_0)}{-k_u}$$

$$Y_d = \frac{(Y_i - v_0)}{-k_v} \tag{2.29}$$

and,

$$\begin{bmatrix} P_{Xc} \\ P_{Yc} \\ P_{Zc} \end{bmatrix} = \begin{bmatrix} \mathbf{R} & T \end{bmatrix} \begin{bmatrix} P_{Xw} \\ P_{Yw} \\ P_{Zw} \\ 1 \end{bmatrix} \tag{2.30}$$

Relation between back and front image plane

Some authors consider that the optical centre C is modelled in front of the image plane as shown in figure 2.13. The image plane is placed at a distance $-f$ of the optical centre, note also that both axes, U and V are inverted with respect to X_c and Y_c. All the mathematical equations introduced in order to model a camera with the image plane in front of the optical centre are also valid to model such camera configuration. However, a sign has changed both as a result of the non-inversion of the U image co-ordinate axe with respect to the X_r retinal axe, and the negative position of the image plane along the Z_c axe. Computing, the same matrix relation shown in equation 2.26 is valid, but,

$$\begin{aligned} \alpha_u &= -fk_u \\ a_v &= fk_v \end{aligned} \tag{2.31}$$

Then, placing the focal point behind or in front of the image plane will only produce an inversion of the α_v parameter.

2.3 Calibrating a camera

Calibration is based on obtaining the intrinsic and extrinsic parameters of the camera model knowing the components of a set of 3D object points and its 2D projective points. In this section, two calibrating algorithms are shown. First, a mehod based on matrix computation which permits to obtain the camera parameters without considering lens distortion. Secondly, a more generic method based on function minimisation which permits to obtain the parameters of any camera model, even considering lens distortion.

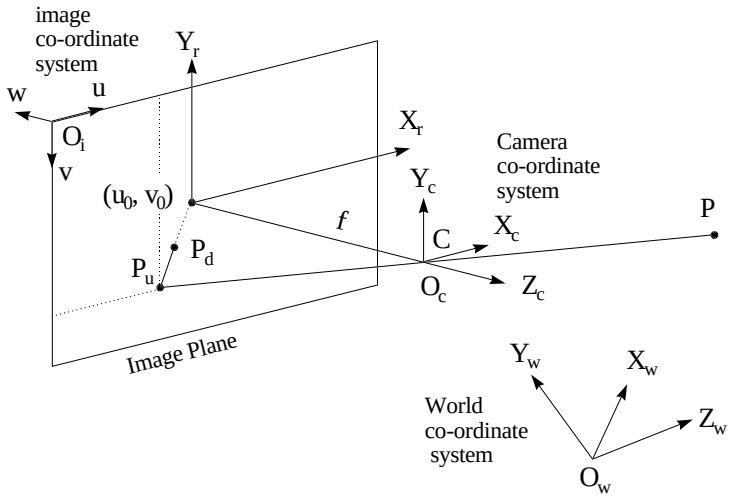

Figure 2.13: The geometric relation between a 3D object point and its 2D projection when the optical centre is located in front of the image plane.

2.3.1 Calibrating the linear model

The linear calibration method is based on obtaining the intrinsic and extrinsic parameters of a camera which has been modelled without considering lens distortion [Tosc 87]. Then, the relation between a 3D object point and its 2D projection is given by the transformation matrix of the equation 2.26. In this case, calibration is reduced to the computation of this transformation matrix, which is called A, and the further extraction of the camera parameters from the obtained matrix [Bort 91] [Bras 95].

When computing the matrix product of equation 2.26, then the following 3x4 transformation matrix is obtained. The matrix can be expressed by three orientation vectors A_i and three translation components A_{i4}.

$$A = \begin{bmatrix} \alpha_u r_1 + u_0 r_3 & \alpha_u t_x + u_0 t_z \\ \alpha_v r_2 + v_0 r_3 & \alpha_v t_y + v_0 t_z \\ r_3 & t_z \end{bmatrix} = \begin{bmatrix} A_1 & A_{14} \\ A_2 & A_{24} \\ A_3 & A_{34} \end{bmatrix} \tag{2.32}$$

Note that,

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} A_1 & A_{14} \\ A_2 & A_{24} \\ A_3 & A_{34} \end{bmatrix} \begin{bmatrix} P \\ 1 \end{bmatrix} \tag{2.33}$$

where $P = (P_{Xc}, P_{Yc}, P_{Zc})^t$ is a 3D object point with respect to the world co-ordinate system, expressed in mm., and (u, v) is its 2D projective point on the image plane expressed in pixels.

Knowing that the orientation vectors r_i are unitarian and orthogonal, and knowing that the scalar product of two vectors is equal to the multiplication of their norms with the cosinus of the angle between them, as

$$v_1 v_2 = \|v_1\| \, \|v_2\| \cos\alpha \tag{2.34}$$

where the norm of a vector is computed by the following equation.

$$\|v\| = \left(v_x^2 + v_y^2 + v_{z2}\right)^{1/2} \tag{2.35}$$

Furthermore, it can be affirmed that the orientation vectors keep the following relation,

$$\begin{aligned} r_i r_j^t &= 0 \qquad i \neq j \\ r_i r_j^t &= 1 \qquad i = j \end{aligned}$$

Then, the four intrinsic parameters of the camera (α_u, α_v, u_0, v_0), and the six extrinsic parameters (r_1,r_2, r_3, t_x, t_y, t_z), can be obtained from 2.32, as,

$$\begin{array}{ll} u_0 = A_1 A_3^t & v_0 = A_2 A_3^t \\ \alpha_u = \left(A_1 A_1^t - u_0^2\right)^{1/2} & \alpha_v = \left(A_2 A_2^t - v_0^2\right)^{1/2} \\ r_1 = \frac{1}{\alpha_u}\left(A_1 - u_0 A_3\right) & t_x = \frac{1}{\alpha_u}\left(A_{14} - u_0 A_{34}\right) \\ r_2 = \frac{1}{\alpha_v}\left(A_2 - v_0 A_3\right) & t_y = \frac{1}{\alpha_v}\left(A_{24} - v_0 A_{34}\right) \\ r_3 = A_3 & t_z = A_{34} \end{array} \tag{2.36}$$

So, it has been proved that, given the transformation matrix A, which relates the computer image co-ordinate system, with respect to the world co-ordinate system, the intrinsic and extrinsic parameters can be obtained. Then, we have to explain the method used to obtain the transformation matrix A. 2.33 must be computed in order to express the relation by linear equations,

$$A_1 P + A_{14} - u(A_3 P + A_{34}) = 0 \tag{2.37}$$

and,

$$A_2 P + A_{24} - v(A_3 P + A_{34}) = 0 \tag{2.38}$$

These equations may be arranged in the following way,

$$u = \frac{A_1}{A_{34}}P + \frac{A_{14}}{A_{34}} - \frac{A_3}{A_{34}}Pu \tag{2.39}$$

and

$$v = \frac{A_2}{A_{34}}P + \frac{A_{24}}{A_{34}} - \frac{A_3}{A_{34}}Pv \tag{2.40}$$

which can be related as a clear simplification by,

$$u = T_1 P + C_1 - T_2 Pu \tag{2.41}$$

and

$$v = T_3 P + C_2 - T_2 Pv \tag{2.42}$$

Where, from the equations 2.36, 2.41 and 2.42, the following equations are obtained,

$$\begin{array}{ll} T_1 = \frac{r_3}{t_z}u_0 + \frac{r_1}{t_z}\alpha_u & C_1 = u_0 + \frac{t_x}{t_z}\alpha_u \\ T_2 = \frac{r_3}{t_z} & \\ T_3 = \frac{r_3}{t_z}v_0 + \frac{r_2}{t_z}\alpha_v & C_2 = v_0 + \frac{t_y}{t_z}\alpha_v \end{array} \tag{2.43}$$

Where each T_1, T_2 and T_3 are three vectors of three components. And C_1 and C_2 are two single values. So, we can define a new vector X, of eleven components.

$$X = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ C_1 \\ C_2 \end{bmatrix} \tag{2.44}$$

In order to infer the X vector, it is necessary to obtain a set of couples of points. Each couple is composed by a metric 3D object point referred to a world co-ordinate system, and its 2D projective point on the image plane, expressed in pixels. The set of points has to be representative, i.e. must present a discrepancy among the 3D object points along the three axis of the world co-ordinate system, that is they have to be non-coplanar. It has been demonstrated theorically that at least six non-coplanar points are needed in order to infer X accurately. That's because six points describes twelve equations, and eleven equations is the minimum number of equations that allow us to obtain the unknowns of X.

Consider the set of point couples as $S = \cup_{i=0}^{n} (P_i^t, (u_i, v_i))$. From the relation introduced in equation 2.41 and 2.42, two new matrices $\mathbf{Q}$ and $\mathbf{B}$ can be defined as,

$$\mathbf{B} = \mathbf{Q}X \tag{2.45}$$

then,

$$\mathbf{Q} = \begin{bmatrix} & & \dots & & \\ P_i^t & -u_i P_i^t & 0_{1x3} & 1 & 0 \\ 0_{1x3} & -v_i P_i^t & P_i^t & 0 & 1 \\ & & \dots & & \end{bmatrix} \tag{2.46}$$

and,

$$\mathbf{B} = \begin{bmatrix} \dots \\ u_i \\ v_i \\ \dots \end{bmatrix} \tag{2.47}$$

Then, the X components can be determined using the following equation,

$$X = \left(\mathbf{Q}^t\mathbf{Q}\right)^{-1}\mathbf{Q}^t\mathbf{B} \tag{2.48}$$

To obtain the intrinsic and extrinsic parameters from the values of the X components, we can use the orientation vector characteristics, of which it is known that they are unitarian, i.e. their norm is equal to 1. Then, from 2.43 the extrinsic t_z. parameter can be obtained. So that, if,

$$\|r_3\| = 1 \tag{2.49}$$

then,

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

and, the other parameters can be obtained using the relationship between two orientation vectors in respect of the scalar and the vector product. So, since it is known that,

$$\begin{aligned} v_1 v_2 &= \|v_1\| \, \|v_2\| \cos\alpha \\ v_1 \wedge v_2 &= \|v_1\| \, \|v_2\| \sin\alpha \end{aligned} \tag{2.50}$$

and

$$\|v\| = \left(v_x^2 + v_y^2 + v_z^2\right)^{1/2} \tag{2.51}$$

then,

$$\begin{aligned} r_i r_j^t &= 0 \qquad i \neq j \\ r_i r_j^t &= 1 \qquad i = j \\ r_i \wedge r_j &= 1 \qquad i \neq j \\ r_i \wedge r_j &= 0 \qquad i = j \end{aligned}$$

The intrinsic parameters can be obtained from,

$$u_0 = \frac{T_1 T_2^t}{\|T_2\|^2} \tag{2.52}$$

$$v_0 = \frac{T_2 T_3^t}{\|T_2\|^2} \tag{2.53}$$

$$\alpha_u = \frac{\|T_1^t \wedge T_2^t\|}{\|T_2\|^2} \tag{2.54}$$

$$\alpha_v = \frac{\|T_2^t \wedge T_3^t\|}{\|T_2\|^2} \tag{2.55}$$

and the extrinsic ones,

$$r_1 = \frac{\|T_2\|}{\|T_1^t \wedge T_2^t\|}\left(T_1 - \frac{T_1 T_2^t}{\|T_2\|^2} T_2\right) \tag{2.56}$$

$$r_2 = \frac{\|T_2\|}{\|T_2^t \wedge T_3^t\|}\left(T_3 - \frac{T_2 T_3^t}{\|T_2\|^2} T_2\right) \tag{2.57}$$

$$r_3 = \frac{T_2}{\|T_2\|} \tag{2.58}$$

and

$$t_x = \frac{\|T_2\|}{\|T_1^t \wedge T_2^t\|}\left(C_1 - \frac{T_1 T_2^t}{\|T_2\|^2}\right) \tag{2.59}$$

$$t_y = \frac{\|T_2\|}{\|T_2^t \wedge T_3^t\|}\left(C_2 - \frac{T_2 T_3^t}{\|T_2\|^2}\right) \tag{2.60}$$

$$t_z = \frac{1}{\|T_2\|} \tag{2.61}$$

Note that we can extract the Euler rotation angles applying the equation 2.7 introduced in the section dedicated to the extrinsic parameters. Then applying the following egalisation,

$$\mathbf{R} = \begin{bmatrix} c\phi c\psi - s\phi c\theta s\psi & -c\phi s\psi - s\phi c\theta c\psi & s\phi s\theta \\ c\phi c\psi + c\phi c\theta s\psi & -s\phi s\psi + c\phi c\theta c\psi & -c\phi s\theta \\ s\theta s\psi & s\theta c\psi & c\theta \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{2.62}$$

the ϕ, θ and ψ angles can be computed applying,

$$\phi = -\arctan\left(\frac{r_{13}}{r_{23}}\right) \tag{2.63}$$

$$\theta = \arccos\left(r_{33}\right) \tag{2.64}$$

$$\psi = \arctan\left(\frac{r_{31}}{r_{32}}\right) \tag{2.65}$$

Algorithm of camera calibration

In the following, the whole algorithm used in the calibration of a linear camera model is presented. The algorithm has been programmed using the Matlab® environment.

```
clear all; close all;
fid = fopen('set_of_points.prn','r');
P = fscanf(fid,'%f',[5,num_of_couples]);
fclose(fid);
npoints = size(P,2);
P2D = P(1:2,:); P3D = P(3:5,:);
xk = P3D(1,:); yk = P3D(2,:); zk = P3D(3,:);
uk = P2D(1,:); vk = P2D(2,:);
A = []; B = [];
for i =1:npoints,
  A = [A ; xk(i) yk(i) zk(i) -uk(i)*xk(i) -uk(i)*yk(i) -uk(i)*zk(i) 0 0 0 1 0;
    0 0 0 -vk(i)*xk(i) -vk(i)*yk(i) -vk(i)*zk(i) xk(i) yk(i) zk(i) 0 1];
  B = [B ; uk(i); vk(i)];
end
X = (inv(A'*A))*A'*B;
X1 = X(1:3,1)'; X2 = X(4:6,1)'; X3 = X(7:9,1)';
C1 = X(10,1); C2 = X(11,1);
NormeX2 = (X2(1,1) * X2(1,1)) + (X2(1,2)*X2(1,2)) + (X2(1,3)*X2(1,3) );
```

```
PV = cross(X1(1,:)',X2(1,:)');
X1TX2T = sum(PV(1)^2+PV(2)^2+PV(3)^2);
PV = cross(X2(1,:)',X3(1,:)');
X2TX3T = sum(PV(1)^2+PV(2)^2+PV(3)^2);
u0 = (X1 * X2') / NormeX2; v0 = (X2 * X3') / NormeX2;
au = sqrt(X1TX2T) /NormeX2; av = sqrt(X2TX3T) /NormeX2;
r1 = (sqrt(NormeX2)/sqrt(X1TX2T)) * (X1-((X1*X2')/NormeX2)*X2);
r2 = (sqrt(NormeX2)/sqrt(X2TX3T)) * (X3-((X2*X3')/NormeX2)*X2);
r3 = X2/sqrt(NormeX2);
tx = (sqrt(NormeX2)/sqrt(X1TX2T)) * (C1-((X1*X2')/NormeX2));
ty = (sqrt(NormeX2)/sqrt(X2TX3T)) * (C2-((X2*X3')/NormeX2));
tz = 1/sqrt(NormeX2);
T1 = [ au 0 u0 0; 0 av v0 0; 0 0 1 0; 0 0 0 1];
T2 = [ r1 tx; r2 ty; r3 tz; 0 0 0 1];
```

2.3.2 Calibrating the non-linear model

The non-linear calibration of a camera is based on the general principle of function minimisation which has been widely used in order to obtain a solution of a non-linear function. Then, this principle can also be used to obtain the intrinsic and extrinsic parameters of any camera model, even if lens distortion has been considered. The relation between the 3D object points and its 2D projective point is a non-linear relation which points us to use an iterative method in order to reduce the discrepancy between the observed 2D projection and the modelled one, iteration by iteration.

The Iterative Approximation Method

The Iterative Approximation Method is an ampliation to n unknowns of the method of Newton-Raphson to obtain the roots of a function. It has been widely used over the past decades and is the most widely used standard optimisation technique. The unknown coefficients are first calculated using an approximate initial guess of the camera parameter values. Then, the function is evaluated and the ideal components are obtained. Differences between the ideal and the real parameters are estimated as the error function. The calibration procedure is the iterative optimisation of camera parameters by minimising the error function.

Hereafter will follow a description of the method, and how it has been oriented to the determination of the camera parameters. If the reader is interested in the mathematical principle of the method, he is invited to read books

on Numerical Analysis [Stoe 80]. However, the general principle is the following: Consider a funtion G which depends on m unknowns $\{x_1, x_2, ..., x_m\}$. Consider that we want to obtain the values of $\{x_1, x_2, ..., x_m\}$ which solve the equation $G(x_1, x_2, ..., x_m) = 0$ iterating from an initial solution quite close to the right unknown values. Then, from a determined iteration we can approximate the next unknowns from the last ones, using equation 2.69.

We know that a closer solution $G(X_k)$ of $G(X)$ can be approximated from $G(X_{k-1})$ by 2.66. We know also that the solution we want to find is the vector X of unknowns which solve 2.67.

$$G(X_k) \approx G(X_{k-1}) + J(X_{k-1})\,\Delta X_k \tag{2.66}$$
$$G(X_k) = 0 \tag{2.67}$$

So,

$$G(X_{k-1}) + J(X_{k-1})\,\Delta X_k = 0 \tag{2.68}$$

Then, 2.69 must be iterated until the obtained parameters are smaller than a fixed value ε, as shows 2.70.

$$\Delta X_k = -J^{-1}(X_{k-1})\,G(X_{k-1}) \tag{2.69}$$
$$\Delta X_k < \varepsilon \tag{2.70}$$

To adapt equation 2.69 to the problem of minimising an equation G of m unknowns $\{x_1, x_2, ..., x_m\}$,. we suppose having obtained n couples of points $S = \cup_{i=0}^{n} (P_i^t, (u_i, v_i))$, where P_i is a 3D object point and (u_i, v_i) is its projective point on the camera image plane. Then the vector of unknowns $\{x_1, x_2, ..., x_m\}$, i.e. the camera parameters, can be obtained in the following way.

Define $G_i(X_{k-1})$ as the evaluation of function G using the unknowns estimated in the $k-1$ iteration and the point correspondence $(P_i^t, (u_i, v_i))$. Define vector $G(X_{k-1})$ as it is related in equation 2.71.

$$G(X_{k-1}) = (G_1(X_{k-1}), G_2(X_{k-1}), ..., G_n(X_{k-1}))^t \tag{2.71}$$

Then, we can define matrix $J(G(X_{k-1}))$ as a matrix of partial differentials of $G(X)$ evaluated with the values of vector X in the iteration $k-1$, as shown in 2.72,

$$J(G(X_{k-1})) = \begin{pmatrix} J(G_1(X_{k-1})) \\ J(G_2(X_{k-1})) \\ ... \\ J(G_n(X_{k-1})) \end{pmatrix} \tag{2.72}$$

where,

$$J(G_i(X_{k-1})) = \left(\frac{\partial G_i(X_{k-1})}{\partial x_1}, \frac{\partial G_i(X_{k-1})}{\partial x_2}, ..., \frac{\partial G_i(X_{k-1})}{\partial x_m} \right) \tag{2.73}$$

and X is the vector of unknowns as shows 2.74.

$$X = (x_1, x_2, ..., x_m) \tag{2.74}$$

then,

$$\Delta X = (\Delta x_1, \Delta x_2, ..., \Delta x_m) \tag{2.75}$$

So, equation 2.68 can be expressed as 2.76.

$$\begin{pmatrix} G_1(X_{k-1}) \\ G_2(X_{k-1}) \\ ... \\ G_n(X_{k-1}) \end{pmatrix} + \begin{pmatrix} \frac{\partial G_1(X_{k-1})}{\partial x_1} & \frac{\partial G_1(X_{k-1})}{\partial x_2} & ... & \frac{\partial G_1(X_{k-1})}{\partial x_m} \\ \frac{\partial G_2(X_{k-1})}{\partial x_1} & ... & ... & ... \\ ... & ... & ... & ... \\ \frac{\partial G_n(X_{k-1})}{\partial x_1} & ... & ... & \frac{\partial G_n(X_{k-1})}{\partial x_m} \end{pmatrix} \begin{pmatrix} \Delta x_1 \\ \Delta x_2 \\ ... \\ \Delta x_m \end{pmatrix} = 0_{nx1} \tag{2.76}$$

Then, we can solve each iteration arranging equation 2.76 in the form of equation 2.69. We will iterate until $\Delta X_k < \varepsilon$, then, we will have the vector of unknowns which solve G. So, if the equation G is the camera model which permits to obtain a 2D projective point on the image plane from a given 3D object point, the camera parameters of such a model can be computed.

Note that equation 2.69 must be solved using the pseudo-inverse matrix as it does not have the same number of rows and columns, so equation 2.69 can be solved using equation 2.77.

$$\Delta X_k = -\left(J^t(X_{k-1})J(X_{k-1})\right)^{-1} J^t(X_{k-1})G(X_{k-1}) \tag{2.77}$$

The reader is referred to the next section for an introduction of the Modified Newton-Raphson Method, which imposes a reduction of the error at each iteration.

The modified Newton-Raphson method

The Newton-Raphson theorem guarantees the convergence of a given function to a local solution only if the starting solution X_0 is chosen sufficiently close to the desired solution. Then, the Newton-Raphson method may diverge in some cases. The modified Newton Raphson Method guarantees always

a reduction of the error in each iteration, i.e. a convergence of the method [Stoe 80]. However, the reader should note that the method may converge to an undesired solution, even having a large error.

From the equation 2.69 of the Newton-Raphson method, we know that the error of each iteration should be computed using equation 2.78,

$$\Delta X_k = -J^{-1}\left(X_{k-1}\right) G(X_{k-1}) \tag{2.78}$$

However, we have already said that equation 2.78 is not always satisfied, i.e. the error does not reduce in each iteration, then, equation 2.79 is not always satisfied.

$$\Delta X_k < \Delta X_{k-1} \tag{2.79}$$

The modified Newton-Rapson method proposes to compute the error using the following equation,

$$\Delta X_k = -2^{-j} J^{-1}\left(X_{k-1}\right) G(X_{k-1}) \tag{2.80}$$

then,

$$X_k = X_{k-1} - 2^{-j} J^{-1}\left(X_{k-1}\right) G(X_{k-1}) \tag{2.81}$$

Consider the following substitution to simplify the computation:

$$d_k = J^{-1}\left(X_{k-1}\right) G(X_{k-1}) \tag{2.82}$$

Then, the computation of the Modified Newton-Rapson method is reduced to compute the j variable, i.e. it is reduced to obtain the minimum ponderation of the Jacobian matrix which reduces the error in each iteration,

$$j = min\left\{i \geq 0 \mid \left\|G(X_k - 2^{-i} d_k)\right\| < \|G(X_k)\|\right\} \tag{2.83}$$

where,

$$\|G(X_k)\| = \sqrt{\sum_m x_i^2} \approx \sum_m x_i^2 \tag{2.84}$$

The Modified Newton-Raphsons algorithm is the following,

1. Fix an initial solution X_0.
2. Iterate,
   - *2.1.* Compute $G(X_k)$ and $J(X_k)$.
   - *2.2.* Compute d_k.
   - *2.3.* Iterate i from 0 up to $\|G(X_k - 2^{-i} d_k)\| < \|G(X_k)\|$.

2.4. Compute $\Delta X_k = -2^{-j} J^{-1}(X_{k-1})\, G(X_{k-1})$.
2.5. Compute $X_k = X_{k-1} + \Delta X_k$.
3. until $\Delta X_k < \varepsilon$ is reached.

Algorithm of camera calibration

In the following, the entire algorithm used in the calibration of any camera model is presented. The algorithm has been programmed using the Maple V release 4® environment. The example sets forth the programming of a camera model considering the first coefficient of lens distortion. The initial solution could be obtained applying the method described in the section which describes the calibration of a linear camera model and considering no lens distortion, i.e. $k_1 = 0$. The focal distance has been fixed, otherwise the system is unstable because several combinations of (f, k_u, k_v) can be obtained without changing the (a_u, a_v) values. Actually, the algoritm can be easily modified to obtain only the (a_u, a_v) parameters.

```
> with(linalg):
> ##############################
> # Obtaining the camera model
> ##############################
> Rx:=array([[1,0,0,0],[0,cos(t1),sin(t1),0],[0,-sin(t1),cos(t1),0],[0,0,0,1]]);
> Ry:=array([[cos(t2),0,-sin(t2),0],[0,1,0,0],[sin(t2),0,cos(t2),0],[0,0,0,1]]);
> Rz:=array([[cos(t3),sin(t3),0,0],[-sin(t3),cos(t3),0,0],[0,0,1,0],[0,0,0,1]]);
> T:=array([[1,0,0,Tx],[0,1,0,Ty],[0,0,1,Tz],[0,0,0,1]]);
> R:=multiply(multiply(Rz,Ry),Rx);
> AA:=multiply(T,R);
> M:=array([[xr],[yr],[zr],[1]]);
> G:=multiply(AA,M);
> X:=G[1,1]; Y:=G[2,1]; Z:=G[3,1];
> Xu:=X*f/Z;
> Yu:=Y*f/Z;
> Xd:=(uk-u0)/ku; Yd:=(vk-v0)/kv;
> U:=Xu-Xd-k1*(Xd*Xd+Yd*Yd)*Xd;
> V:=Yu-Yd-k1*(Xd*Xd+Yd*Yd)*Yd;
> ###################################
> # Obtaining the partial derivative
> ###################################
> dUt1:=diff(U,t1); dVt1:=diff(V,t1);
> dUt2:=diff(U,t2); dVt2:=diff(V,t2);
> dUt3:=diff(U,t3); dVt3:=diff(V,t3);
```

```
> dUTx:=diff(U,Tx); dVTx:=diff(V,Tx);
> dUTy:=diff(U,Ty); dVTy:=diff(V,Ty);
> dUTz:=diff(U,Tz); dVTz:=diff(V,Tz);
> dUf:=diff(U,f); dVf:=diff(V,f);
> dUku:=diff(U,ku); dVku:=diff(V,ku);
> dUkv:=diff(U,kv); dVkv:=diff(V,kv);
> dUu0:=diff(U,u0); dVu0:=diff(V,u0);
> dUv0:=diff(U,v0); dVv0:=diff(V,v0);
> dUk1:=diff(U,k1); dVk1:=diff(V,k1);
> ####################################
> # Reading the set of points couples
> ####################################
> # Reading the M Matrix from an ASCII file.
> readlib(readdata);
> Pp:=readdata(set_of_couples,5);
> P2D:=matrix(rowdim(Pp),2,0):P3D:=matrix(rowdim(Pp),3,0):
> for i from 1 by 1 to rowdim(Pp) do
> P2D[i,1]:=op(1,op(i,Pp)): P2D[i,2]:=op(2,op(i,Pp)):
> P3D[i,1]:=op(3,op(i,Pp)): P3D[i,2]:=op(4,op(i,Pp)): P3D[i,3]:=op(5,op(i,Pp)):
> od:
> ##################
> # Initial solution
> ##################
> PI:=3.141592654;
> vt1:=initial_value_t1;
> vt2:=initial_value_t2;
> vt3:=initial_value_t3;
> vTx:=initial_value_Tx;
> vTy:=initial_value_Ty;
> vTz:=initial_value_Tz;
> vu0:=initial_value_u0;
> vv0:=initial_value_v0;
> vk1:=0;
> vau:=initial_value_au;
> vav:=initial_value_av;
> vf:=fixed_value; vku:=vau/vf; vkv:=vav/vf;
> ############################
> # Minimizing by iterations
> ############################
> VG:=vector(2*rowdim(Pp),0):
> J:=matrix(2*rowdim(Pp),11,0): VI:=vector(11,0):
```

> for j from 1 by 1 while j <= 20 do
> for i from 1 by 1 while i <= rowdim(Pp) do
> vuk:=P2D[i,1]; vvk:=P2D[i,2];
> vxr:=P3D[i,1]; vyr:=P3D[i,2]; vzr:=P3D[i,3];
> VG[i*2-1]:=simplify(subs(xr=vxr, yr=vyr, zr=vzr, t1=vt1, t2=vt2, t3=vt3, Tx=vTx, Ty=vTy, Tz=vTz, f=vf, ku=vku, kv=vkv, u0=vu0, v0=vv0, k1=vk1, uk=vuk, vk=vvk, U));
> VG[i*2]:=simplify(subs(xr=vxr, yr=vyr, zr=vzr, t1=vt1, t2=vt2, t3=vt3, Tx=vTx, Ty=vTy, Tz=vTz, f=vf, ku=vku, kv=vkv, u0=vu0, v0=vv0, k1=vk1, uk=vuk, vk=vvk, V));
> J[i*2-1,1]:=simplify(subs(xr=vxr, yr=vyr, zr=vzr, t1=vt1, t2=vt2, t3=vt3, Tx=vTx, Ty=vTy, Tz=vTz, f=vf, ku=vku, kv=vkv, u0=vu0, v0=vv0, k1=vk1, uk=vuk, vk=vvk,dUt1)); J[i*2,1]:=simplify(subs(xr=vxr, yr=vyr, zr=vzr, t1=vt1, t2=vt2, t3=vt3, Tx=vTx, Ty=vTy, Tz=vTz, f=vf, ku=vku, kv=vkv, u0=vu0, v0=vv0, k1=vk1, uk=vuk, vk=vvk, dVt1));
> J[i*2-1,2]:=simplify(subs(xr=vxr, yr=vyr, zr=vzr, t1=vt1, t2=vt2, t3=vt3, Tx=vTx, Ty=vTy, Tz=vTz, f=vf, ku=vku, kv=vkv, u0=vu0, v0=vv0, k1=vk1, uk=vuk, vk=vvk,dUt2)); J[i*2,2]:=simplify(subs(xr=vxr, yr=vyr, zr=vzr, t1=vt1, t2=vt2, t3=vt3, Tx=vTx, Ty=vTy, Tz=vTz, f=vf, ku=vku, kv=vkv, u0=vu0, v0=vv0, k1=vk1,uk=vuk, vk=vvk, dVt2));
> J[i*2-1,3]:=simplify(subs(xr=vxr, yr=vyr, zr=vzr, t1=vt1, t2=vt2, t3=vt3, Tx=vTx, Ty=vTy, Tz=vTz, f=vf, ku=vku, kv=vkv, u0=vu0, v0=vv0, k1=vk1, uk=vuk, vk=vvk,dUt3)); J[i*2,3]:=simplify(subs(xr=vxr, yr=vyr, zr=vzr, t1=vt1, t2=vt2, t3=vt3, Tx=vTx, Ty=vTy, Tz=vTz, f=vf, ku=vku, kv=vkv, u0=vu0, v0=vv0, k1=vk1,uk=vuk, vk=vvk,dVt3));
> J[i*2-1,4]:=simplify(subs(xr=vxr, yr=vyr, zr=vzr, t1=vt1, t2=vt2, t3=vt3, Tx=vTx, Ty=vTy, Tz=vTz, f=vf, ku=vku, kv=vkv, u0=vu0, v0=vv0, k1=vk1, uk=vuk, vk=vvk,dUTx)); J[i*2,4]:=simplify(subs(xr=vxr, yr=vyr, zr=vzr, t1=vt1, t2=vt2, t3=vt3, Tx=vTx, Ty=vTy, Tz=vTz, f=vf, ku=vku, kv=vkv, u0=vu0, v0=vv0, k1=vk1,uk=vuk, vk=vvk,dVTx));
> J[i*2-1,5]:=simplify(subs(xr=vxr, yr=vyr, zr=vzr, t1=vt1, t2=vt2, t3=vt3, Tx=vTx, Ty=vTy, Tz=vTz, f=vf, ku=vku, kv=vkv, u0=vu0, v0=vv0, k1=vk1, uk=vuk, vk=vvk,dUTy)); J[i*2,5]:=simplify(subs(xr=vxr, yr=vyr, zr=vzr, t1=vt1, t2=vt2, t3=vt3, Tx=vTx, Ty=vTy, Tz=vTz, f=vf, ku=vku, kv=vkv, u0=vu0, v0=vv0, k1=vk1,uk=vuk, vk=vvk,dVTy));
> J[i*2-1,6]:=simplify(subs(xr=vxr, yr=vyr, zr=vzr, t1=vt1, t2=vt2, t3=vt3, Tx=vTx, Ty=vTy, Tz=vTz, f=vf, ku=vku, kv=vkv, u0=vu0, v0=vv0, k1=vk1, uk=vuk, vk=vvk,dUTz)); J[i*2,6]:=simplify(subs(xr=vxr, yr=vyr, zr=vzr, t1=vt1, t2=vt2, t3=vt3, Tx=vTx, Ty=vTy, Tz=vTz, f=vf, ku=vku, kv=vkv, u0=vu0, v0=vv0, k1=vk1,uk=vuk, vk=vvk,dVTz));

> J[i*2-1,7]:=simplify(subs(xr=vxr, yr=vyr, zr=vzr, t1=vt1, t2=vt2, t3=vt3, Tx=vTx, Ty=vTy, Tz=vTz, f=vf, ku=vku, kv=vkv, u0=vu0, v0=vv0, k1=vk1, uk=vuk, vk=vvk,dUku)); J[i*2,7]:=simplify(subs(xr=vxr, yr=vyr, zr=vzr, t1=vt1, t2=vt2, t3=vt3, Tx=vTx, Ty=vTy, Tz=vTz, f=vf, ku=vku, kv=vkv, u0=vu0, v0=vv0, k1=vk1,uk=vuk, vk=vvk,dVku));

> J[i*2-1,8]:=simplify(subs(xr=vxr, yr=vyr, zr=vzr, t1=vt1, t2=vt2, t3=vt3, Tx=vTx, Ty=vTy, Tz=vTz, f=vf, ku=vku, kv=vkv, u0=vu0, v0=vv0, k1=vk1, uk=vuk, vk=vvk,dUkv)); J[i*2,8]:=simplify(subs(xr=vxr, yr=vyr, zr=vzr, t1=vt1, t2=vt2, t3=vt3, Tx=vTx, Ty=vTy, Tz=vTz, f=vf, ku=vku, kv=vkv, u0=vu0, v0=vv0, k1=vk1,uk=vuk, vk=vvk,dVkv));

> J[i*2-1,9]:=simplify(subs(xr=vxr, yr=vyr, zr=vzr, t1=vt1, t2=vt2, t3=vt3, Tx=vTx, Ty=vTy, Tz=vTz, f=vf, ku=vku, kv=vkv, u0=vu0, v0=vv0, k1=vk1, uk=vuk, vk=vvk,dUu0)); J[i*2,9]:=simplify(subs(xr=vxr, yr=vyr, zr=vzr, t1=vt1, t2=vt2, t3=vt3, Tx=vTx, Ty=vTy, Tz=vTz, f=vf, ku=vku, kv=vkv, u0=vu0, v0=vv0, k1=vk1,uk=vuk, vk=vvk,dVu0));

> J[i*2-1,10]:=simplify(subs(xr=vxr, yr=vyr, zr=vzr, t1=vt1, t2=vt2, t3=vt3, Tx=vTx, Ty=vTy, Tz=vTz, f=vf, ku=vku, kv=vkv, u0=vu0, v0=vv0, k1=vk1, uk=vuk, vk=vvk,dUv0)); J[i*2,10]:=simplify(subs(xr=vxr, yr=vyr, zr=vzr, t1=vt1, t2=vt2, t3=vt3, Tx=vTx, Ty=vTy, Tz=vTz, f=vf, ku=vku, kv=vkv, u0=vu0, v0=vv0, k1=vk1,uk=vuk, vk=vvk,dVv0));

> J[i*2-1,11]:=simplify(subs(xr=vxr, yr=vyr, zr=vzr, t1=vt1, t2=vt2, t3=vt3, Tx=vTx, Ty=vTy, Tz=vTz, f=vf, ku=vku, kv=vkv, u0=vu0, v0=vv0, k1=vk1, uk=vuk, vk=vvk,dUk1)); J[i*2,11]:=simplify(subs(xr=vxr, yr=vyr, zr=vzr, t1=vt1, t2=vt2, t3=vt3, Tx=vTx, Ty=vTy, Tz=vTz, f=vf, ku=vku, kv=vkv, u0=vu0, v0=vv0, k1=vk1,uk=vuk, vk=vvk,dVk1));

> od;

> VI :=evalm(-1.0 * multiply(multiply(inverse(multiply(transpose(J),J)), transpose(J)),VG));

> vt1:=vt1+VI[1]; vt2:=vt2+VI[2]; vt3:=vt3+VI[3];

> vTx:=vTx+VI[4]; vTy:=vTy+VI[5]; vTz:=vTz+VI[6];

> vku:=vku+VI[7]; vkv:=vkv+VI[8];

> vu0:=vu0+VI[9]; vv0:=vv0+VI[10];

> vk1:=vk1+VI[11];

> od;

2.4 3D Reconstruction

Since our 3D imaging system is based on a camera sensor and a pattern projector system, two different sets of parameters have to be computed by calibration. Once both systems have been calibrated, it is possible to obtain

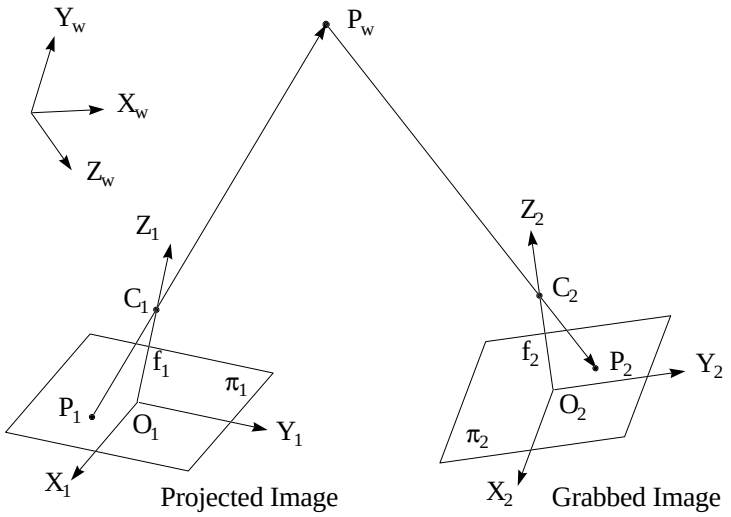

Figure 2.14: The principle of triangulation used in order to obtain the 3D scene information.

a 3D reconstruction of the measuring scene. So that, given the co-ordinates of the projective point on the image plane P_2 and given the co-ordinates of the 2D position from where this point has been projected on the scene P_1, the 3D co-ordinates of the object point where the projecting point has been reflected on the image sensor P_w can be computed by triangulation. Figure 2.14 illustrates this principle.

As we have explained two different calibrating methods, that is the linear and the non-linear one, we have also to explain two different methods of 3D reconstruction as a result of the method of calibration used.

2.4.1 Reconstructing from linear calibration

Given the transformation matrix, which relates an object point with its projection on the captured image and on the projector image, respectively, and the 2D co-ordinates of the two projections of the same 3D object point, the co-ordinates of this object point can be determined.

Summarising the equations obtained in the calibration process.

$$\begin{bmatrix} w_1 x_{P1} \\ w_1 x_{P1} \\ w_1 \end{bmatrix} = \begin{bmatrix} A_{111} & A_{112} & A_{113} & A_{114} \\ A_{121} & A_{122} & A_{123} & A_{124} \\ A_{131} & A_{132} & A_{133} & A_{134} \end{bmatrix} \begin{bmatrix} x_{Pw} \\ y_{Pw} \\ z_{Pw} \\ 1 \end{bmatrix} \tag{2.85}$$

and,

$$\begin{bmatrix} w_2 x_{P2} \\ w_2 x_{P2} \\ w_2 \end{bmatrix} = \begin{bmatrix} A_{211} & A_{212} & A_{213} & A_{214} \\ A_{221} & A_{222} & A_{223} & A_{224} \\ A_{231} & A_{232} & A_{233} & A_{234} \end{bmatrix} \begin{bmatrix} x_{Pw} \\ y_{Pw} \\ z_{Pw} \\ 1 \end{bmatrix} \tag{2.86}$$

Operating, and re-arranging the variables,

$$\begin{aligned} & (A_{111} - A_{131}x_{p1})\, x_{pw} + (A_{112} - A_{132}x_{p1})\, y_{pw} + (A_{113} - A_{133}x_{p1})\, z_{pw} \\ = \;& A_{134}x_{p1} - A_{114} \end{aligned} \tag{2.87}$$

$$\begin{aligned} & (A_{121} - A_{131}y_{p1})\, x_{pw} + (A_{122} - A_{132}y_{p1})\, y_{pw} + (A_{123} - A_{133}y_{p1})\, z_{pw} \\ = \;& A_{134}y_{p1} - A_{124} \end{aligned} \tag{2.88}$$

and,

$$\begin{aligned} & (A_{211} - A_{231}x_{p2})\, x_{pw} + (A_{212} - A_{232}x_{p2})\, y_{pw} + (A_{213} - A_{233}x_{p2})\, z_{pw} \\ = \;& A_{234}x_{p2} - A_{214} \end{aligned} \tag{2.89}$$

$$\begin{aligned} & (A_{221} - A_{231}y_{p2})\, x_{pw} + (A_{222} - A_{232}y_{p2})\, y_{pw} + (A_{223} - A_{233}y_{p2})\, z_{pw} \\ = \;& A_{234}y_{p2} - A_{224} \end{aligned} \tag{2.90}$$

So, arranging the 2D point co-ordinates and the object point co-ordinates in matricial form, the relation can be expressed as,

$$\mathbf{P}N = F \tag{2.91}$$

where, the matrices are,

$$\mathbf{P} = \begin{bmatrix} A_{111} - A_{131}x_{p1} & A_{112} - A_{132}x_{p1} & A_{113} - A_{133}x_{p1} \\ A_{121} - A_{131}y_{p1} & A_{122} - A_{132}y_{p1} & A_{123} - A_{133}y_{p1} \\ A_{211} - A_{231}x_{p2} & A_{212} - A_{232}x_{p2} & A_{213} - A_{233}x_{p2} \\ A_{221} - A_{231}y_{p2} & A_{222} - A_{232}y_{p2} & A_{223} - A_{233}y_{p2} \end{bmatrix} \tag{2.92}$$

$$N = \begin{bmatrix} x_{pw} \\ y_{pw} \\ z_{pw} \end{bmatrix} \tag{2.93}$$

$$F = \begin{bmatrix} A_{134}x_{p1} - A_{114} \\ A_{134}y_{p1} - A_{124} \\ A_{234}x_{p2} - A_{214} \\ A_{234}y_{p2} - A_{224} \end{bmatrix} \tag{2.94}$$

Finally, the 3D object co-ordinates can be obtained by,

$$N = \left(\mathbf{P}^t\mathbf{P}\right)^{-1}\mathbf{P}^t F \tag{2.95}$$

As the object co-ordinates depend on the correct association of the projected point (x_{p1}, y_{p1}) and the captured image (x_{p2}, y_{p2}), any mistake in the correspondence establishment leads to an error in the object point co-ordinates determination.

Algorithm of linear 3D reconstruction

```
% clear all; clf; clg; close all;
    % Camera calibrated by linear method
    % cam_cal
    % A1 = T1*T2
    % save A1.mat A1;
    % Projector calibrated by linear method
    % pro_cal2
    % A2 = T1*T2
    % save A2.mat A2;
    % 3D reconstruction from 2D projections
    load A1.mat;
    load A2.mat;
    npoints = num_of_couples;
    fid = fopen('set_of_couples.prn','r');
    P = fscanf(fid,'%d',[4,npoints]);
    fclose(fid);
    P1 = P(1:2,:);
    P2 = P(3:4,:);
    for i=1:npoints,
      P = [A1(1,1)-A1(3,1)*P1(1,i),A1(1,2)-A1(3,2)*P1(1,i),A1(1,3)-A1(3,3)*P1(1,i);
        A1(2,1)-A1(3,1)*P1(2,i),A1(2,2)-A1(3,2)*P1(2,i),A1(2,3)-A1(3,3)*P1(2,i);
        A2(1,1)-A2(3,1)*P2(1,i),A2(1,2)-A2(3,2)*P2(1,i),A2(1,3)-A2(3,3)*P2(1,i);
        A2(2,1)-A2(3,1)*P2(2,i),A2(2,2)-A2(3,2)*P2(2,i),A2(2,3)-A2(3,3)*P2(2,i);];
      F = [A1(3,4)*P1(1,i)-A1(1,4); A1(3,4)*P1(2,i)-A1(2,4); A2(3,4)*P2(1,i)-A2(1,4);
A2(3,4)*P2(2,i)-A2(2,4)];
      V(:,i) = inv(P'*P)*P'*F;
    end;
```

2.4.2 Reconstructing from non-linear calibration

From the section dedicated to camera modelling with lens distortion, the following relation has been obtained,

$$\begin{aligned} U &= \left(f\frac{P_{Xc}}{P_{Zc}} = X_d + k_1 r^2 X_d \right) \\ V &= \left(f\frac{P_{Yc}}{P_{Zc}} = Y_d + k_1 r^2 Y_d \right) \end{aligned} \tag{2.96}$$

where,

$$\begin{aligned} X_d &= \frac{(X_i - u_0)}{k_u} \\ Y_d &= \frac{(Y_i - v_0)}{k_v} \end{aligned} \tag{2.97}$$

and,

$$\begin{bmatrix} P_{Xc} \\ P_{Yc} \\ P_{Zc} \end{bmatrix} = \begin{bmatrix} \mathbf{R} & T \end{bmatrix} \begin{bmatrix} P_{Xw} \\ P_{Yw} \\ P_{Zw} \\ 1 \end{bmatrix} \tag{2.98}$$

where $\mathbf{R}$ is obtained from 2.5 and T is the translation vector $T = (t_x, t_y, t_z)$.

We have not taken into account the k_u and k_v sign as they depend on the camera image plane or projector plane position. Actually, they can be considered positive in modelling as they are further determined in calibration.

Consider that the following set of parameters S_c and S_p are determined from the camera calibration and the projector calibration, respectively.

$$S_c = \left\{ \phi_x^c, \phi_y^c, \phi_z^c, t_x^c, t_y^c, t_z^c, f^c, k_u^c, k_v^c, u_0^c, v_0^c, k_1^c \right\} \tag{2.99}$$

$$S_p = \left\{ \phi_x^p, \phi_y^p, \phi_z^p, t_x^p, t_y^p, t_z^p, f^p, k_u^p, k_v^p, u_0^p, v_0^p, k_1^p \right\} \tag{2.100}$$

Consider (X_i^c, Y_i^c) and (X_i^p, Y_i^p) the projection of the 3D object point (P_{Xw}, P_{Yw}, P_{Zw}) on the camera image plane and the projector frame, respectively. Then, substituting the parameter values of the set from the generic ones, the following equations are obtained.

$$U^c = subs\left(\begin{array}{c} U, X_i = X_i^c, Y_i = Y_i^c, \phi_x = \phi_x^c, \phi_y = \phi_y^c, \phi_z = \phi_z^c, \\ t_x = t_x^c, t_y = t_y^c, t_z = t_z^c, f = f^c, k_u = k_u^c, k_v = k_v^c, \\ u_0 = u_0^c, v_0 = v_0^c, k_1 = k_1^c \end{array}\right) \quad (2.101)$$

$$V^c = subs\left(\begin{array}{c} V, X_i = X_i^c, Y_i = Y_i^c, \phi_x = \phi_x^c, \phi_y = \phi_y^c, \phi_z = \phi_z^c, \\ t_x = t_x^c, t_y = t_y^c, t_z = t_z^c, f = f^c, k_u = k_u^c, k_v = k_v^c, \\ u_0 = u_0^c, v_0 = v_0^c, k_1 = k_1^c \end{array}\right) \quad (2.102)$$

$$U^p = subs\left(\begin{array}{c} U, X_i = X_i^p, Y_i = Y_i^p, \phi_x = \phi_x^p, \phi_y = \phi_y^p, \phi_z = \phi_z^p, \\ t_x = t_x^p, t_y = t_y^p, t_z = t_z^p, f = f^p, k_u = k_u^p, k_v = k_v^p, \\ u_0 = u_0^p, v_0 = v_0^p, k_1 = k_1^p \end{array}\right) \quad (2.103)$$

$$V^p = subs\left(\begin{array}{c} V, X_i = X_i^p, Y_i = Y_i^p, \phi_x = \phi_x^p, \phi_y = \phi_y^p, \phi_z = \phi_z^p, \\ t_x = t_x^p, t_y = t_y^p, t_z = t_z^p, f = f^p, k_u = k_u^p, k_v = k_v^p, \\ u_0 = u_0^p, v_0 = v_0^p, k_1 = k_1^p \end{array}\right) \quad (2.104)$$

These four equations permit to obtain the (P_{Xw}, P_{Yw}, P_{Zw}) unknowns using a least squares relation similar to the one described in equation 2.95.

Algorithm of non-linear 3D reconstruction

```
> with(linalg):with(linalg,leastsqrs):
  > ##########################
  > # Obtaining the equations.
  > ##########################
  > Rx:=array([[1,0,0,0],[0,cos(t1),sin(t1),0],[0,-sin(t1),cos(t1),0],[0,0,0,1]]);
  > Ry:=array([[cos(t2),0,-sin(t2),0],[0,1,0,0],[sin(t2),0,cos(t2),0],[0,0,0,1]]);
  > Rz:=array([[cos(t3),sin(t3),0,0],[-sin(t3),cos(t3),0,0],[0,0,1,0],[0,0,0,1]]);
  > T:=array([[1,0,0,Tx],[0,1,0,Ty],[0,0,1,Tz],[0,0,0,1]]);
  > R:=multiply(multiply(Rz,Ry),Rx);
  > AA:=multiply(T,R);
  > M:=array([[xr],[yr],[zr],[1]]);
  > G:=multiply(AA,M);
  > X:=G[1,1]; Y:=G[2,1]; Z:=G[3,1];
  > Xu:=X*f/Z;
  > Yu:=Y*f/Z;
  > Xd:=(uk-u0)/ku; Yd:=(vk-v0)/kv;
```

```
> U:=Xu-Xd-k1*(Xd*Xd+Yd*Yd)*Xd;
> V:=Yu-Yd-k1*(Xd*Xd+Yd*Yd)*Yd;
> UC:=subs(uk=uc,vk=vc,U);VC:=subs(uk=uc,vk=vc,V);
> UP:=subs(uk=up,vk=vp,U);VP:=subs(uk=up,vk=vp,V);
> # Camera model
> vt1:=valor_t1c;
> vt2:=valor_t2c;
> vt3:=valor_t3c;
> vTx:=valor_txc;
> vTy:=valor_tyc;
> vTz:=valor_tzc;
> vku:=valor_kuc;
> vkv:=valor_kvc;
> vu0:=valor_u0c;
> vv0:=valor_v0c;
> vf:=valor_vfc;
> vk1:=valor_k1c;
> UC:=subs(t1=vt1, t2=vt2, t3=vt3, Tx=vTx, Ty=vTy, Tz=vTz, f=vf, ku=vku,
> kv=vkv, u0=vu0, v0=vv0, k1=vk1, UC);
> VC:=subs(t1=vt1, t2=vt2, t3=vt3, Tx=vTx, Ty=vTy, Tz=vTz, f=vf, ku=vku,
> kv=vkv, u0=vu0, v0=vv0, k1=vk1, VC);
> # Projector model
> vt1:=valor_t1p;
> vt2:=valor_t2p;
> vt3:=valor_t3p;
> vTx:=valor_txp;
> vTy:=valor_typ;
> vTz:=valor_tzp;
> vku:=valor_kup;
> vkv:=valor_kvp;
> vu0:=valor_u0p;
> vv0:=valor_v0p;
> vf:=valor_vfp;
> vk1:=valor_k1p;
> UP:=subs(t1=vt1, t2=vt2, t3=vt3, Tx=vTx, Ty=vTy, Tz=vTz, f=vf, ku=vku,
> kv=vkv, u0=vu0, v0=vv0, k1=vk1, UP);
> VP:=subs(t1=vt1, t2=vt2, t3=vt3, Tx=vTx, Ty=vTy, Tz=vTz, f=vf, ku=vku,
> kv=vkv, u0=vu0, v0=vv0, k1=vk1, VP);
> ##########################
> # 3D Reconstruction
> ##########################
```

```
> # Read set of couples 2D
> readlib(readdata);
> Pp:=readdata(set_of_couples,4);
> P2DC:=matrix(rowdim(Pp),2,0):P2DP:=matrix(rowdim(Pp),2,0):
> rowdim(Pp),3,0):
> for i from 1 by 1 to rowdim(Pp) do
> P2DC[i,1]:=op(1,op(i,Pp)):P2DC[i,2]:=op(2,op(i,Pp)):
> P2DP[i,1]:=op(3,op(i,Pp)):P2DP[i,2]:=op(4,op(i,Pp)):
> od:
> P:=matrix(4,3,0): F:=matrix(4,1,0): P3DS:=matrix(rowdim(Pp),3,0):
> for i from 1 by 1 while i <= rowdim(Pp) do
> vuc:=P2DC[i,1]: vvc:=P2DC[i,2]: vup:=P2DP[i,1]: vvp:=P2DP[i,2]:
> UCS:=subs(uc=vuc, vc=vvc, UC):
> A:=convert(UCS,list):B:=convert(A[1],list):UCS:=B[2]+A[2]/B[1]*(1/B[3]):
> VCS:=subs(uc=vuc, vc=vvc, VC):
> A:=convert(VCS,list):B:=convert(A[1],list):VCS:=B[2]+A[2]/B[1]*(1/B[3]):
> UPS:=subs(up=vup, vp=vvp, UP):
> A:=convert(UPS,list):B:=convert(A[1],list):UPS:=B[2]+A[2]/B[1]*(1/B[3]):
> VPS:=subs(up=vup, vp=vvp, VP):
> A:=convert(VPS,list):B:=convert(A[1],list):VPS:=B[2]+A[2]/B[1]*(1/B[3]):
> A:=convert(UCS,list):
> P[1,1] := convert(A[1],list)[1]: P[1,2] := convert(A[2],list)[1]:
> P[1,3] := convert(A[3],list)[1]: F[1,1]:= -A[4]:
> A:=convert(VCS,list):
> P[2,1] := convert(A[1],list)[1]: P[2,2] := convert(A[2],list)[1]:
> P[2,3] := convert(A[3],list)[1]: F[2,1]:= -A[4]:
> A:=convert(UPS,list):
> P[3,1] := convert(A[1],list)[1]: P[3,2] := convert(A[2],list)[1]:
> P[3,3] := convert(A[3],list)[1]: F[3,1]:= -A[4]:
> A:=convert(VPS,list):
> P[4,1] := convert(A[1],list)[1]: P[4,2] := convert(A[2],list)[1]:
> P[4,3] := convert(A[3],list)[1]: F[4,1]:= -A[4]:
> V:=multiply(multiply(inverse(multiply(transpose(P),P)),transpose(P)),F):
> P3DS[i,1]:=V[1,1]:P3DS[i,2]:=V[2,1]:P3DS[i,3]:=V[3,1]:
> od:
> # Reconstruction obtained in P3DS[i,1..3]
```

Chapter 3

The Epipolar Geometry

In this chapter we will provide a description of the epipolar geometry of a stereoscopic system made by the relation between two pinhole models. We will also explain how this geometry is modelled through the fundamental matrix and a method to estimate this matrix. The epipolar geometry is a powerful concept which has been widely used in order to reduce the correspondence problem from a two-dimensional to a one-dimensional search.

3.1 The fundamental matrix

The epipolar geometry takes into account the relationship that exists between two pinhole models placed in the space [Zhan 96c]. Basically, the epipolar geometry models the behaviour of epipoles and epipolar lines. Furthermore, the Fundamental matrix is defined by the relation of an image point on an image plane and its epipolar line on the other plane [Luon 92]. In the following, all these concepts are described in detail.

Consider a stereoscopic system made by the relation of two pinhole models which project the same scene, as shown in figure 3.1. Each model has an associated co-ordinate system X and X', located in their focal point C and C' respectively. The intersection of the line (C, C') with each image plane, I and I' produces two epipoles, named e and e'. Consider an image point m on I, then, the epipolar geometry allows us to search its correspondence point m' on the epipolar line of m on I', i.e. on the line l'_m where it is defined by the intersection of the plane I' and the plane defined by the three points C, C' and m. Note that we do not need any three dimensional information from M to infer m'.

Hereafter both co-ordinate systems X and X' are related. Consider that X is fixed as the world co-ordinate system O. 3-D measures of a scene are

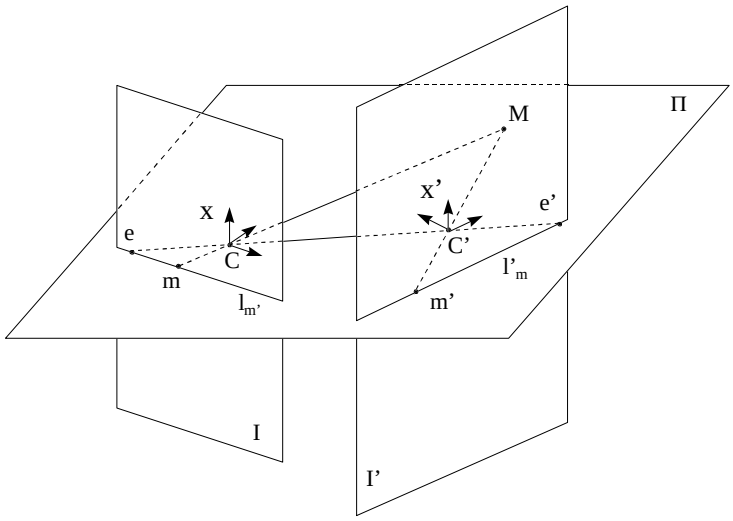

Figure 3.1: The epiplar geometry between two pinhole models.

always measured with respect to the world co-ordinate system. Then, the second co-ordinate system X' is related to X by a 3x3 rotation matrix $\mathbf{R}$ and a translation vector t.

$$X = O \qquad X = \left[\begin{array}{cc} \mathbf{I} & 0 \end{array} \right] O \tag{3.1}$$

$$X' = \left[\begin{array}{cc} \mathbf{R} & t \end{array} \right] X \qquad X' = \left[\begin{array}{cc} \mathbf{R} & t \end{array} \right] O \tag{3.2}$$

Given an object point M, it will be projected on both image planes through the focal point. Each projected point can be defined with a matrix which models the optical and internal geometry of the sensor, and a matrix which places the sensor in the space. The reader is referred to chapter two to understand what these two matrices look like. The product of both matrices determines the projection of an object point on the image plane. Consider $\mathbf{A}$ and $\mathbf{A}'$ the intrinsic matrices, and $\mathbf{B}$ and $\mathbf{B}'$ the extrinsic matrices, of sensors I and I', respectively. Then the projections of M can be expressed as,

$$sm = \mathbf{AB} \left[\begin{array}{c} M \\ 1 \end{array} \right] \tag{3.3}$$

$$s'm' = \mathbf{A}'\mathbf{B}' \left[\begin{array}{c} M \\ 1 \end{array} \right] \tag{3.4}$$

As s and s' are always constant values given by the homogeneous co-ordinates, we can express the relation with M and both projections m and m' by 3.5 and 3.6.

$$m = \widetilde{\mathbf{P}}M \tag{3.5}$$
$$m' = \widetilde{\mathbf{P}}'M \tag{3.6}$$

where,

$$\widetilde{\mathbf{P}} = \frac{1}{s}\mathbf{AB} \tag{3.7}$$
$$\widetilde{\mathbf{P}}' = \frac{1}{s'}\mathbf{A}'\mathbf{B}' \tag{3.8}$$

and,

$$m = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \qquad M = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{3.9}$$

In the following, consider the single transformation matrix $\widetilde{\mathbf{P}}$. We can give a geometrical interpretation of the rows of $\widetilde{\mathbf{P}}$ expressed in equation 3.10,

$$\widetilde{\mathbf{P}} = \begin{bmatrix} Q_1^t \\ Q_2^t \\ Q_3^t \end{bmatrix} = \begin{bmatrix} q_1^t & q_{14} \\ q_2^t & q_{24} \\ q_3^t & q_{34} \end{bmatrix} \tag{3.10}$$

where each vector Q_i^t represents a projective plane with a point equation $Q_i^t M = 0$. As can be deduced from equation 3.3, 3.5, 3.7, $Q_3^t M = 0$ corresponds to points on the image plane such as $s = 0$, that is points at infinite. That is why $Q_3^t M = 0$ is the focal plane. The other two planes $Q_1^t M = 0$ and $Q_2^t M = 0$ correspond to points on the image plane such as $u = 0$ and $v = 0$ respectively. The intersection of these two planes is the line going through the focal point C, and the origin of the co-ordinate system of the camera image plane. Note that the projection c of the focal point C does not correspond to the origin of the image co-ordinate system O_i. In fact, it is always located at a point (u_0, v_0) from O_i (see chapter two to prove this fact).

As it is shown in figure 3.2, the focal point of the camera is defined by the intersection of the three planes $Q_i^t M = 0$, i.e. of the planes $Q_1^t M = 0$, $Q_2^t M = 0$ and $Q_3^t M = 0$. Then, it can be obtained from the equation 3.11.

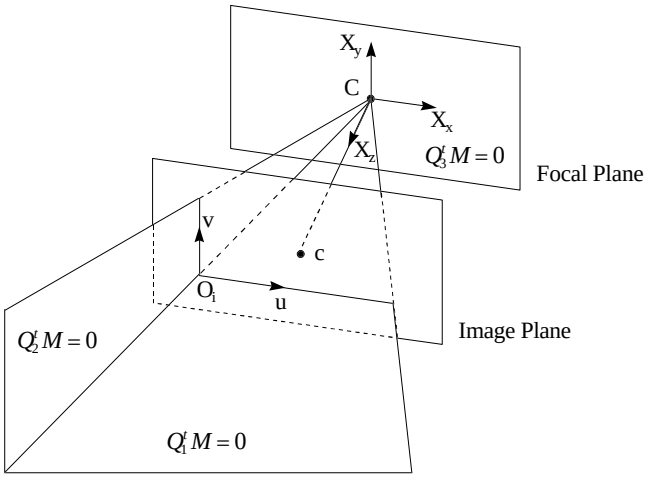

Figure 3.2: Geometrical representations of the three line vectors of the matrix $\widetilde{\mathbf{P}}$.

$$\begin{bmatrix} Q_1^t \\ Q_2^t \\ Q_3^t \end{bmatrix} \begin{bmatrix} C_x \\ C_y \\ C_z \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad \widetilde{\mathbf{P}} \begin{bmatrix} C \\ 1 \end{bmatrix} = 0 \tag{3.11}$$

Let us express the 3x4 matrix $\widetilde{\mathbf{P}}$ as described by formula 3.12,

$$\widetilde{\mathbf{P}} = \begin{bmatrix} \mathbf{P} & p \end{bmatrix} \tag{3.12}$$

where $\mathbf{P}$ is a 3x3 matrix and p is a 3x1 vector, both related by the following equations,

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix} \qquad p = \begin{bmatrix} p_{14} \\ p_{24} \\ p_{34} \end{bmatrix} \tag{3.13}$$

Then, equation 3.11 can be expressed as,

$$\begin{bmatrix} \mathbf{P} & p \end{bmatrix} \begin{bmatrix} C \\ 1 \end{bmatrix} = 0 \tag{3.14}$$

Operating, it is easy to obtain equation 3.15. The process is described in the following equations:

$$\begin{aligned} \mathbf{P}C + p &= 0 \\ -\mathbf{P}C &= p \\ -\mathbf{P}^{-1}\mathbf{P}C &= \mathbf{P}^{-1}p \\ C &= -\mathbf{P}^{-1}p \end{aligned} \tag{3.15}$$

The same relation is applied for obtaining the focal point of the other camera using equation 3.16.

$$C' = -\mathbf{P}'^{-1}p' \tag{3.16}$$

The reader must note that we have computed the focal point from a matrix $\widetilde{\mathbf{P}}$ defined up to a scale factor. But note that if $\widetilde{\mathbf{P}}$ is replaced by $\widetilde{\lambda\mathbf{P}}$, then p is replaced by λp and $\mathbf{P}^{-1}$ by $\frac{1}{\lambda}\mathbf{P}^{-1}$, resulting that point C does not change, as is required.

Let us explain how the epipolar line of a given image point can be computed. In order to compute a line, we need two points. One is the epipole which is given by the projection of the focal point of a camera on the image plane of the other camera. The other image point is the projection of an object point placed at the infinite.

The epipole e is given by the projection of C' on I, and can be computed from 3.5 and 3.15. The other epipole e' is given by the projection of C on I', and can be computed from 3.6 and 3.16. Then, both epipoles are given by 3.17 and 3.18 respectively.

$$e = \widetilde{\mathbf{P}} \begin{bmatrix} -\mathbf{P}'^{-1}p' \\ 1 \end{bmatrix} \tag{3.17}$$

$$e' = \widetilde{\mathbf{P}}' \begin{bmatrix} -\mathbf{P}^{-1}p \\ 1 \end{bmatrix} \tag{3.18}$$

The other point of interest is the projection of a point placed at the infinite defined as M_∞ . In fact, this point satisfies the equation 3.5, which can be expressed as 3.20.

$$m = \widetilde{\mathbf{P}}M_\infty \tag{3.19}$$

$$M_\infty = \widetilde{\mathbf{P}}^{-1}m \tag{3.20}$$

Then, the other point which defines the epipolar line of m is the projection of M_∞ on I', see figure 3.3, which can be computed through the equation 3.21,

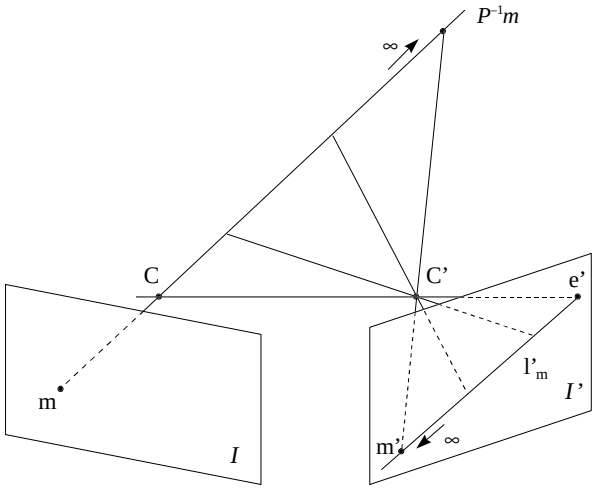

Figure 3.3: The relation of a 2D image point m in I and its epipolar line l'_m on I'.

$$m'_\infty = \widetilde{\mathbf{P}}'\mathbf{P}^{-1}m \tag{3.21}$$

The reader is referred to section 2.2.4 to verify that m is not influenced by the translation vector p, thus equation 3.21 can be simplified by equation 3.22.

$$m'_\infty = \mathbf{P}'\mathbf{P}^{-1}m \tag{3.22}$$

Now we are able to define the epipolar line from both end points. Since the epipolar line lies on an image plane, it can be represented from the cross-product of both points, which allows us to compute the orthogonal vector of the line. Then, the epipolar line l'_m will be defined from the cross-product between e' and m', related in equation 3.23.

$$l'_m = e' \wedge m'_\infty \tag{3.23}$$

Computing, we can express l'_m by equation 3.25,

$$l'_m = e' \wedge \mathbf{P}'\mathbf{P}^{-1}m \tag{3.24}$$

$$l'_m = [e']_x \mathbf{P}'\mathbf{P}^{-1}m \tag{3.25}$$

Let us justify how a cross-product between two vectors can be expressed in a matricial way using the antisymmetric matrix. Consider two vectors p and p', then the cross-product $p \wedge p'$ can be expressed as the product of the antisymmetric matrix of p defined as $[p]_x$ by p', as has been proved in 3.27.

$$p = \begin{pmatrix} x & y & z \end{pmatrix}^t \qquad p' = \begin{pmatrix} x' & y' & z' \end{pmatrix}^t \tag{3.26}$$

$$\begin{aligned} l &= p \wedge p' = \begin{bmatrix} i & j & k \\ x & y & z \\ x' & y' & z' \end{bmatrix} = \begin{bmatrix} yz' - zy' \\ zx' - xz' \\ xy' - yx' \end{bmatrix} = \\ &= \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = [p]_x \, p' \end{aligned} \tag{3.27}$$

At this point, the fundamental matrix can be computed as the relation of the epipolar line l'_m of an image point m and its correspondence point m'. Given a correspondence point m', of m, it must lie on line l'_m. Then,

$$m'^{\,t} l'_m = 0 \tag{3.28}$$

since,

$$pp' = \|p\| \, \|p'\| \cos(\varphi) \tag{3.29}$$

In fact 3.28 expresses the product of two vectors. The first one is the vector defined by the focal point C' and the correspondence point m'. The second one, $l'm$ is defined by the cross-product between vector $C'e'$ and vector $C'm'_\infty$. If this product is equal to 0 , then the three vectors are co-planar.

Operating, 3.28 can be expressed as,

$$m'^{\,t} [e']_x \mathbf{P}'\mathbf{P}^{-1} m = 0 \tag{3.30}$$

Then, we can define the fundamental matrix $\mathbf{F}$, as the 3_x3 matrix, which relates the epipolar line l'_m of an image point m and its correspondence point m' in the following way,

$$m'^{\,t} \mathbf{F} m = 0 \tag{3.31}$$

$$\mathbf{F} = [e']_x \mathbf{P}'\mathbf{P}^{-1} \tag{3.32}$$

Without complexity the reader can understand that another fundamental matrix can be extracted from the relation of both cameras. $\mathbf{F}'$ will be

defined as the fundamental matrix which relates the epipolar line $l_{m'}$ of an object point m' and its correspondence point m. $\mathbf{F}'$ is computed through the following equations,

$$\begin{aligned} l_{m'} &= e \wedge m_\infty && (3.33)\\ l_{m'} &= e \wedge \mathbf{P}\mathbf{P}'^{-1}m' && (3.34)\\ l_{m'} &= [e]_x \mathbf{P}\mathbf{P}'^{-1}m' && (3.35)\\ m^t l_{m'} &= 0 && (3.36)\\ m^t [e]_x \mathbf{P}\mathbf{P}'^{-1}m' &= 0 && (3.37)\\ m^t \mathbf{F}' m' &= 0 && (3.38)\\ \mathbf{F}' &= [e]_x \mathbf{P}\mathbf{P}'^{-1} && (3.39) \end{aligned}$$

Where it can be proved that $\mathbf{F}' = \mathbf{F}^t$, so,

$$\begin{aligned} m'\ {}^t\mathbf{F} m &= 0 && (3.40)\\ m^t \mathbf{F}^t m' &= 0 && (3.41) \end{aligned}$$

The reader is referred to section 3.2.3 where it is set forth and proved that $\mathbf{F}' = \mathbf{F}^t$.

3.2 Changing the co-ordinate system

In the following, we indicate which simplifications can be considered in the definition of the fundamental matrix if we assume that we are using the normalised camera model and, if the world reference co-ordinate system coincides with the co-ordinate system of one camera. As a first approach, consider only the relation with both camera co-ordinate systems, without taking into account the intrinsic matrices $\mathbf{A}$ or $\mathbf{A}'$. This fact will simplify the equations. However, the intrinsic matrices will be used later.

3.2.1 World co-ordinate system fixed at the second camera, operating with respect to the first camera.

Let us establish the world co-ordinate system W at the second camera, with an origin C' and a co-ordinate system X'. Then, the co-ordinate system X of the first camera can be related to the second one by a rotational and a

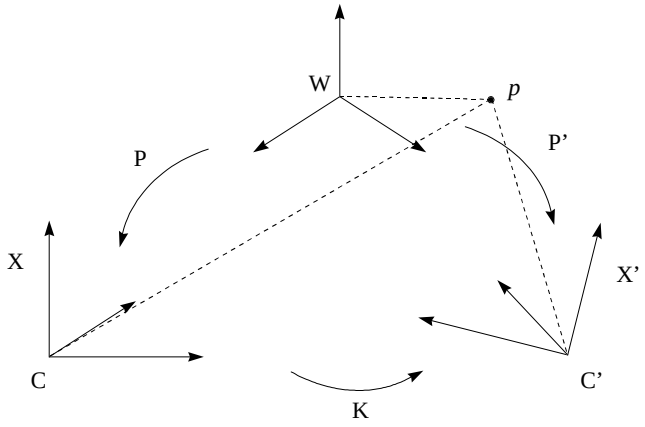

Figure 3.4: Geometrical relation of both camera axes and a world co-ordinate system. World co-ordinate system fixed at the first camera.

translational movement, which allow us to express the fundamental matrix from both movements without considering the general $\mathbf{P}$ and $\mathbf{P}'$ matrices.

Consider the geometrical relation shown in figure 3.4.

If the world co-ordinate system W is placed at C, with the same orientation of X, the transformation matrix of the first camera, already introduced in the previous section, is given by,

$$\widetilde{\mathbf{P}} = \left[\begin{array}{cc} \mathbf{I} & 0 \end{array} \right] \tag{3.42}$$

which allows us to define the second co-ordinate system X', by a matrix defined by a rotation matrix $\mathbf{R}$ and a translation vector t,

$$\widetilde{\mathbf{P}}' = \widetilde{\mathbf{P}}\mathbf{K}^{-1} \qquad \mathbf{K} = \left[\begin{array}{cc} \mathbf{R} & t \\ 0_3^t & 1 \end{array} \right] \tag{3.43}$$

Then, the equation 3.43 can be obtained from figure 3.4 applying the following relations,

$$\begin{aligned} p|_C &= \mathbf{K}^{-1} p|_{C'} && (3.44)\\ \widetilde{\mathbf{P}}^{-1} p|_W &= \mathbf{K}^{-1} \widetilde{\mathbf{P}}'^{-1} p|_W && (3.45)\\ \widetilde{\mathbf{P}}^{-1} &= \mathbf{K}^{-1} \widetilde{\mathbf{P}}'^{-1} && (3.46)\\ \widetilde{\mathbf{P}} &= \widetilde{\mathbf{P}}'\mathbf{K} && (3.47)\\ \widetilde{\mathbf{P}}\mathbf{K}^{-1} &= \widetilde{\mathbf{P}}'\mathbf{K}\mathbf{K}^{-1} && (3.48) \end{aligned}$$

$$\widetilde{\mathbf{P}}' = \widetilde{\mathbf{P}}\mathbf{K}^{-1} \tag{3.49}$$

Operating the equation 3.43 we obtain the matrix $\widetilde{\mathbf{P}}'$ expressed in equation 3.50.

$$\widetilde{\mathbf{P}}' = \begin{bmatrix} \mathbf{R}^t & -\mathbf{R}^t t \end{bmatrix} \tag{3.50}$$

As,

$$\begin{aligned} \mathbf{K}^{-1} &= \begin{bmatrix} \mathbf{R} & t \\ 0_3^t & 1 \end{bmatrix}^{-1} = \left(\begin{bmatrix} \mathbf{I} & t \\ 0_3^t & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R} & 0 \\ 0_3^t & 1 \end{bmatrix} \right)^{-1} = \\ &= \begin{bmatrix} \mathbf{R} & 0 \\ 0_3^t & 1 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{I} & t \\ 0_3^t & 1 \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{R}^t & 0 \\ 0_3^t & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I} & -t \\ 0_3^t & 1 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{R}^t & -\mathbf{R}^t t \\ 0_3^t & 1 \end{bmatrix} \end{aligned} \tag{3.51}$$

Next, we will simplify the following equations which describe the epipoles,

$$e = \widetilde{\mathbf{P}} \begin{bmatrix} C' \\ 1 \end{bmatrix} \tag{3.52}$$

$$e' = \widetilde{\mathbf{P}}' \begin{bmatrix} C \\ 1 \end{bmatrix} \tag{3.53}$$

We can relate the focal point C' with respect to X, as the point located at a distance $\overline{CC'}$ from the origin C. Then, the epipole e can be simplified in the following way,

$$e = \widetilde{\mathbf{P}} \begin{bmatrix} \overline{CC'} \\ 1 \end{bmatrix} = \widetilde{\mathbf{P}} \begin{bmatrix} t \\ 1 \end{bmatrix} = t \tag{3.54}$$

In the same way, using equation 3.50 and the fact that C is the origin of the world co-ordinate system, we obtain,

$$e' = \widetilde{\mathbf{P}}' \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}^t & -\mathbf{R}^t t \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = -\mathbf{R}^t t \tag{3.55}$$

The epipolar line l'_m can be simplified by the following equation,

$$l'_m = [e']_x \mathbf{P}'\mathbf{P}^{-1} m = -\mathbf{R}^t t \wedge \mathbf{R}^t m \tag{3.56}$$

as,

$$\mathbf{P}'\mathbf{P}^{-1} = \mathbf{R}^t\mathbf{I} = \mathbf{R}^t \tag{3.57}$$

then, from the features of the cross-product, we know that equation 3.56 can be expressed as,

$$l'_m = -\mathbf{R}^t\left(t \wedge m\right) = -\mathbf{R}^t\left[t\right]_x m \tag{3.58}$$

As $\mathbf{F}$ is always defined from a scale factor we can ignore the sign, then,

$$l'_m = \mathbf{R}^t\left[t\right]_x m \tag{3.59}$$

And the other epipolar line $l_{m'}$ can be expressed by,

$$l_{m'} = e \wedge \mathbf{P}\mathbf{P}'^{-1}m' = t \wedge \mathbf{R}m' = \left[t\right]_x \mathbf{R}m' \tag{3.60}$$

that is because $\mathbf{P}'^{-1} = \left(\mathbf{R}^t\right)^{-1} = \mathbf{R}$, as for any rotation matrix $\mathbf{R}$ it is always true that $\mathbf{R}^{-1} = \mathbf{R}^t$.

Now, we want to consider the intrinsic matrices. Then, given an ideal projection m and m' in metric units, its real projection $\widetilde{m}$ and $\widetilde{m}'$ in pixels, are given by the following equations,

$$\begin{array}{ll} \widetilde{m} = \mathbf{A}m & m = \mathbf{A}^{-\mathbf{1}}\widetilde{m} \\ \widetilde{m}' = \mathbf{A}'m' & m' = \mathbf{A}'^{-\mathbf{1}}\widetilde{m}' \end{array} \tag{3.61}$$

Then, we obtain the following equation,

$$\widetilde{m}'^{\,t}\mathbf{F}\widetilde{m} = 0 \tag{3.62}$$

where,

$$\mathbf{F} = \mathbf{A}'^{\,-t}\mathbf{R}^t\left[t\right]_x \mathbf{A}^{-1} \tag{3.63}$$

The following explains how we have obtained equation 3.63. We define an essential matrix $\mathbf{E}$ which does not consider the intrinsic matrices like the old fundamental matrix,

$$\mathbf{E} = \mathbf{R}^t\left[t\right]_x \tag{3.64}$$

then,

$$m'^{\,t}\mathbf{E}m = 0 \tag{3.65}$$

using the equations defined in 3.61 we obtain,

$$\left(\mathbf{A}'^{-\mathbf{1}}\widetilde{m}'\right)^t \mathbf{E}\left(\mathbf{A}^{-\mathbf{1}}\widetilde{m}\right) = 0 \tag{3.66}$$

since we know that, $(\mathbf{AB})^t = \mathbf{B}^t\mathbf{A}^t$.

Then we can obtain equation 3.67 which is similar to 3.63 if we substitute $\mathbf{E}$ by equation 3.64.

$$\widetilde{m}'^{\,t}\mathbf{A}'^{\,-t}\mathbf{E}\mathbf{A}^{-1}\widetilde{m} = 0 \tag{3.67}$$

The other fundamental matrix can be calculated in the same way, obtaining,

$$m^t\mathbf{F}'m' = 0 \tag{3.68}$$

where,

$$\mathbf{F}' = \mathbf{A}^{-t}\left[t\right]_x\mathbf{R}\mathbf{A}'^{-1} \tag{3.69}$$

In the same way, we can prove how we have obtained equation 3.69 defining another essential matrix $\mathbf{E}'$, as related in the following equation.

$$\mathbf{E}' = \left[t\right]_x\mathbf{R} \tag{3.70}$$

using 3.61 we obtain,

$$\left(\mathbf{A}'^{\,-1}\widetilde{m}\right)^t\mathbf{E}'\left(\mathbf{A}'^{\,-1}\widetilde{m}'\right) = 0 \tag{3.71}$$

which allows us to give the following equation, which is similar to 3.69 if we substitute $\mathbf{E}'$ by 3.70,

$$\widetilde{m}^t\mathbf{A}^{\,-t}\mathbf{E}'\mathbf{A}'^{\,-1}\widetilde{m}' = 0 \tag{3.72}$$

3.2.2 World co-ordinate system fixed at the first camera, operating with respect to the second camera.

Some authors prefer to establish the world co-ordinate system at the first camera. Then, the whole geometry is computed considering that we are placed at the second co-ordinate system.

Let us establish the world co-ordinate system W at the first camera, with an origin C and a co-ordinate system X. Then, the co-ordinate system X' of the second camera can be related with respect to the first one by a rotation and a translation. This relation, as has been explained in the previous section, allows us to express the fundamental matrix without considering the general $\mathbf{P}$ and $\mathbf{P}'$ matrices.

Consider the geometrical relation shown in figure 3.5.

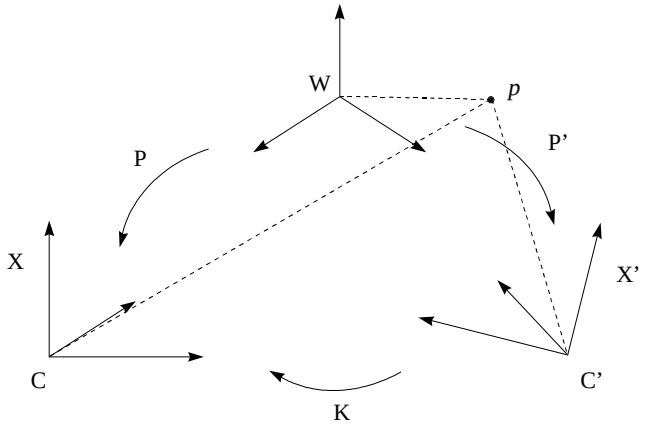

Figure 3.5: Geometrical relation of both camera axes and a world co-ordinate system. World co-ordinate system fixed at the second camera.

Let us place the world co-ordinate system W at C', with the same orientation of the co-ordinate system X'. Then, the transformation matrix $\widetilde{\mathbf{P}}'$ of the second camera is given by,

$$\widetilde{\mathbf{P}}' = \left[\begin{array}{cc} \mathbf{I} & 0 \end{array} \right] \tag{3.73}$$

which allows us to define the first co-ordinate system X, by a matrix $\widetilde{\mathbf{P}}$ defined by a rotation matrix $\mathbf{R}$ and a translation vector t,

$$\widetilde{\mathbf{P}} = \widetilde{\mathbf{P}}'\mathbf{K}^{-1} \qquad \mathbf{K} = \left[\begin{array}{cc} \mathbf{R} & t \\ 0_3^t & 1 \end{array} \right] \tag{3.74}$$

Operating,

$$\widetilde{\mathbf{P}} = \left[\begin{array}{cc} \mathbf{R}^t & -\mathbf{R}^t t \end{array} \right] \tag{3.75}$$

Then, the two epipoles can be computed,

$$e = \widetilde{\mathbf{P}} \left[\begin{array}{c} C' \\ 1 \end{array} \right] = \left[\begin{array}{cc} \mathbf{R}^t & -\mathbf{R}^t t \end{array} \right] \left[\begin{array}{c} 0 \\ 1 \end{array} \right] = -\mathbf{R}^t t \tag{3.76}$$

$$e' = \widetilde{\mathbf{P}}' \left[\begin{array}{c} C \\ 1 \end{array} \right] = \widetilde{\mathbf{P}}' \left[\begin{array}{c} \overline{CC'} \\ 1 \end{array} \right] = \left[\begin{array}{cc} \mathbf{I} & 0 \end{array} \right] \left[\begin{array}{c} t \\ 1 \end{array} \right] = t \tag{3.77}$$

The epipolar line l'_m can be simplified by the following equation,

$$l'_m = [e']_x \mathbf{P}'\mathbf{P}^{-1} m = t \wedge \mathbf{R}m = [t]_x \mathbf{R}m \tag{3.78}$$

and the other epipolar line $l_{m'}$ by,

$$l_{m'} = e \wedge \mathbf{P}\mathbf{P}'^{-1} m' = -\mathbf{R}^t t \wedge \mathbf{R}^t m' = -\mathbf{R}^t [t]_x m' = \mathbf{R}^t [t]_x m' \tag{3.79}$$

Considering the same mathematical procedure explained in equations 3.61-3.72 we can include the intrinsic matrices. Finally,a fundamental matrix can be obtained from each epipolar line,

$$\begin{aligned} \widetilde{m}'^{\,t} l'_m &= \widetilde{m}'^{\,t} \mathbf{F} \widetilde{m} = 0 && (3.80)\\ \mathbf{F} &= \mathbf{A}'^{\,-t} \mathbf{E} \mathbf{A}^{-1} && (3.81)\\ \mathbf{E} &= [t]_x \mathbf{R} && (3.82) \end{aligned}$$

and,

$$\begin{aligned} \widetilde{m}^t l_{m'} &= \widetilde{m}^t \mathbf{F}' \widetilde{m}' = 0 && (3.83)\\ \mathbf{F}' &= \mathbf{A}^{-t} \mathbf{E}' \mathbf{A}'^{\,-1} && (3.84)\\ \mathbf{E}' &= \mathbf{R}^t [t]_x && (3.85) \end{aligned}$$

3.2.3 Relation between both Fundamental Matrices

This section will prove that both fundamental matrices $\mathbf{F}$ and $\mathbf{F}'$ are related by a transpose, that is,

$$\begin{aligned} \mathbf{F}' &= \mathbf{F}^t && (3.86)\\ \mathbf{F} &= \mathbf{F}'^{\,t} && (3.87) \end{aligned}$$

Firstly, we have to remember some useful features of the matricial algebra:

$$\begin{aligned} (\mathbf{AB})^{-1} &= \mathbf{B}^{-1}\mathbf{A}^{-1}\\ (\mathbf{AB})^t &= \mathbf{B}^t\mathbf{A}^t\\ (\mathbf{A}^{-1})^t &= (\mathbf{A}^t)^{-1} = \mathbf{A}^{-t}\\ [t]_x^t &= -[t]_x \end{aligned}$$

We will demonstrate the transpose relation between the fundamental matrices of section 3.2.1. The reader may note that this demonstration also verifies the fundamental matrices of section 3.2.2.

$$\begin{aligned} \mathbf{F} &= \mathbf{A}'^{-t}\mathbf{R}^t\,[t]_x\,\mathbf{A}^{-1} && (3.88)\\ \mathbf{F}^t &= \left(\mathbf{A}'^{-t}\mathbf{R}^t\,[t]_x\,\mathbf{A}^{-1}\right)^t = \left(\mathbf{R}^t\,[t]_x\,\mathbf{A}^{-1}\right)^t\mathbf{A}'^{-1} = && (3.89)\\ &= \left([t]_x\,\mathbf{A}^{-1}\right)^t\mathbf{R}\mathbf{A}'^{-1} = -\mathbf{A}^{-t}\,[t]_x\,\mathbf{R}\mathbf{A}'^{-1} = -\mathbf{F} && (3.90) \end{aligned}$$

and,

$$\begin{aligned} \mathbf{F}' &= \mathbf{A}^{-t}\,[t]_x\,\mathbf{R}\mathbf{A}'^{-1} && (3.91)\\ \mathbf{F}'^{t} &= \left(\mathbf{A}^{-t}\,[t]_x\,\mathbf{R}\mathbf{A}'^{-1}\right)^t = \left([t]_x\,\mathbf{R}\mathbf{A}'^{-1}\right)\mathbf{A}^{-1} = && (3.92)\\ &= -\left(\mathbf{R}\mathbf{A}'^{-1}\right)^t\,[t]_x\,\mathbf{A}'^{-1} = -\mathbf{A}'^{-t}\mathbf{R}^t\,[t]_x\,\mathbf{A}^{-1} = -\mathbf{F} && (3.93) \end{aligned}$$

Since $\mathbf{F}$ and $\mathbf{F}'$ are always defined from a scale factor, we have demonstrated the relation, as the sign can be ignored.

3.2.4 Features of Points and Vectors

We have decided to relate both co-ordinate sytems by a rotation matrix $\mathbf{R}$ and a translation vector t. The second system can be related from the first by the following equation,

$$M_{last} = \mathbf{R}M_{first} + t \tag{3.94}$$

Any image point m_p is defined as vector m from the focal point C and the point on the image plane m_p. For a convenience, we have used m as a point, although we have to bear in mind that it is actually a vector.

$$m = \overline{Cm_p} \tag{3.95}$$

At this point we have to remember two features :

1) Given a point C in the first image which has been fixed as the origin of the co-ordinate system. Its correspondence point in the second co-ordinate system is given only by the translation vector t.

$$C_{last} = \mathbf{R}C_{first} + t = \mathbf{R}\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + t = t \tag{3.96}$$

$$t = \overline{C_{last}C_{first}} \tag{3.97}$$

2) Given a vector v related from the first co-ordinate system, its transformed vector from the second co-ordinate system is defined only by the rotation matrix $\mathbf{R}$, as a vector is only defined by its orientation with respect to each axe, not with respect to its position in the space.

$$v_{last} = \mathbf{R}v_{first} + t = \mathbf{R}v_{first} \tag{3.98}$$

Both features have been broadly used in the previous sections in order to deduce the fundamental matrix.

3.3 Determining the fundamental matrix

The problem to consider is the estimation of the fundamental matrix from a set of image point correspondences $\{\widetilde{m}_i, \widetilde{m}'_i\}$. We are not interested in how the correspondences between two images can be established. However, the reader must know that the fundamental matrix will be calculated from these correspondences. The accuracy of the results will depend directly on the accuracy of the technique used to estimate the matrix, and in a great magnitude, it also depends on the technique used to determine the correspondences between both images. The correspondences between two images can be established using any of the techniques described in [Zhan 93b].

In fact, the determination of the fundamental matrix is the problem of solving equation 3.99. In other words, it is the problem of finding out the coefficients of matrix $\mathbf{F}$ which solve the following equation,

$$\widetilde{m}'^{\,t}\mathbf{F}\widetilde{m} = 0 \tag{3.99}$$

Several methods have been proposed in recent years. The reader is referred to [Luon 93b] to have an overview of the problems involved in determining the fundamental matrix. Two methods have been studied, the first one is based on the iterative technique of function minimisation proposed by Newton-Raphson which has been introduced in chapter two, and

therefore, not explained in this section. However, iterative methods usually spend a lot of computing time, contrary to non-iterative methods. So, the widely known non-iterative Eight-Points Method [Zhan 96c] which uses eight or more matching points in order to obtain the Fundamental Matrix, is presented and analysed in the following section.

3.3.1 The Eight-Points Method

This method does not addresses the obtention of the unknowns parameters of the camera, but the obtaining of the coefficients of the fundamental matrix. So, the method will allow us to compute the epipolar geometry, but, it does not lead us to extract the camera parameters of the system.

In practice, more than eight points will be used to compute the fundamental matrix to assure an accurate result. The method is called the Eight-Points Method because a minimum of eight points are needed to use this method, as there are only eight unknowns to determine the matrix if we consider that the matrix is defined up to a scale factor.

Equation 3.99 can be written in the following way.

$$[x'_i, y'_i, 1] \begin{bmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = 0 \tag{3.100}$$

Operating, we obtain that,

$$U_n f = 0 \tag{3.101}$$

where,

$$U_n = (u_1, u_2, ..., u_n)^t \tag{3.102}$$

$$f = (F_{11}, F_{12}, F_{13}, F_{21}, F_{22}, F_{23}, F_{31}, F_{32}, F_{33})^t \tag{3.103}$$

and,

$$u_i = (x_i x'_i, y_i x'_i, x'_i, x_i y'_i, y_i y'_i, y'_i, x_i, y_i, 1) \tag{3.104}$$

The solution of equation 3.101 is $f = 0$, which is not wanted. To avoid this, we need to impose some constraints on the coefficients of the Fundamental Matrix. However, several methods have been presented in recent years [Luon 93b]. We explain the easiest one, that sets one of the coefficients of $\mathbf{F}$ to a known value. Since the fundamental matrix is always defined from a scale factor, we can set one of the coefficients of $\mathbf{F}$ to 1. This is one of

the constraints often used in the calculation of the fundamental matrix. The process of which is described in the following section. Note that equation 3.101 has a unique solution if $Rank(U) \leq min\{n, m\}$, then, another method is also described which is based on achieving the singular values of U.

The eight-point method is a standard and it has been proved that is very sensitive to noise and quite unstable. Consider the following improvement proposed by some authors [Zhan 96c] [Mohr 96]. A typical image co-ordinate frame in a 512x512 image might be about 200. Some of the entries in a row of u_i are xx' about 200^2, others are x about 200 and the last entry is 1. So, there is a variation in size of 200^2 among the entries of U_n, and some of 200^4 in the entries of $U_n^t U_n$. This means that $U_n^t U_n$ is extremely ill-conditioned. A simple solution to this is to normalise the pixels co-ordinates from $[0, 512]$ to $[-1, 1]$ before proceeding to the calculation of the fundamental matrix. This provides a well-balanced matrix and much more stable and accurate results for $\mathbf{F}$.

Solving F from constraint $F_{33} = 1$

If we impose that coefficient $F_{33} = 1$, then from equation 3.100 we can relate equation 3.101 in the following manner,

$$U'_n f' = -1_n \tag{3.105}$$

where,

$$U'_n = (u'_1, u'_2, ..., u'_n)^t \tag{3.106}$$

$$f' = (F_{11}, F_{12}, F_{13}, F_{21}, F_{22}, F_{23}, F_{31}, F_{32})^t \tag{3.107}$$

and,

$$u'_i = (x_i x'_i, y_i x'_i, x'_i, x_i y'_i, y_i y'_i, y'_i, x_i, y_i) \tag{3.108}$$

Operating equation 3.105 we can obtain the following equation, which allows us to obtain the 8 unknown coefficients of the fundamental matrix.

$$U'^{-1}_n U'_n f' = -U'^{-1}_n 1_n \tag{3.109}$$

$$f' = -U'^{-1}_n 1_n \tag{3.110}$$

$$f' = -\left(U'^{t}_n U'_n\right)^{-1} U'^{t}_n 1_n \tag{3.111}$$

The problem of this method is that we set a determined coefficient to be known and equal to 1. If this coefficient is not equal to 0 no problems

will be come up, but if this is not the case, then the result will be really ill-conditioned. A solution is to try all nine possibilities fixing each F_{ij} coefficient of F to 1 and retainig the best matrix. The best matrix will be the one which reduces the distance of the correspondence points to their epipolar lines.

The following section describes a method which allows us to obtain the 9 unknowns of **F** without considering the assumption that a coefficient of **F** has to be fixed.

Solving F from the eigen values.

Expressing the equation 3.101 in a matricial way,

$$\begin{bmatrix} u_{11} & u_{12} & ... & u_{1m} \\ u_{21} & & & u_{2m} \\ ... & & & ... \\ u_{n1} & u_{n2} & ... & u_{nm} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ \\ f_m \end{bmatrix} = 0 \tag{3.112}$$

Where U is a nxm known matrix and f is the vector to determine. In fact, we know that from any nxm matrix,

$$\forall U_{n_x m} \qquad Rank(U) \leq min\{n, m\} \tag{3.113}$$

that is, the rank of a matrix is the number of columns or rows linearly independent. Consider that $n > m$ as we will always have more equations (correspondence points) than unknowns to determine.

$$\forall U_{n_x m} \qquad Rank(U) \leq m \tag{3.114}$$

Then, from the matrix properties, we can determine f if $Rank(U) < m$, i.e,

$$Rank(U) = m \implies f = 0 \tag{3.115}$$

$$Rank(U) < m \implies f \text{ can be determined} \tag{3.116}$$

At this point, we can affirm that if equation 3.117 is satisfied, then the equation 3.118 is also satisfied [Rote 95].

$$Uf = 0 \tag{3.117}$$

$$U^t U f = 0 \tag{3.118}$$

For each eigen value $\lambda_0 \leq \lambda_1 \leq \lambda_2 ... \leq \lambda p$ of $U^t U$ we can obtain its eigen vector $v_0, v_1, v_2 ... v_p$ from the following equation,

$$\left(\lambda_i \mathbf{I} - U^t U\right) v_i = 0 \tag{3.119}$$

where $\mathbf{I}$ is a $m_x m$ identity matrix. It is interesting to compute the normalised eigen vector, so, we will calculate the following equation from each vector.

$$v_i' = \frac{1}{\left(vx_i + vy_i\right)^{1/2}}(vx, vy) \tag{3.120}$$

If equation 3.116 is satisfied then the first eigen value must be equal to 0, i.e. $\lambda_0 = 0$. If we substitute it into 3.119 we obtain an equation equal to 3.118, then we are able to affirm that f is equal to the eigen vector associated to the smallest eigen value of $U^t U$.

Determination of the eigen values and the eigen vectors of a matrix

Theorem : Any nxn matrix $\mathbf{A}$ has a sequence of eigen values given by the roots of the equation 3.121, where equation 3.121 can be calculated from equation 3.122 known as the characteristic polynom of a matrix [Rote 95].

$$\Delta(\lambda) = 0 \tag{3.121}$$
$$\Delta(\lambda) = \det(\lambda \mathbf{I} - \mathbf{A}) \tag{3.122}$$

Equation 3.122 can be solved, from the rules of the determinants, using 3.123 we can obtain 3.124.

$$\det(\mathbf{A}) = \sum_{j=1}^{n} (-1)^{i+j} a_{ij} \det(\mathbf{A}_{ij}) \tag{3.123}$$
$$\Delta(\lambda) = \lambda^n + a_{n-1}\lambda^{n-1} + ... + a_1\lambda^1 + a_0 \tag{3.124}$$

Then, the characteristic polynom of $\mathbf{A}$ be can expressed from its roots in the following form given by equation 3.125.

$$\Delta(\lambda) = (\lambda - \lambda_0)(\lambda - \lambda_1)...(\lambda - \lambda_n) \tag{3.125}$$

where,

$$\lambda_0 \leq \lambda_1 \leq \lambda_2 ... \leq \lambda_n \tag{3.126}$$

are known as the eigen values of $\mathbf{A}$.

At this point, the eigen vector associated at each eigen value can be calculated by the following equation.

$$(\lambda_i \mathbf{I} - \mathbf{A})\, v_i = 0 \tag{3.127}$$

3.3.2 Conclusions

The eight-points method can be applied for obtaining the fundamental matrix. Although the reader must realise that the intrinsic and extrinsic camera parameters can be achieved from the transformation matrix, the process to obtain them from the fundamental matrix is not so easy (the reader is addressed to chapter two to have a description of the transformation matrix of a single camera). The problem is based on the fact that the parameters of both cameras have been joined in a single 3x3 matrix, so we have 4 intrinsic parameters for each camera and 6 extrinsic parameters, expressing the orientation and translation of a camera from the other one, to determine. Then, we will have 14 unknowns to be determined from only 12 equations. Some authors, as [Luon 92] have considered that the stereo vision system is made by two views taken from the same camera at two different positions, then the intrinsic camera parameters can be determined from the Kruppa equations as only 4 variables are required. When the intrinsic parameters are known, the relative movement of a single camera, or the translation between two or more cameras, can be calculated.

The fact that the methods proposed have always the constraint of using a single camera, or more than one camera with some optical characteristics, has made us decide to use the fundamental matrix only to obtain the epipolar geometry of our stereo system made by a single camera and the light projector. No parameters of the camera and the projector are obtained from the fundamental matrix.

In order to test the equations, a stereoscopic system made by the relationship between two pinhole models has been simulated. Both pinhole models have been placed in a virtual 3D scenario. A set of 3D object points has also been placed in there. Intrinsic parameters have been fixed. Then, 2D correspondences on both image planes, of each 3D object point, have been computed. The sequence of points is shown in the following matrix $\mathbf{M}$, where each row is obtained from the projection of a 3D point on both image planes, that is from $x_{P_1}, y_{P_1}, x_{P_2}, y_{P_2}$.

$$\mathbf{M} = \begin{bmatrix} 134.25 & 117.00 & 117.08 & 102.29 \\ 138.50 & 123.00 & 105.76 & 114.75 \\ 140.62 & 126.00 & 100.06 & 121.01 \\ 159.75 & 117.00 & 136.79 & 104.77 \\ 155.50 & 123.00 & 121.21 & 114.94 \\ 132.12 & 133.87 & 91.55 & 130.75 \\ 142.75 & 132.75 & 126.89 & 119.40 \\ 144.16 & 133.50 & 113.05 & 125.87 \\ 144.87 & 133.87 & 105.82 & 129.24 \\ 168.25 & 132.75 & 143.78 & 118.88 \\ 132.83 & 136.50 & 101.98 & 130.28 \\ 136.37 & 136.12 & 97.06 & 132.78 \\ 151.25 & 137.25 & 133.84 & 123.51 \\ 149.83 & 136.50 & 118.53 & 128.45 \\ 149.12 & 136.12 & 110.29 & 131.12 \\ 134.25 & 153.00 & 122.90 & 141.61 \\ 138.50 & 147.00 & 109.77 & 141.08 \\ 140.62 & 144.00 & 103.12 & 140.81 \\ 159.75 & 153.00 & 141.38 & 137.17 \\ 155.50 & 147.00 & 124.60 & 137.98 \end{bmatrix} \tag{3.128}$$

Only these couples of 2D points have been used to obtain the fundamental matrix $\mathbf{F}$. The iterative Newton-Raphson method has been used, which has been proved to be more accurate than the analytical methods we have explained. The obtained matrix is the following,

$$\mathbf{F} = \begin{bmatrix} -.03159226519 & -.03124566660 & 4.537270990 \\ -.06217629449 & .0006722417077 & -2.109958851 \\ 14.44425460 & 15.62371607 & -2312.072718 \end{bmatrix} \tag{3.129}$$

The epipolar geometry has been recovered. The following figures show the epipolar lines and the correspondences points on both image planes. Note, that all the correspondence points lie on the epipolar line defined by the fundamental matrix and its point couple. No discrepancy is presented as the system has been simulated and does not adjust to a real scenario.

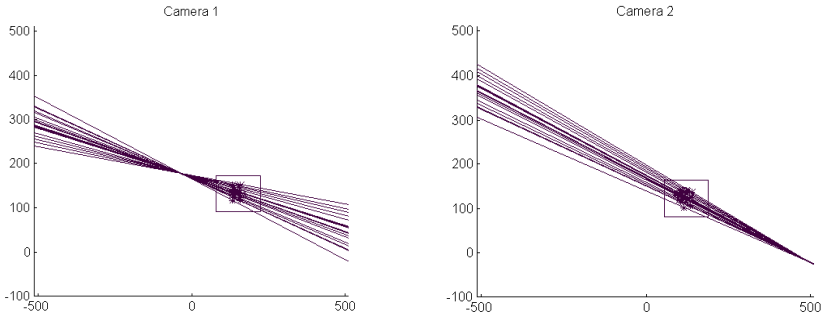

Figure 3.6: An example of the epipolar geometry modelled by the fundamental matrix. Note the epipoles and the epipolar lines found. The area delimited by the correspondence points, have been marked by a square.

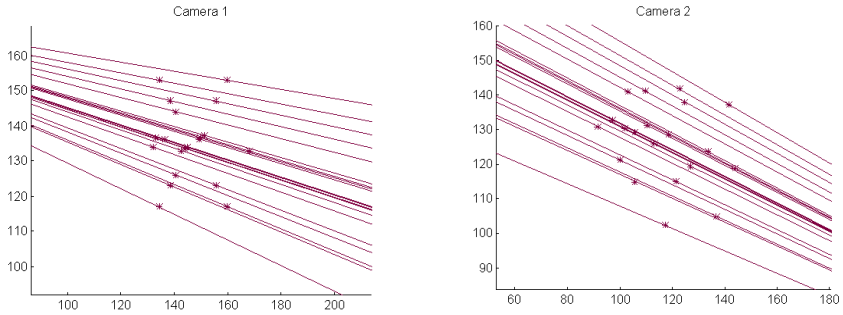

Figure 3.7: A zoom of the square areas of the previous images. Note that the correspondence points lie always on the epipolar lines defined by their couple points.

3.3.3 The algoritm used to infer the fundamental matrix

The algoritm has been programmed using the Maple V Release 4® environment.

```
> with(linalg):
> ############################
> # Obtaining the fundamental matrix
> ############################
> Rx:=array([[1,0,0],[0,cos(a),sin(a)],[0,-sin(a),cos(a)]]);
> Ry:=array([[cos(b),0,-sin(b)],[0,1,0],[sin(b),0,cos(b)]]);
> Rz:=array([[cos(c),sin(c),0],[-sin(c),cos(c),0],[0,0,1]]);
> T:=array([[0,-tz,ty],[tz,0,-tx],[-ty,tx,0]]); R:=multiply(Rx,multiply(Ry,Rz));
> A1:=array([[au1,0,uo1],[0,av1,vo1],[0,0,1]]);
> A2:=array([[au2,0,uo2],[0,av2,vo2],[0,0,1]]);
> A1i:=inverse(A1); A2i:=inverse(A2); A2it:=transpose(A2i);
> FI:=multiply(A2it,multiply(T,multiply(R,A1i)));
> F:=array([[f11,f12,f13],[f21,f22,f23],[f31,f32,f33]]);
> m2:=vector([u2,v2,1]); m1:=vector([u1,v1,1]);
> m2t:=transpose(m2);
> G:=multiply(m2t,multiply(F,m1));
> ############################
> # Obtaining the partial derivatives
> ############################
> dGf11:=diff(G,f11); dGf12:=diff(G,f12);
> dGf13:=diff(G,f13); dGf21:=diff(G,f21);
> dGf22:=diff(G,f22); dGf23:=diff(G,f23);
> dGf31:=diff(G,f31); dGf32:=diff(G,f32);
>dGf33:=diff(G,f33);
> # Read the correspondence points
> readlib(readdata);
> Mp:=readdata(corr_5,6);
> M:=matrix(rowdim(Mp),4,0):
> for i from 1 by 1 to rowdim(Mp) do
> M[i,1]:=op(1,op(i,Mp))/op(3,op(i,Mp)):
> M[i,2]:=op(2,op(i,Mp))/op(3,op(i,Mp)):
> M[i,3]:=op(4,op(i,Mp))/op(6,op(i,Mp)):
> M[i,4]:=op(5,op(i,Mp))/op(6,op(i,Mp)):
> od:
> eval(M);
> # Initial values of the parameters
```

```
> PI:=Pi;
> va:=0.1; vb:=0.7; vc:=0.2; vtx:=1000; vty:=-400; vtz:=230;
> vau1:=85; vav1:=90; vau2:=85; vav2:=90;
> vuo1:=130; vvo1:=135; vuo2:=130; vvo2:=135;
> vf11:=simplify(subs(a=va, b=vb, c=vc, tx=vtx, ty=vty, tz=vtz, au1=vau1,
av1=vav1, au2=vau2, av2=vav2, uo1=vuo1, vo1=vvo1, uo2=vuo2, vo2=vvo2,
FI[1,1]));
> vf12:=simplify(subs(a=va, b=vb, c=vc, tx=vtx, ty=vty, tz=vtz, au1=vau1,
av1=vav1, au2=vau2, av2=vav2, uo1=vuo1, vo1=vvo1, uo2=vuo2, vo2=vvo2,
FI[1,2]));
> vf13:=simplify(subs(a=va, b=vb, c=vc, tx=vtx, ty=vty, tz=vtz, au1=vau1,
av1=vav1, au2=vau2, av2=vav2, uo1=vuo1, vo1=vvo1, uo2=vuo2, vo2=vvo2,
FI[1,3]));
> vf21:=simplify(subs(a=va, b=vb, c=vc, tx=vtx, ty=vty, tz=vtz, au1=vau1,
av1=vav1, au2=vau2, av2=vav2, uo1=vuo1, vo1=vvo1, uo2=vuo2, vo2=vvo2,
FI[2,1]));
> vf22:=simplify(subs(a=va, b=vb, c=vc, tx=vtx, ty=vty, tz=vtz, au1=vau1,
av1=vav1, au2=vau2, av2=vav2, uo1=vuo1, vo1=vvo1, uo2=vuo2, vo2=vvo2,
FI[2,2]));
> vf23:=simplify(subs(a=va, b=vb, c=vc, tx=vtx, ty=vty, tz=vtz, au1=vau1,
av1=vav1, au2=vau2, av2=vav2, uo1=vuo1, vo1=vvo1, uo2=vuo2, vo2=vvo2,
FI[2,3]));
> vf31:=simplify(subs(a=va, b=vb, c=vc, tx=vtx, ty=vty, tz=vtz, au1=vau1,
av1=vav1, au2=vau2, av2=vav2, uo1=vuo1, vo1=vvo1, uo2=vuo2, vo2=vvo2,
FI[3,1]));
> vf32:=simplify(subs(a=va, b=vb, c=vc, tx=vtx, ty=vty, tz=vtz, au1=vau1,
av1=vav1, au2=vau2, av2=vav2, uo1=vuo1, vo1=vvo1, uo2=vuo2, vo2=vvo2,
FI[3,2]));
> vf33:=simplify(subs(a=va, b=vb, c=vc, tx=vtx, ty=vty, tz=vtz, au1=vau1,
av1=vav1, au2=vau2, av2=vav2, uo1=vuo1, vo1=vvo1, uo2=vuo2, vo2=vvo2,
FI[3,3]));
> ########################################
> # Calibrating by the Modified Newton Raphson
> ########################################
> VG:=vector(rowdim(M),0); J:=matrix(rowdim(M),9,0);
> dk:=vector(9,0); VG2:=vector(rowdim(M),0);
> JT:=matrix(9,rowdim(M),0); JTJ:=matrix(9,9,0);
> IJTJ:=matrix(9,9,0); IJTJJT:=matrix(9,rowdim(M),0);
> CV:=matrix(30,10,0);
> for k from 1 by 1 to 15 do
> for i from 1 by 1 to rowdim(M) do
```

```
> vu1:=M[i,1]: vv1:=M[i,2]: vu2:=M[i,3]: vv2:=M[i,4]:
> VG[i]:=simplify(subs(u1=vu1, v1=vv1, u2=vu2, v2=vv2,f11=vf11, f12=vf12,
f13=vf13, f21=vf21, f22=vf22, f23=vf23, f31=vf31, f32=vf32, f33=vf33, G)):
> J[i,1]:=simplify(subs(u1=vu1, v1=vv1, u2=vu2, v2=vv2,f11=vf11, f12=vf12,
f13=vf13, f21=vf21, f22=vf22, f23=vf23, f31=vf31, f32=vf32, f33=vf33, dGf11)):
> J[i,2]:=simplify(subs(u1=vu1, v1=vv1, u2=vu2, v2=vv2,f11=vf11, f12=vf12,
f13=vf13, f21=vf21, f22=vf22, f23=vf23, f31=vf31, f32=vf32, f33=vf33, dGf12)):
> J[i,3]:=simplify(subs(u1=vu1, v1=vv1, u2=vu2, v2=vv2,f11=vf11, f12=vf12,
f13=vf13, f21=vf21, f22=vf22, f23=vf23, f31=vf31, f32=vf32, f33=vf33, dGf13)):
> J[i,4]:=simplify(subs(u1=vu1, v1=vv1, u2=vu2, v2=vv2,f11=vf11, f12=vf12,
f13=vf13, f21=vf21, f22=vf22, f23=vf23, f31=vf31, f32=vf32, f33=vf33, dGf21)):
> J[i,5]:=simplify(subs(u1=vu1, v1=vv1, u2=vu2, v2=vv2,f11=vf11, f12=vf12,
f13=vf13, f21=vf21, f22=vf22, f23=vf23, f31=vf31, f32=vf32, f33=vf33, dGf22)):
> J[i,6]:=simplify(subs(u1=vu1, v1=vv1, u2=vu2, v2=vv2,f11=vf11, f12=vf12,
f13=vf13, f21=vf21, f22=vf22, f23=vf23, f31=vf31, f32=vf32, f33=vf33, dGf23)):
> J[i,7]:=simplify(subs(u1=vu1, v1=vv1, u2=vu2, v2=vv2,f11=vf11, f12=vf12,
f13=vf13, f21=vf21, f22=vf22, f23=vf23, f31=vf31, f32=vf32, f33=vf33, dGf31)):
> J[i,8]:=simplify(subs(u1=vu1, v1=vv1, u2=vu2, v2=vv2,f11=vf11, f12=vf12,
f13=vf13, f21=vf21, f22=vf22, f23=vf23, f31=vf31, f32=vf32, f33=vf33, dGf32)):
> J[i,9]:=simplify(subs(u1=vu1, v1=vv1, u2=vu2, v2=vv2,f11=vf11, f12=vf12,
f13=vf13, f21=vf21, f22=vf22, f23=vf23, f31=vf31, f32=vf32, f33=vf33, dGf33)):
> od;
> #|f(xk)|
> nVGk:= multiply(transpose(VG),VG);
> # dk
> JT := transpose(J): JTJ := multiply(JT,J):
> IJTJ:=inverse(JTJ): IJTJJT:=multiply(IJTJ,JT):
> dk:=evalm(multiply(IJTJJT,VG)):
> # xk
> xk:=vector([vf11,vf12,vf13,vf21,vf22,vf23,vf31,vf32,vf33]);
> # buscant j
> j:=0: acabar:=0:
> for j from 0 by 1 while acabar < 1 do
> xk2:=evalm(xk - 2^(-j)*dk);
> # |f(xk - 2^(-j)*dk)|
> for i from 1 by 1 while i <= rowdim(M) do
> vu1:=M[i,1]: vv1:=M[i,2]: vu2:=M[i,3]: vv2:=M[i,4]:
> VG2[i]:=simplify(subs(u1=vu1, v1=vv1, u2=vu2, v2=vv2, f11=xk2[1], f12=xk2[2],
f13=xk2[3], f21=xk2[4], f22=xk2[5], f23=xk2[6], f31=xk2[7], f32=xk2[8], f33=xk2[9],
G)):
> od:
```

```
> nVG2k:=multiply(transpose(VG2),VG2);
> #|f(xk-2^(-i)*dk)| < |f(xk)|
> if nVG2k < nVGk then acabar:=1 fi;
> od:
> # xk+1
> j:=j-1;
> xk2:=evalm(xk - 2^(-j)*dk);
> vf11:=xk2[1]: vf12:=xk2[2]: vf13:=xk2[3]: vf21:=xk2[4]: vf22:=xk2[5]:
> vf23:=xk2[6]: vf31:=xk2[7]: vf32:=xk2[8]: vf33:=xk2[9]:
> CV[k,1]:=vf11; CV[k,2]:=vf12; CV[k,3]:=vf13; CV[k,4]:=vf21;
> CV[k,5]:=vf22; CV[k,6]:=vf23; CV[k,7]:=vf31; CV[k,8]:=vf32;
> CV[k,9]:=vf33; CV[k,10]:=nVGk;
> if nVGk < 10^(-10) then break; fi;
> od:
> CV(1);
> # The matrix F is obtained in vector CV(1).
```

3.3.4 The algorithm used to obtain the epipolar lines from F

The algoritm has been programmed using the Matlab® environment.

```
clear all; close all;
% Generation of the correspondence points
 % modelling Camera 1. Origin of O world co-ordinate system
 A1 = sym('[au1 0 uo1 0;0 av1 vo1 0;0 0 1 0;0 0 0 1]');
 au1 = 85; av1 = 90; uo1 = 130; vo1 = 135;
 AART1 = A1;
 ART1(1,:) = eval(AART1(1,:));
 ART1(2,:) = eval(AART1(2,:));
 ART1(3,:) = eval(AART1(3,:));
 ART1(4,:) = eval(AART1(4,:));
 % modelling Camera 2. Moved from O.
 Rx = sym('[1 0 0 0; 0 cos(a) sin(a) 0;0 -sin(a) cos(a) 0;0 0 0 1]');
 Ry = sym('[cos(b) 0 -sin(b) 0; 0 1 0 0;sin(b) 0 cos(b) 0;0 0 0 1]');
 Rz = sym('[cos(c) sin(c) 0 0;-sin(c) cos(c) 0 0; 0 0 1 0;0 0 0 1]');
 R2 = symmul(Rx,symmul(Ry,Rz));
 T2 = sym('[1 0 0 tx;0 1 0 ty;0 0 1 tz;0 0 0 1]');
 RT2 = symmul(T2,R2);
 A2 = sym('[au2 0 uo2 0;0 av2 vo2 0;0 0 1 0;0 0 0 1]');
 AART2 = symmul(A2,RT2);
```

```
a = 0.1; b = 0.7; c = 0.2;
tx = 1000; ty = -400; tz = 230;
au2 = 85; av2 = 90; uo2 = 130; vo2 = 135;
ART2(1,:) = eval(AART2(1,:));
ART2(2,:) = eval(AART2(2,:));
ART2(3,:) = eval(AART2(3,:));
ART2(4,:) = eval(AART2(4,:));
% Generating the 3D world point vector
V(:,1) = [100;-400;2000;1]; V(:,2) = [300;-400;3000;1];
V(:,3) = [500;-400;4000;1]; V(:,4) = [700;-400;2000;1];
V(:,5) = [900;-400;3000;1]; V(:,6) = [100;-50;4000;1];
V(:,7) = [300;-50;2000;1]; V(:,8) = [500;-50;3000;1];
V(:,9) = [700;-50;4000;1]; V(:,10) = [900;-50;2000;1];
V(:,11) = [100;50;3000;1]; V(:,12) = [300;50;4000;1];
V(:,13) = [500;50;2000;1]; V(:,14) = [700;50;3000;1];
V(:,15) = [900;50;4000;1]; V(:,16) = [100;400;2000;1];
V(:,17) = [300;400;3000;1]; V(:,18) = [500;400;4000;1];
V(:,19) = [700;400;2000;1]; V(:,20) = [900;400;3000;1];
% Obtening the 2D points in both image planes.
for i=1:20,
P1(:,i) = ART1*V(:,i);
P1(1,i) = P1(1,i)/P1(3,i);
P1(2,i) = P1(2,i)/P1(3,i);
P1(3,i) = P1(3,i)/P1(3,i);
P2(:,i) = ART2*V(:,i);
P2(1,i) = P2(1,i)/P2(3,i);
P2(2,i) = P2(2,i)/P2(3,i);
P2(3,i) = P2(3,i)/P2(3,i);
end
% Fundamental Matrix given by Maple.
F = [-.03159226519 -.03124566660 4.537270990; -.06217629449 .0006722417077
-2.109958851; 14.44425460 15.62371607 -2312.072718];
% Epipolar Geometry of Camera 2.
figure(1);
hold on;
axis([-512,512,-100,512]);
title('Camera 2');
for i=1:20,
lm2(1:3,i)=F*P1(1:3,i);
plot(P2(1,i),P2(2,i),'g*');
```

```
 plot([-512;512] , [(-512 * lm2(1,i) + lm2(3,i)) / (-lm2(2,i)) , (512 * lm2(1,i) +
lm2(3,i)) / (-lm2(2,i))] );
end;
zoom on;
% Epipolar Geometry of Camera 1.
figure(2);
hold on;
axis([-512,512,-100,512]);
title('Camera 1');
for i=1:20,
 lm1(1:3,i)=F'*P2(1:3,i);
 plot(P1(1,i),P1(2,i),'g*');
 plot([-512;512] , [(-512 * lm1(1,i) + lm1(3,i)) / (-lm1(2,i)) , (512 * lm1(1,i) +
lm1(3,i)) / (-lm1(2,i))] );
end;
zoom on;
```

Chapter 4

Structured Light

Stereovision is an attractive and widely used method, but, it is rather limited to build 3D surface maps, due to the correspondence problem. The correspondence problem can be reduced using a method based on the structured light concept, projecting a given pattern on the measuring surfaces. However, some relations between the projected pattern and the reflected one must be solved. This relationship can be directly found codifying the projected light, so that, each imaged region of the projected pattern provides the needed information to solve the correspondence problem. This chapter will analyse stereo vision problems going through structured light. Furthermore, coded structured light is presented, and a survey of the most important techniques is related.

4.1 Introduction

When 3D information of a given surface is needed, we have to choose between a passive method and an active one [Ahle 89]. The most widely known passive method is stereovision [Barn 82] which can be achieved in two different ways. In the first way, an optical sensor is moved to known relative positions in the scene. In the second way, two or more optical sensors are previously fixed in known positions [Ayac 91]. The surface to be measured is projected on the image plane of each sensor through each focal point. As described in chapter 2, 3D co-ordinates of the object point can be obtained by trigonometry [Hara 93] [Hall 82] [Faug 93] from the known projections of an object point and, furthermore, from the relationship between the optical sensors. But we have to know, with correctness, for each object point, its projections on the optical sensor image planes. In fact, in order to obtain the 3D co-ordinates of a given point from n given projections (one from each sensor), these projections must necessarily come from the same object point. This

problem is known as the correspondence problem.

There are some essential problems in stereo vision which difficult to solve the correspondence problem between both image planes [Zhan 93b]. Some authors consider to solve the problem of matching from singular points, i.e. finding points in both image planes with similar neighbourhood characteristics, classifying them as corners and vertexes. Since both 2D images are projections of the same scene, images taken from different positions may be quite different one from the other. Consequently, this hypothesis is rather poor and becomes even poorer if we take into account that the precision of the 3D measurement depends on the distance between both cameras. The precision of the measurement improves increasing this distance. However, when one increases the distance between both cameras, the projective images of the 3D scenes become more different, constraining considerably their match using similarity. Of course, we must also consider that in the same 3D scene more than one object with approximately the same shape and size could be presented. In this case, it becomes rather difficult to match the images from the mere concept of singular points. However, some geometrical constraints can be used to reduce the problem of matching. One of the most popular constraints is known as the epipolar constraint imposed by the geometrical relationship between both cameras ([Faug 93], p 169), shown in figure 4.1. The epipolar relation is defined as follows. The intersection of the line (C_1, C_2) which each image plane, π_1 and π_2, produces two epipoles, named E_1 and E_2. Consider an image point P_1 on π_1, the epipolar geometry allows us to search its correspondence point P_2 on the epipolar line of P_1 on π_2, i.e. on the line e_{p1}, where e_{p1} is defined by the intersection of the plane π_2 and the plane defined by the three points C_1, C_2 and P_1. Note that we do not need any three dimensional information from P_w to infer P_2.

The epipolar constraint allows us to reduce the searching of the correspondence from the two-dimensional space of the image plane into the one-dimensional one of the epipolar line. However, we must be aware that some points in an image plane might not have a correspondence on the other one due to a surface occlusion or simply because it has been projected out of the scope of the camera, see figure 4.2. Even, imaged points may not lie on the epipolar lines in the same order (see figure 4.3 and 4.4). Note that we are unable to know a priory whether a point has a correspondence or not, which difficults enormously the matching and imposes the use of a posterior step to remove false matchings mostly by using a region growing algorithm.

At this point another constraint which arises from the structure of the objects of the scene, can be used. It could be assumed that the scene surfaces depth-vary smoothly almost everywhere. This constraint is known as the disparity gradient. Although this constraint can be used to reduce false

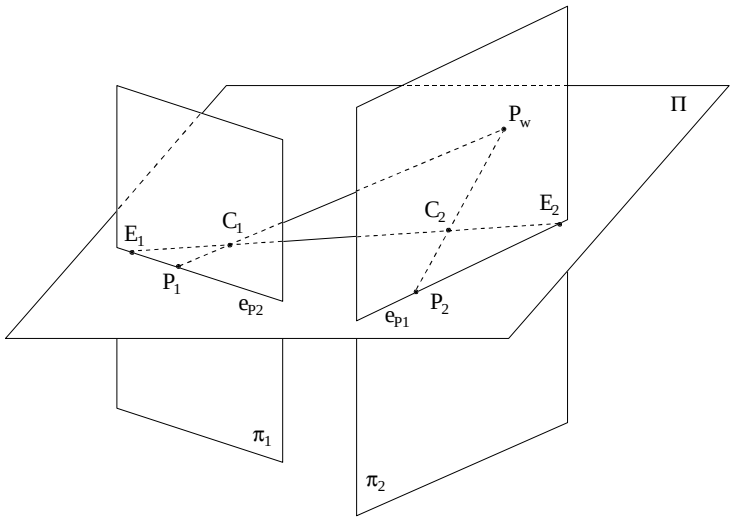

Figure 4.1: The epipolar geometry as a tool to reduce the correspondence problem.

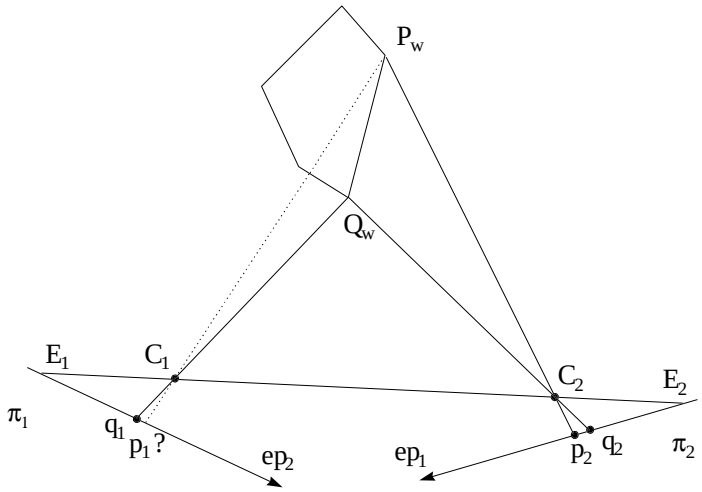

Figure 4.2: An example of a point without correspondence due to a surface occlusion of P_w.

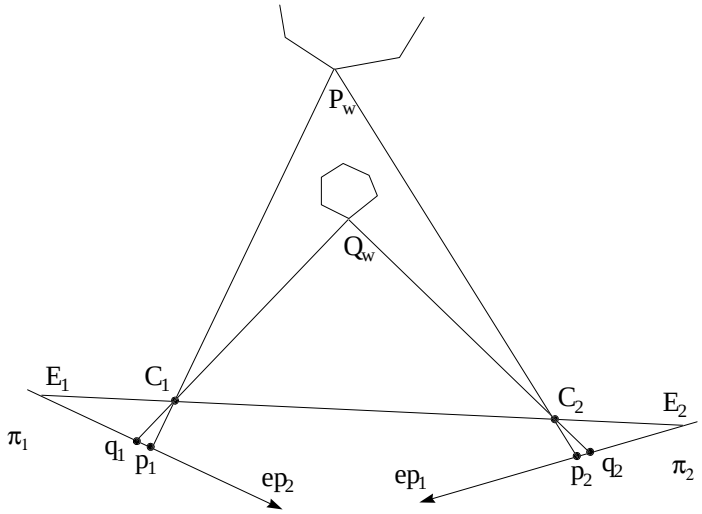

Figure 4.3: An example of correspondence points imaged in the same order from the epipoles.

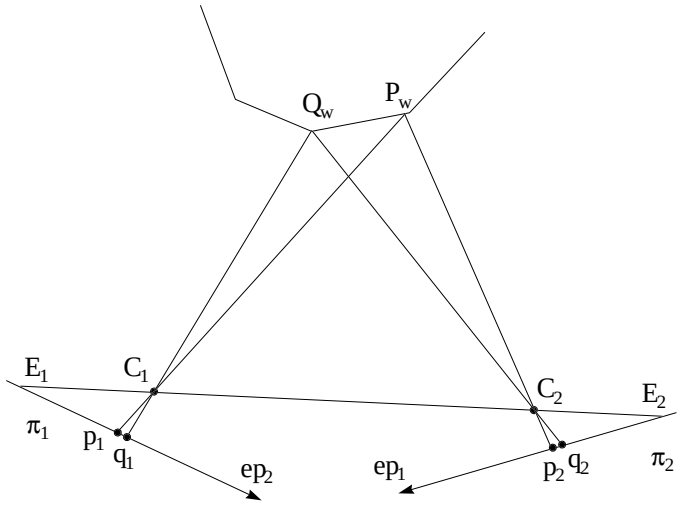

Figure 4.4: An example of correspondence points imaged in a different order from the epipoles.

matches obtained from the geometrical properties of the epipolar line, the reader must know that it can not be used at depth discontinuities. Note that depth discontinuities mainly produce the edges and vertexes that will be used as tokens to the matching process. We must also note that almost all stereo vision systems restrict the obtaining of 3D information from the vertexes and corners of the objects. Hence, the disparity gradient could not be used except if we obtain 3D information following the edges, fact that hardly complicates the matching process increasing the computing time.

It is known that the correspondence problem can be alleviated leaving off stereo vision, and going to an active method [Jarv 83] [Jarv 93] [Brad 88] [Rock 75] [Besl 88]. One of the most widely used active methods is based on structured light projection [Will 71]. Here, the second stereo camera is replaced by a light source, which projects a known pattern of light on the measuring scene. The first stereo camera images the illuminated scene and, analysing the deformations of the imaged pattern with respect to the projected one, the desired 3D information can be obtained. Of course, depending on the chosen pattern, some correspondences between the projected pattern and the imaged one, should be solved. Most of the proposed structured light techniques are based on the projection of regular patterns on the measuring scene. If a single light dot or a slit line is projected on the scene, then, there is no correspondence problem to be solved, but whole the scene has to be scanned to obtain the 3D map. Shirai et al., in 1971, proposed a slit line projection to recognise polihedric objects [Shir 71b]. In 1973, Agin et al. generalised this idea to recognise curvilinear objects [Agin 73]. Two years later, with the goal to recognise either polihedric or curvilinear objects, Popplestone et al. proposed a more general system [Popp 75]. In 1986, Yamamoto et al.[Yama 86] proposed a half plane illumination system instead of a slit line. In fact, binarising an image of a scene illuminated by a half plane pattern could be compared to the boundary edge detection between the illuminated and the obscured area. Recently, some systems to obtain 3D maps of a scene have been presented. The most common systems are similar to the high-speed method presented by Ozeki et al.[Ozek 86], and the video rate method presented by Yokoyama et al.[Yoko 94]. There are also some authors, as Sato et al. [Sato 82], who use a method based on the projection of two slit lines with different orientation and position in the 3D co-ordinates system. In a similar way, Kemmotsu and Kanade [Kemm 94] [Kemm 95] chose to project three lines on the measuring scene.

In order to improve the accuracy of the system, an alternative way is to project a grid of dots or lines on the scene to cover the entire scope of the camera. Asada et al. proposed to use a pattern made by a set of vertical, parallel and equidistant, stripe lines [Asad 85] [Asad 86] [Asad 88]. Wang

et al., have extended Asada's idea with the sequential projection of two orthogonal stripe patterns [Wang 87] [Wang 89] [Wang 91]. Furthermore, in order to obtain 3D surface properties, Gu et al. [Hu 86] [Hu 89], Stockman et al. [Stoc 86] and Shrikhande et al. [Shri 89] have proposed the widely known method based on the projection of a grid. Other authors also use the information given by an intensity image to obtain, with a better accuracy, the boundary edges of the scene [Ikeu 91] [Wang 92] [Hu 89b] [Vemu 86] [Hosh 90]. Then, an easier correspondence problem has to be solved: for each point of the imaged pattern, the corresponding point of the projected pattern has to be identified. All these methods allow us to obtain 3D information from the geometric constraint propagation, especially from the epipolar constraint, and some of them are rather limited to measure surfaces with depth discontinuities.

In recent years, a new structured light technique has increased in importance. This technique is based on an unique codification of each token of light projected on the scene. When the token is imaged by the camera, this codification allows us to obtain the correspondence i.e. to know where it comes from. Then, 3D measurements are directly obtained as we do not have to use hard computational geometric constraints. This technique is basically known as coded structured light. Several coded structured light techniques have been proposed in the last years, which have been discussed and compared in the following survey.

4.2 A Survey: Coded Structured Light

Coded structured light is a technique based on the projection of light patterns on the measuring scene. The projected patterns are conveniently coded in order to solve the correspondence problem. Each token of light is projected on the scene carrying a label which indicates where it comes from. When the token is imaged by the camera, we have to read this label to obtain an unique match.

As it can be deduced from the section dedicated to 3D reconstruction in chapter 2 (see also figure 2.14), we can use four equations in order to calculate the three unknowns that determine the co-ordinates of the object point P_w. In fact, one of the four equations is linearly dependent on the other ones, so, with the aim to determine the three unknowns, only three of the four equations have to be used. As an image of the scene has to be captured to deduce depth scene information, the image point co-ordinates (x_{p2}, y_{p2}) are known, as it is the projection of the object point with co-ordinates (x_{pw}, y_{pw}, z_{pw}). Then, only one equation of the two, which relates

the 3D object point with respect to the 2D projected token on the projector frame, has to be used. That is because with only one equation, there is sufficient information to deduce the 3D object points co-ordinates. Therefore, only one of the two co-ordinates (x_{p1}, y_{p1}) of the projected point has to be known. If the x co-ordinate x_{p1} is known, then the pattern is called a column coded pattern. Otherwise, if it is y_{p1} the only known co-ordinate, then the pattern is called a row coded pattern. This idea allows the projected pattern to be coded just along one component co-ordinate, so the captured light, at the point (x_{p2}, y_{p2}) on the image plane, carries information from row x_{p1}, or column y_{p1}, from which it has been emitted. Obviously, a pattern based on both axes codification is more robust and the results of the 3D measuring points more accurate than single axe codification.

The survey done allowed us to classify the different techniques coding the projected pattern, into three classifications.

Looking only at the temporal dependence, patterns can be classified into:

Static: The pattern is limited to static scenes with motionless objects. This is always due to the necessity to project a set of different patterns to obtain, in a coded way, the label for each column (or row) of the pattern. Any movement inside the scene between two pattern projection frames always produces a correspondence error.

Dynamic: The pattern is not limited to static scenes. Then, if the scene objects can move, the column or row of the projector image has to be coded with a single pattern projection.

Looking only at the light projected, patterns can be classified into:

Binary: Any of the (x_{p1}, y_{p1}) points of the pattern can only have one of two possible values, which are coded with 0 and 1 respectively. This binary value normally represents opacity and transparency, i.e. projected light absence or presence on the object.

Grey level: Each pattern point can have an associated grey value, which represents the transparency (or opacity) level of the point against the projected light. Since the information is coded as a grey light level, normally two steps are necessary in order to find out 3D information. Firstly, we have to obtain an image of the scene illuminated with the same light intensity for each point (without coding). Secondly, we must obtain the reference light

needed to cancel the surface reflection effect, which depends on the kind of surfaces where the light is reflected. This limitation means that the pattern has to be also classified as a static pattern.

Colour: Each pattern point has to be associated with a hue value. In order to use the colour constancy property, the hue values used have to be quite different from each other. The main goal is to get an efficient and accurate segmentation. Since the system projects colour on the scene, its use is limited to a neutral colour scene, as highly saturated colour objects can produce loosing of pattern regions in the segmentation step and posterior decodification. Even pale coloured objects may produce a colour frequency shifting as a result of the intrinsic colour of the measuring objects. Obviously, the discrepancy suffered by the colour captured by the camera with respect to the one projected by the pattern, rather complicates the segmentation step.

Furthermore, another classification can be proposed, which classifies the patterns from its surface depth discontinuities dependence. Then, looking at this dependence, the patterns can be classified as:

Periodical: The codification is periodically repeated along the pattern. Normally, this technique is used to reduce the number of bits that code the pattern, but produce a limitation of the depth discontinuity which can not be larger than half of the period length.

Absolute: Each column or row of the projected pattern has a unique codification. So, it does not have any depth discontinuity dependence.

Hereafter follows a survey of the coded patterns presented in recent years, analysing for each one its advantages and disadvantages.

4.2.1 Posdamer-Altschuler

Altschuler et al. [Alts 81] [Alts 87] and Posdamer et al. [Posd 82] proposed a static binary codification of a striped pattern. The same principle is also proposed by Mundy and Porter [Mund 87] and Minou et al. [Mino 81]. The system, shown in figure 4.5, is based on the utilisation of a pattern structured as a dot matrix of nxn binary light beams. Each n_i column of the pattern can be independently controlled, so it can be lighted or obscured. Then, several

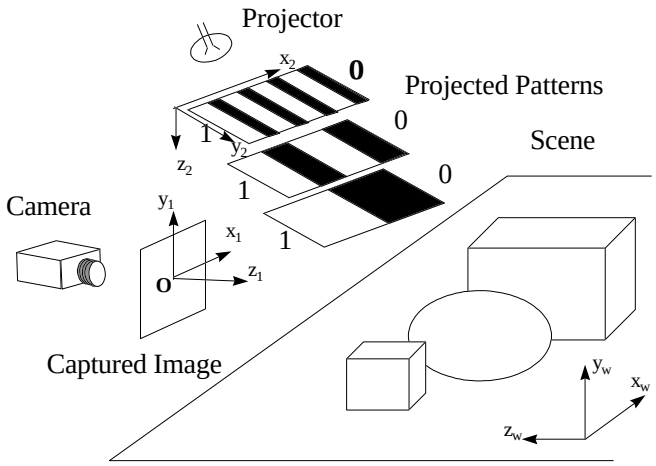

Figure 4.5: The temporary codification.

masks can be used to allow coding any pattern dot, in a temporal way, as a sequential projection of different patterns, as shown in figure 4.6. The number of patterns to be projected is determined by the number of columns to be coded.

Blocking light beams, a pattern image sequence can be projected which projects a sequence of 0 and 1 for each beam. Then, each position of the measuring surface is temporally and sequentially illuminated for the different values of the same dot beam, which produces a code that can distinguish one dot from its neighbours.

In fact, the system works as follows. Firstly, an entire dot illuminated pattern is projected on the scene. The camera images all the dots reflected from the surfaces of the scene, and stores their positions in a data base. In the following steps, each coded pattern is projected and the system can ask for the light information in the stored positions. When all the patterns have been projected, each stored point has a code which allows us to determine from which beam column it has been projected.

The system proposed by Posdamer et al. and Altschuler et al. is limited to static scenes as it has to capture an image from each projected pattern. But, as proposed by Altschuler et al. [Alts 81], the system can be improved to be used in dynamic scenes. In this case, all the patterns are projected with a different frequency wave and n cameras are used, one for each projected pattern. Each camera should have an optical filter, as each camera will image

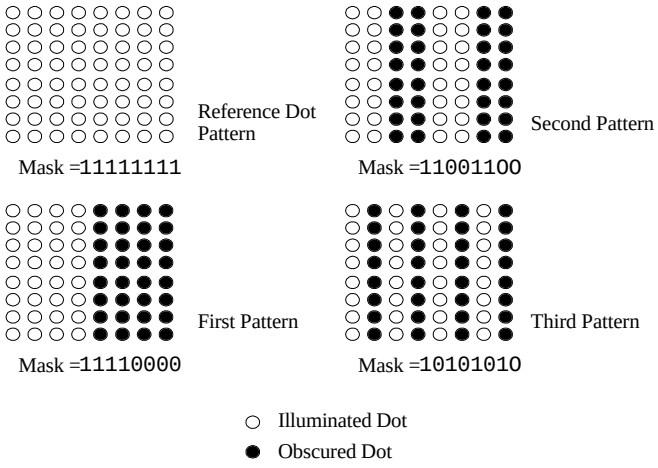

Figure 4.6: Coding a *nxn* matrix of laser beam dots.

only one pattern frequency. Then, all the patterns can be projected at the same time. The cameras should be located as close as possible. Even so, we could also find that some points can not be decoded as they are not imaged from all the cameras.

The idea, proposed by Posdamer and Altschuler, has been widely studied. Basically, the codification and the speed measurement have been improved. For example, Inokuchi et al. [Inok 84] in 1984 proposed changing the binary codification, shown in figure 4.7, to a Gray one, shown in figure 4.8, which is more error robust.

Later, in 1986, Sato et al. [Sato 86], and in 1987, Sato and Inokuchi [Sato 87], proposed to use a liquid crystal device, already studied by Inokuchi et al. [Inok 72] in 1972, which permits a larger number of columns to be projected with a high accuracy. The system also improves the coding speed, against the use of a slide projector, since the LCD can be electronically controlled.

If an object has a high textural contrast or any high reflected surface regions, then, some pattern segmentation errors can be produced. Normally, this problem can be solved capturing the first image without pattern projection (or with the same light projection for each dot) to obtain a reference light intensity. This intensity gives a dynamic threshold for each pixel. Sato et al. [Sato 86], in 1986, and Sato and Inokuchi [Sato 87], in 1987, proposed a complementary method of segmentation based on the projection for each

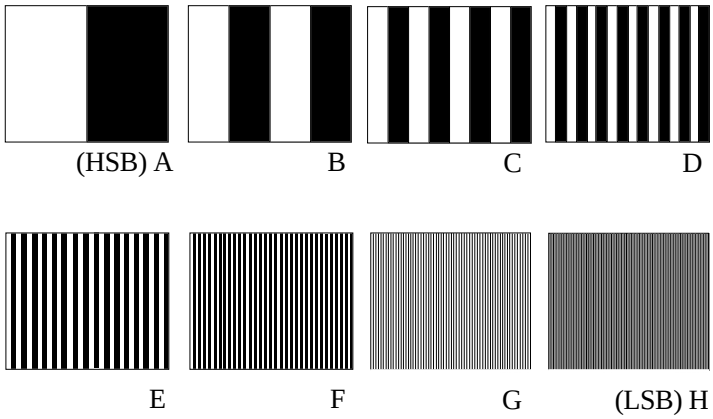

Figure 4.7: 8 bits temporally binary coded pattern projection.

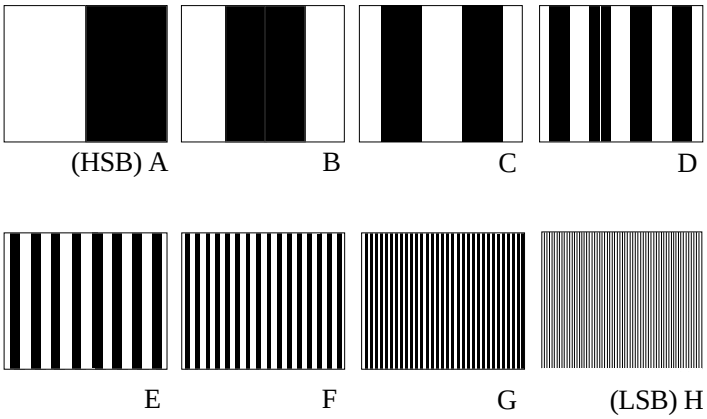

Figure 4.8: 8 bits temporally Gray coded pattern projection.

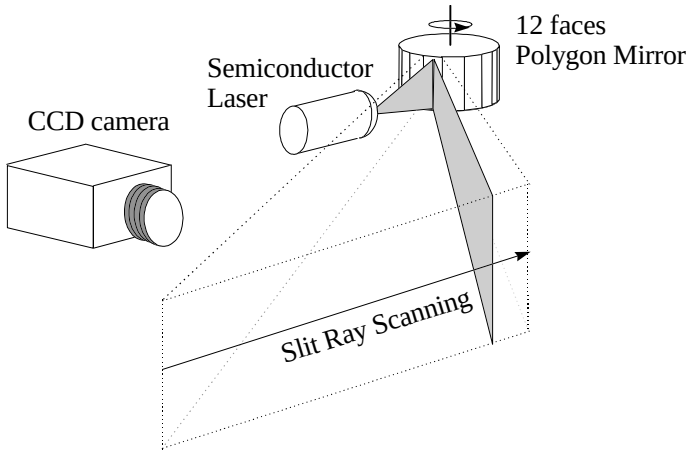

Figure 4.9: The Cubiscope system.

coded pattern, of its positive and negative representation. Then, comparing the intensity image with both representations, a better segmentation of the projected pattern can be obtained.

The problem of a light projector is sometimes a result of heat irradiation onto the scene, and of its big size and weight. In 1995, Hattori et al. [Hatt 95] proposed to replace the light projector with a semi-conductor laser, that gives a high power illumination with low heat irradiation. The proposed system, named Cubiscope, is shown in figure 4.9 and is composed by a CCD camera, a semi-conductor laser and a scanned mirror, synchronously controlled. The main goal of this system is to obtain scene images based on the temporary codification proposed by Posdamer et al. and Altschuler et al.

In 1995, the same temporary codification described by Posdamer et al. and Altschuler et al. in 1982 is again proposed by Müller [Mull 95]. In 1996, Sato [Sato 96] proposed a new moving modulated pattern light and spatio-temporal image processing.

4.2.2 Carrihill-Hummel

The system proposed by Carrihill et al. [Carr 85], in 1985, is based on getting 3D information of any scene from two frame images. The first image is captured with a constant illumination projection on the scene. A linear wedge filter is used to project on the scene a pattern brightly illuminated on

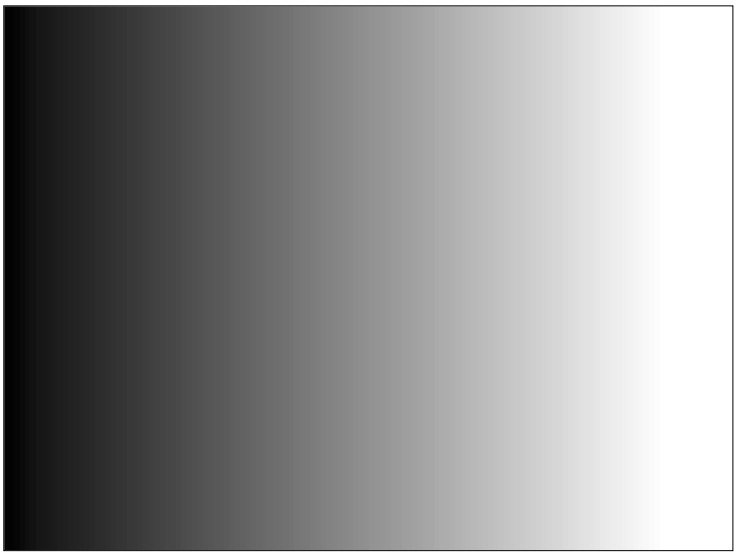

Figure 4.10: The pattern proposed by Carrihill et al.

one side, and half brightly on the other, in order to snap the second image. This pattern is shown in figure 4.10.

For the first projection, it is supposed that the illumination intensity is constant along the x-axis and the y-axis of the projector. For the second projection, illumination has to be a constant along the y-axis. Then, for each captured pixel, the intensity subtraction of the two images, can be calculated. This difference allows to obtain the column of the projector from which the dot has been emitted.

Using a difference ratio for each pixel makes it easier to cancel possible surface reflections or highlights since an illumination ratio change is always produced by a change in the projected illumination.

Using electro-optics filters, the system can provide 3D information at high speed, but, like systems that project more than one image, it is limited to static scenes. Obviously, any moving object in two consecutive frames would produce a correspondence error.

Carrihill et al. affirm that the system provides an accurate measurement with pixels of 8 bits depth (bits per pixel). The test has been limited to 12 bits depth, although Carrihill et al. affirm that there is a good correlation between the sensor perception and reality.

In order to improve the accuracy measurement without increasing the bit depth of the captured image, a new method based on a sawtooth pattern could be proposed. Two sawtooth patterns of different period length are

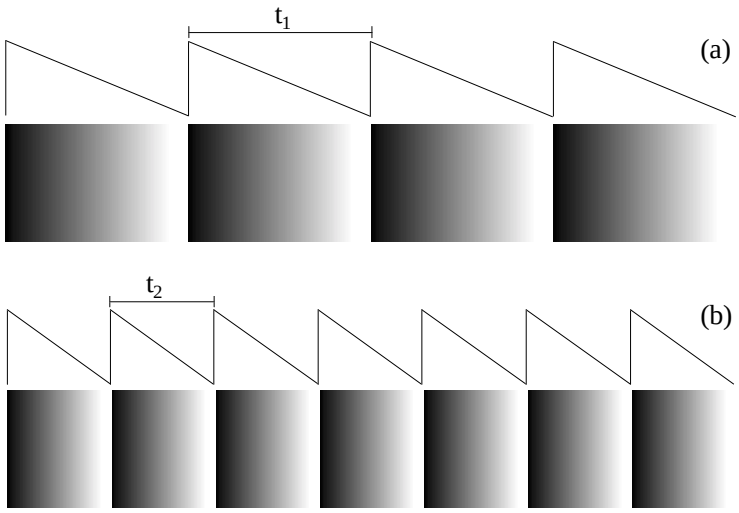

Figure 4.11: The use of the period length to improve the resolution of the measurement without increasing the number of bits depth of the captured image. Obviously, period patterns are limited to measure only surfaces with a depth discontinuity smaller than its period length.

shown in figure 4.11. Obviously, the maximum surface depth discontinuity to be measured is limited to the chosen period.

In 1993, Hung [Hung 93] proposed a grey level sinusoidal pattern. The period of the captured pattern depends on the depth of the surface where it is reflected. Hung proposed to triangulate from the column phase of the imaged point. For each pixel, this phase can be approximately obtained from the light intensity. However, it seems that a robust 3D reconstruction could be obtained projecting first a constant illuminated pattern as propose Carrihil el al. This method, as the sawtooth one, also suffers the limitation of a periodical codification.

4.2.3 Boyer-Kak

In 1987, Boyer and Kak [Boye 87] proposed to illuminate the scene using a single pattern projection. The pattern is coded by vertical coloured slits. The correspondence problem, i.e. given an imaged slit, to obtain from which projector slit it has been emitted, can be solved as the slit codification used to build the pattern, is known. Obviously, as with any other system that projects colour, its use is limited to predominantly colour neutral surfaces,

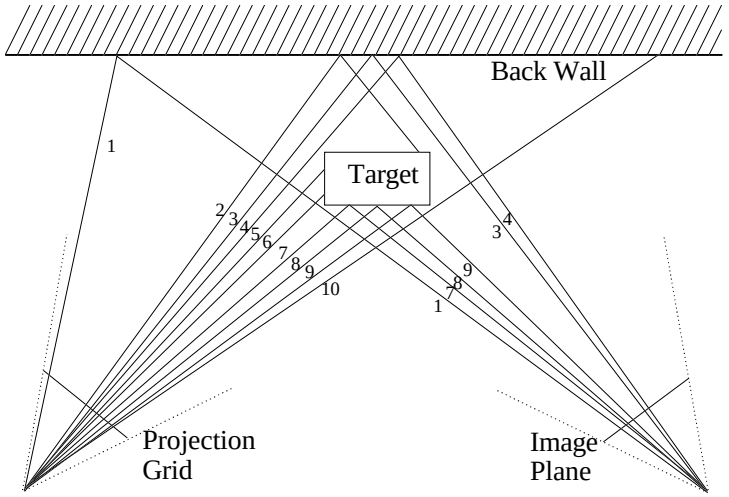

Figure 4.12: An example of the correspondence problem that has to be solved between projected and imaged slits.

as highly saturated hues could produce slit identification errors.

Figure 4.12 shows only a stripe pattern projection up to which the matching problem can be complicated between projected and imaged slits. Note that, slits projected out of the scope of the camera or reflected on occluded surfaces can not be imaged. Also, the slits imaged by the camera can be obtained in a different order.

Boyer et al. proposed to codify the pattern using the three basic colour components: red, green and blue. The pattern will be made by a sequence of vertical slits coloured with any of the three basic components. See figure 4.13. The slit can also be white. Optionally, between each slit, a black gap can be placed, this gap can be used as a slit separator. Note that, white can be easily obtained and identified as the colour which has approximately the same red, green and blue intensity values.

If it is supposed that the pattern is made by n vertical slits, Boyer et al. proposed to divide this pattern in m subpatterns, each one made by k vertical slits, as $n = m \cdot k$. In order to code the m patterns in a unique way, k is a value as large as necessary. As each subpattern is emitted without any beginning and ending code, multiple matchings can be obtained in reception. Actually, we do not know where a subpattern ends and the next begins, and moreover, the slits have been re-ordered as a function of the measured surface discontinuities. An example of the pattern deformation can be shown

Figure 4.13: A detail of the pattern proposed by Boyer.

in figure 4.14, where a 72 slit pattern was emitted and a 50 slit one has been received.

For colour slit detection, Boyer et al. proposed a hardware that detects red, green and blue peaks in the RGB signal given by the camera. Thus, the specific hardware can obtain the index and position of each peak in real-time. In fact, the index can have a proper value for each different colour (for example, the 1, 2 and 3 index values can be associated with red, green and blue colours respectively). Consequently, the position can be obtained from the camera synchronism. Obviously, CAG feature allows the equalisation of the RGB signals.

In order to obtain the subpatterns from the index and position provided by the hardware, a software algorithm is used. In the first step, all the possible matchings between the emitted and received subpatterns are found. In the second one, a region growing algorithm is used to remove all the erroneous matchings. In the third one, for each subpattern not matched, a heuristic algorithm is used to find its better matching. When the third step is completed, a data base, which associates captured and projected subpatterns, has been obtained. Finally, a triangulation process provides the 3D scene information they were looking for.

Monks et al. [Monk 92] [Monk 93] assert that the matching process proposed by Boyer is not optimal, because if the matches for the first vertical slit contain errors, these will be propagated through the rest of the data. Monks

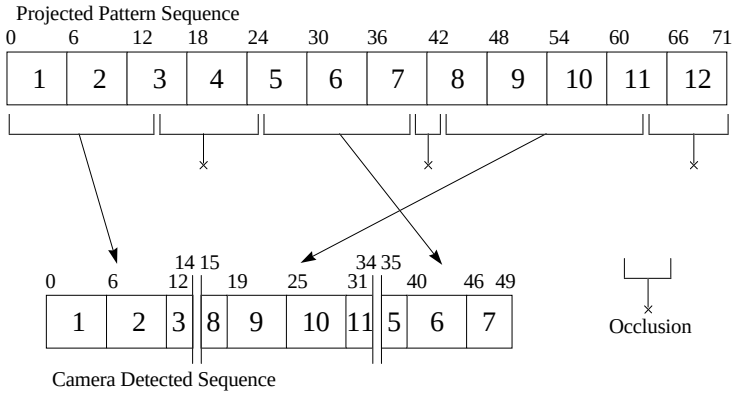

Figure 4.14: A possible relation between a projected pattern and the captured one.

et al. have developed a new technique that eliminates this problem and has the advantage of being non-iterative, encoding the topology of the entire set of stripes as a directed acyclic graph [Bras 90].

As we said before, the pattern proposed by Boyer et al., is limited to measure predominantly neutral colour surfaces, as highly saturated hues could produce slit identification errors. Even so, it is specially recommended in dynamic environments since only one projection is needed.

4.2.4 Le Moigne-Waxman

There are a lot of different patterns that can be generated based on the grid pattern concept. Some examples can be shown in figure 4.15. When a grid pattern is considered, the number of crossing points to be projected has to be chosen, but the line thickness has also to be chosen, since it depends directly on the smoothness of the imaged surface texture. A very thick line will give a low resolution, and, with a very thin one, we will obtain a lot of discontinuities, which will complicate the matching process. Obviously, the thickness of the line has to be chosen knowing the kind of scenes to be measured.

Note that the patterns shown in figure 4.15 are not coded. Le Moigne and Waxman [Moig 84] [Moig 85] [Moig 88] proposed to add dots on the grid, which can be used as landmarks to initiate the decodification or labelling of the projected pattern. In figure 4.16 some grid patterns coded with dots are shown.

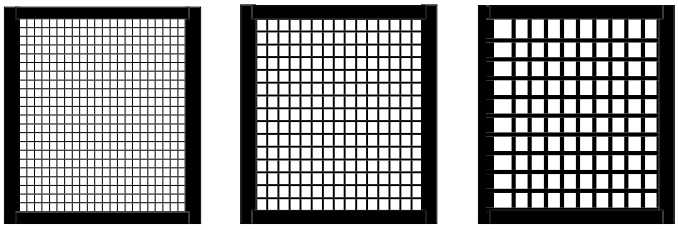

Figure 4.15: Some not coded grid patterns. The pattern resolution is given by the number of crossing points and the line thickness.

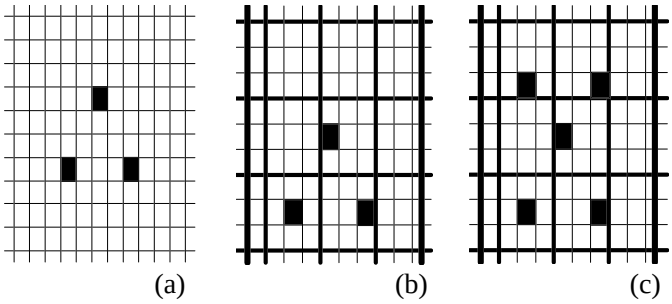

Figure 4.16: Grid patterns partially coded by the position of some dots, which are used as landmarks to initiate the labelling process.

The configuration used for the system locates the projector at a fixed distance from the camera, along the y-axis. This means that vertical lines are imaged nearly without deformation, keeping its natural parallelism and continuity, so they will be easily detectable. A square neighbourhood operator is computed in each imaged pixel to reduce the albedo or highlight effect. The system uses the known location of the vertical lines as guides to search the dots and horizontal line intersections. Then, the horizontal lines are explored to join their discontinuities. Note that, if a lot of discontinuities are found, it is due to the fact that the chosen line thickness is too thin. Finally, for each landmark dot, two different edge labelling processes, using a greedy algorithm, are used. One process starts its labelling from the right side, and the other from the left side of the landmark. So, six different labellings are obtained from three landmark dot patterns. Then, a merging algorithm is used to label the whole pattern. Initially, only two labelled edges are considered. Then, they try to label the not labelled edges with a local interpolation.

Although the labelling algorithm looks like rather complicated, horizontal lines can be highly broken due to depth discontinuities of the measuring surfaces, and they can also disappear partially. So, easily matching errors could be obtained using a simpler labelling process.

The pattern proposed by Le Moigne et al. is specially indicated to work in dynamic environments, as a single projection is needed to obtain 3D information of the scene. But it is true that the resolution given by the pattern is rather limited, basically for two reasons: firstly, vertical lines do not give any depth information since they are only used as searching guides. And finally, the matching process may be rather slow in density grids, so, dynamic high resolution is not permitted. But the utilisation of the proposed pattern as a vision sensor for mobile robot navigation in structured indoor environments is highly recommended. In such environments, the surfaces do not usually have high contrast textures, which can deform the pattern, and the scene is quite regular, and small differences do not give interesting information.

4.2.5 Morita-Yakima-Sakata

In 1988 Morita et al. [Mori 88] proposed a binary pattern of light dots as an M-array. The M-array has the already defined coding window property, as any subpattern into a window exists only once and at one place within a period (for more information see the Vuylsteke-Oosterlinck method).

To simplify the correspondence problem, Morita et al. placed the camera and projector so that the Δ_y and Δ_z distances between them are zero, as shown in figure 4.17. Then, the dot identification process is quite simplified,

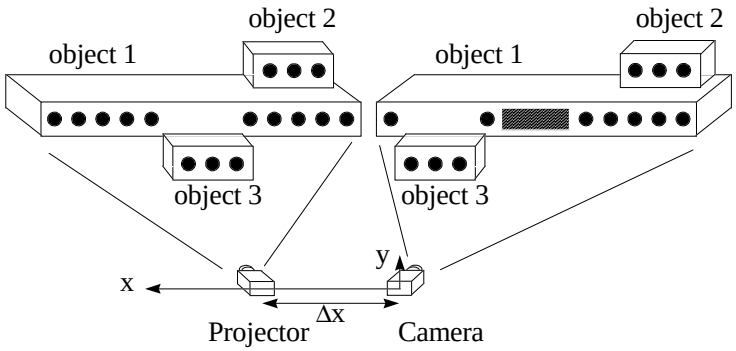

Figure 4.17: The system configuration proposed by Morita et al. with the dot disorders that can be obtained.

as the dot position on the image plane can only move horizontally.

In fact, an M-array pattern is made by a set of circular bright, or dark, dots, represented by the binary number 0, or 1, respectively. From the M-array property, the indices of all the elements of the observed pattern can be uniquely determined relative to the projected pattern if no pattern disorders exist. However, for complex surfaces the following disorders can be obtained:

1.- *Missing dots*: If two or more objects are measured, a dot projection can be reflected on an occluded surface, and, it can not reach the image plane. An example can be seen in figure 4.17 where the points from #2 to #4 are projected on the back surface of the object #3.

2.- *Dot displacement*: If the depth distance between two objects is different, the reflected dots will be imaged with a horizontal displacement between them. As can be seen in figure 4.17, the dots reflected on object 2 move right by one dot on the image plane.

3.- *Dot permutation*: If an object is placed in front of another, the projected dots can swap positions. As shown in figure 4.17, the projected dots form #6 to #8 reach the image plane in the position of #2 to #4, as they have been occluded by object 3.

In order to obtain the image co-ordinates of each dot, this method has to project a whole illuminated dot matrix. Then, a binary (bright and dark) dot matrix is projected, so that each window, with a fixed size, determines the column index from which the imaged point has been projected, see figure 4.18.

The proposed system is quite simple, so that 3D information can be obtained without a lot of computational cost. However, the system is limited

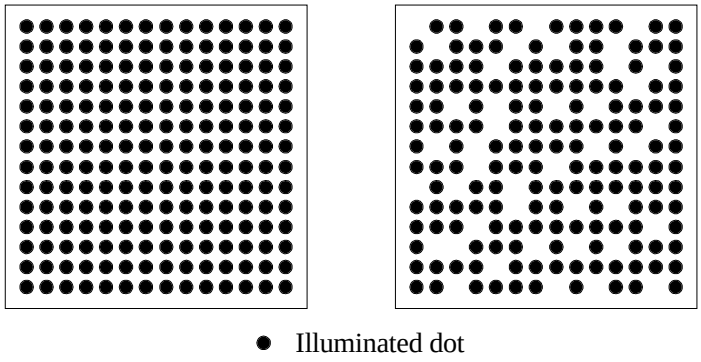

Figure 4.18: An example of the two patterns projected on the measuring scene by Morita.

to static scenes because it needs two projections. As the system projects isolated light dots, not a lot of textural information can be obtained, and, if the dot size is reduced to get more resolution, then, high contrast textural surfaces could modify the dot shape, making its localisation difficult.

Recently, in 1996, Lavoie et al. [Lavo 96] has proposed a binary grid pattern with a pseudo-random dot codification at each grid vertex. The dots are binary coded, that is they can be illuminated or obscured, as the M-array proposed by Morita.

4.2.6 Vuylsteke-Oosterlinck

A model with a dynamic pattern, column coded, is proposed [Vuyl 90]. The basic structure of the pattern is like a regular chess-board, alternating bright and dark squares. Then, the pattern is modulated overlapping a bright or dark spot at every square vertex (figure 4.19), so that, any square of the regular chess-board pattern carries an additional information bit, that together with the neighbouring bits, will be used to code each column.

In order to identify each dot of the coded pattern, they are represented in binary. The dot gets the "1" value if it comes from a bright dot. Otherwise, it takes the "0" value, associated with dark. Additionally, the chess-board allows finding out naturally if the coded dot comes from a bright or dark square, represented by the symbols $-$ and $+$ respectively. Then, four different combinations, shown in figure 4.20, can be obtained.

Following the chess-board, the code assignment is based on two binary sequences, represented by c_k and b_k of 63 bits length. The dots of all the k

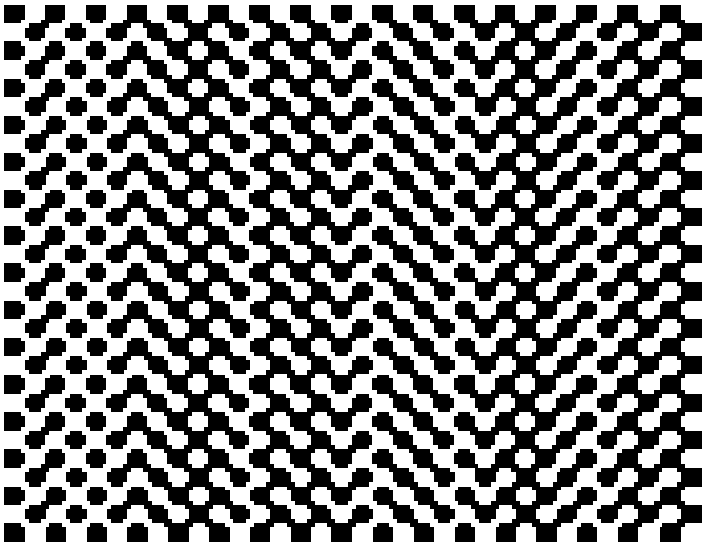

Figure 4.19: A detail of the pattern proposed by Vuylsteke et al.

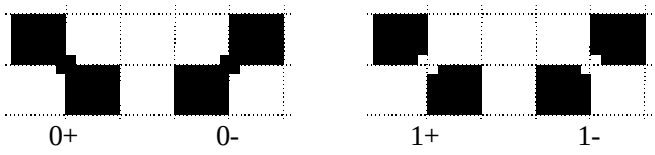

Figure 4.20: The pattern is made by four different combinations. The top left square is represented by the symbols - and +, which represent a bright or dark square respectively. The binary numbers 0 and 1 are used to code the dot at the square vertex, which can also be dark or bright respectively.

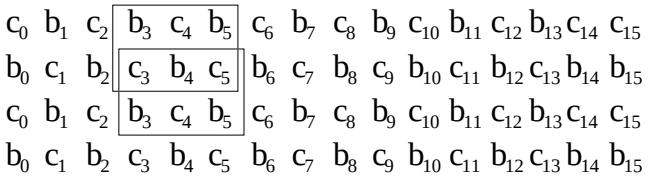

Figure 4.21: The code assignment determines that any 2x3 window can identify the column index Each window is made by a triplet of c_i values and another triplet of b_i. In the example, for the two windows, the column is given by the same sequence order $((c_3, c_4, c_5), (b_3, b_4, b_5))$.

column which corresponds to a + square type are assigned the binary value c_k, while those of the − type are assigned the value b_k. A 6 bits length code is needed to code the 63 different columns of the pattern. Then, any window of 2x3 squares allows coding any column index. In fact, any window with 6 squares size can be used, but obviously, a compact window is less affected by surface discontinuities that an elongated one. This is the main reason why a 2x3 window has been chosen instead of a 1x6.

Each 2x3 window has a triplet of c_k codes and another of b_k, easily differentiable. The sorting of these codes gives a 6 bits length number which codifies a column index of the pattern. An example for the column $k = 3$ is shown in figure 4.21, where the code number, which represents column 3, is given by the sequence: $((c_3, c_4, c_5), (b_3, b_4, b_5))$.

The code assignment can not be chosen independently due the fact that the identification windows overlap. Then a kind of recursive code generation sequence, with an initial birth polynomy, has been used, which allows the obtaining of a desired length period repetition to code in a unique way all the columns of the pattern.

The pattern presented has basically two limitations. The first one is based on the difficulty to measure high textural surfaces, which produce partial lost of regions of the pattern. And the second one is associated with the surface orientation. If these are not perpendicular to the optical axis of the projector, a deformation of the projected pattern is presented, which from a determined angle orientation, does not allow the identification of the pattern. However, the pattern is well-recommended to measure dynamic surfaces where only one projection is permitted.

In 1995, Pajdla [Pajd 95b] has reimplemented the same pattern, explaining the calibration process. Pajdla has proposed, as an improvement, to use

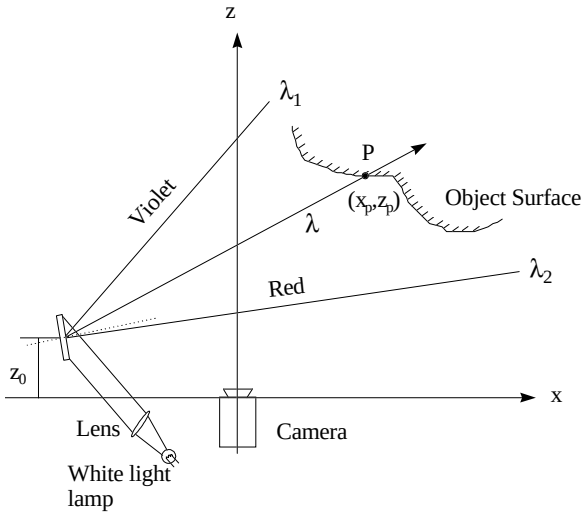

Figure 4.22: The principle of diffracting white light in order to obtain a rainbow pattern.

a hexagonal codification instead of a square one. In this case, the windows size is reduced and as a result, the number of not indexed columns due to depth discontinuities is decreased. However, the identification step is more complicated.

4.2.7 Tajima-Iwakawa

The system proposed by Tajima [Taji 90] in 1990 is based on the vertical slit coding technique, where each slit is emitted with a different wavelength. The projected pattern is like a rainbow pattern, as the whole colour spectrum, from blue up to red, is projected. This can be obtained diffracting white light as shown in figure 4.22. The proposed pattern is shown in figure 4.23.

Depth can be obtained using the triangulation principle, but first the slit angle (the xp_1 co-ordinate) which has produced the colour pixel imaged on the image plane, has to be known. In fact, the slit emission angle can also be determined knowing the slit wavelength.

The objects illuminated by the rainbow pattern are imaged with a single monochromatic camera instead of a colour camera. Two different colour filters placed in front of the camera are used, and two images of the scene are captured. For the same point, the intensity relation between the two images does not depend on the illumination, nor on the colour object. Tajima et al.

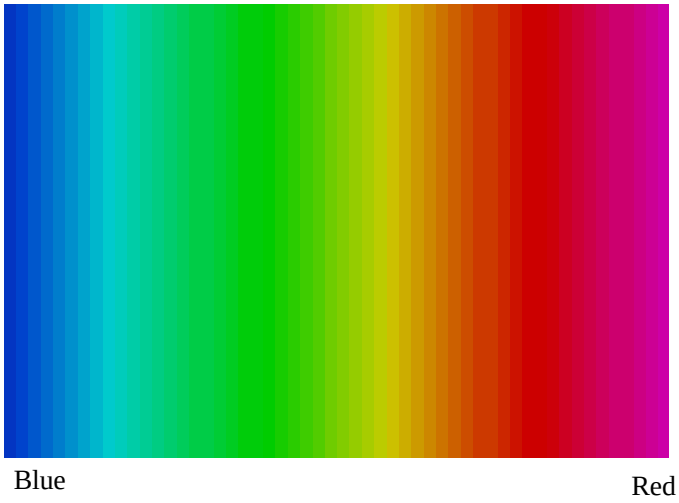

Figure 4.23: The rainbow range pattern projected by Tajima et al. made by a set of vertical slits which use all the colour spectrum from blue up to red.

have proved that this intensity relation depends directly on the wavelength slit (λ).

As the system needs two frames for each measurement, it is limited to static scenes, but obviously the system can measure dynamic scenes using a colour camera. In 1996, Geng [Geng 96], seemingly without knowing the work already done by Tajima et al., proposed the same rainbow pattern in order to obtain 3D information from the measuring surface. However, the initial idea proposed by Tajima et al. was improved by Geng using a CCD colour camera and using a Linear Variable Wavelength Filter (LVWF). Tajima's idea improved by Geng can be shown in figure 4.24.

In this case, the projector emits a white light plane, which, using a cylindrical lens, generates a continuous fan of planes. From the natural design of the LVWF, a continuous colour spectrum is obtained, so there are not two light planes projected on the scene with the same wavelength. Obviously, the system resolution depends on the camera features to distinguish among different colours, or different wavelengths.

Recently, Smutny and Pajdla [Smut 96] have reimplemented the system proposed by Tajima et al. With regard to the limitations of the rainbow pattern, they said that the surface can have any colour, but has to be opaque. Obviously, no other wavelength not coming from the projector, can be emitted on the scene. The scene has to be light controlled and the measuring

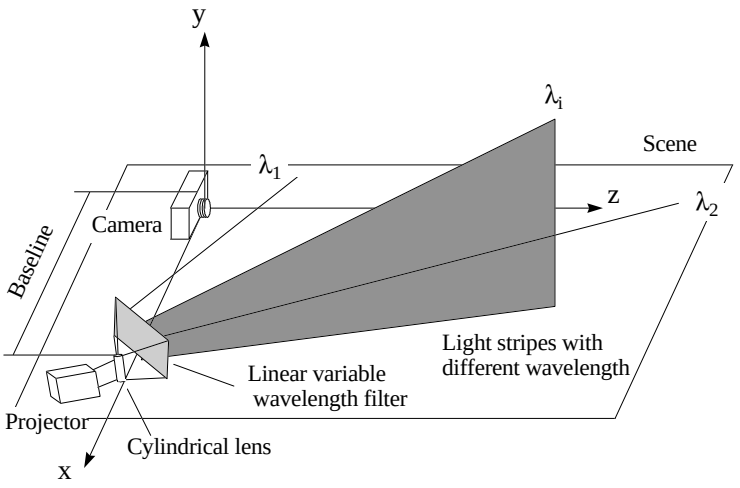

Figure 4.24: The system proposed by Geng.

objects can not be fluorescent, nor phosphorescent.

4.2.8 Wust-Capson

In 1991, Wust and Capson [Wust 91] proposed the projection of a sinusoidal intensity pattern on the measuring surfaces. The proposed pattern is made by the overlapping of three sinusoids, of n periods along the x-axis. Each sinusoid is associated with each primary colour (red, green and blue). The greens sinusoid is shifted 90° with respect to the red, and the blue is shifted 90° with respect to the green, as shown in figure 4.25. The pattern is column coded, so all the rows are identical, resulting a colour vertical fringe pattern, as shown in figure 4.26.

Instead of obtaining the column index by decoding the imaged pattern, Wust and Capson proposed to obtain the depth directly from the wave phase shifting. This technique is widely used in Moire methods to measure continuous surfaces [Reid 86].

Some limitations can be observed in the method. The scene has to be predominantly colour neutral, in spite of colour projection. As the pattern is made by periodical fringes, it is limited to measure surfaces without discontinuities larger than a fringe period. According to Wust and Capson, camera response depends on the frequency of the emitted fringes, and the histogram equalisation used to compensate for the non-linear intensity response also

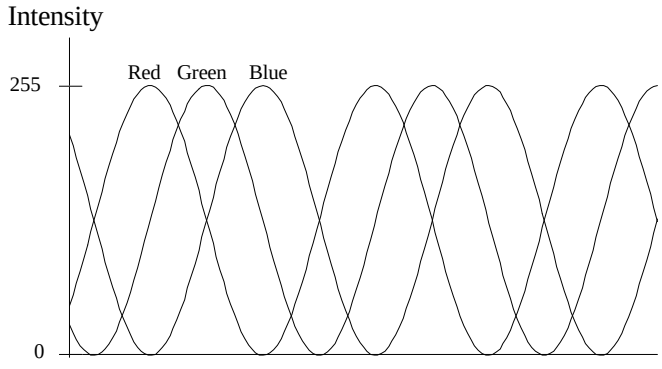

Figure 4.25: The principle of overlaping three sinusoidal intensity patterns.

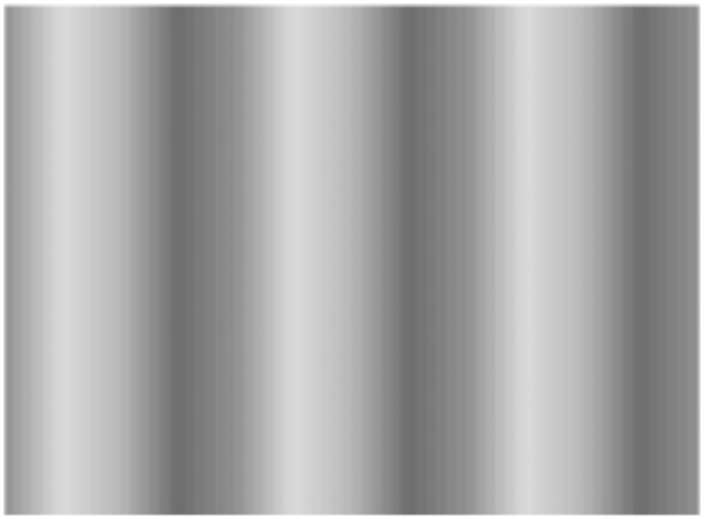

Figure 4.26: The periodical pattern proposed by Wust and Capson, made by the sinusoidal overlapping of the three primary colour components.

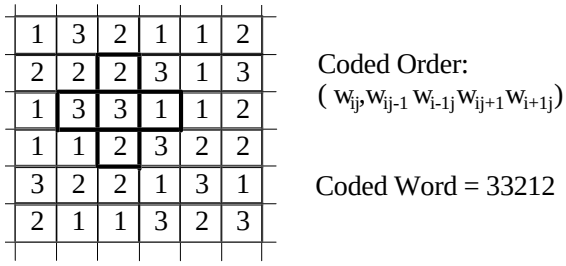

Figure 4.27: Dot codification example using its four neighbours and a basis equal to 3, i.e. only three different symbols can be used.

produces some measuring errors. Improving these aspects the system can obtain a better resolution and robustness.

4.2.9 Griffin-Narasimhan-Yee

Griffin et al. [Grif 92], in 1992, have carried out a mathematical study about which should be the largest size allowed for a coded matrix dot pattern. It is supposed that: 1.- A dot position is coded with information emitted by itself and the information of its four neighbours (North, South, East and West). 2.- There can not be two different dot positions with the same code. 3.- The information is determined using a fixed basis, which determines the symbols used to code the matrix. 4.- The biggest matrix, which gives a better resolution, is desired.

If a basis equal to 3 is done, a possible dot codification is shown in figure 4.27.

Griffin et al. proved that, given a basis b, the largest matrix (the biggest nxm matrix) can be obtained from its largest horizontal vector (Vhm), and its largest vertical vector (Vvm). Vhm is a vector composed by the sequence of all the triplets of numbers that can be build without repetition using a b basis. Vvm is a vector made by the sequence of all the pairs of numbers that can be obtained without repetition. Then, the first row of the matrix is given directly by Vhm,

$$f_{0i} = Vhm_i \tag{4.1}$$

and, the other matrix elements can be determined applying the equation 4.2 where i is the row index and varies from 0 to the Vhm length, and j is the column index and varies from 0 to the Vvm length.

$$f_{ij} = 1 + ((f_{i-1j} + Vvm_j) \ mod \ b) \tag{4.2}$$

For example, if a basis equal to 3 is supposed, then its largest vectors are:

$$Vhm = (331321311231221211133232223 33) \tag{4.3}$$
$$Vvm = (3121132233) \tag{4.4}$$

So, the obtained matrix is,

331321311231221211133232223 33
331321311231221211133232223 33
112132122312332322213213331 11
331321311231221211132132223 33
112132122312332322213213331 11
223213233123113133321321112 22
223213233123113133321321112 22
112132122312332322213213331 11
331321311231221211133232223 33
331321311231221211133232223 33
331321311231221211133232223 33

After the coded matrix is found out, a different projection can be associated for each value, that is, for each number which belongs to the interval $\{1, b\}$. For example, a coloured dot pattern can be obtained if the red, green and blue colours are associated to the 1, 2 and 3 numbers respectively, obtaining a pattern like the one shown in figure 4.28.

The resolution of the pattern can be increased by simply increasing the basis value. Depending on the colour discriminating capability of the system employed to visualise the scene, almost any degree of resolution can be obtained.

In many applications, the scene is not built by colour neutral surfaces. A simple reason could be that the imaging system used is only able to capture monochromatic images. Then, monochromatic light has to be projected on the scene. In this case, each number of the dot matrix can be changed to geometric form. An example is shown in figure 4.29.

The method proposed by Griffin et al. is the unique method studied by which, from the decodification of the pattern captured by the camera, for each

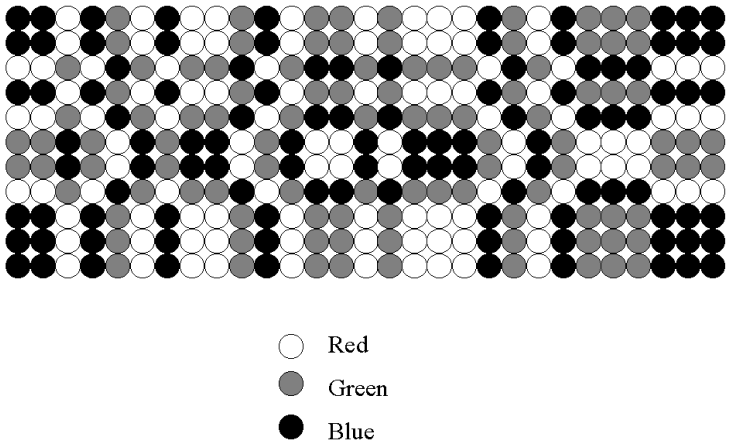

Figure 4.28: A possible coded dot matrix, obtained by the method proposed by Griffin et al. A basis equal to 3 has been supposed and to each symbol a coloured dot has been associated.

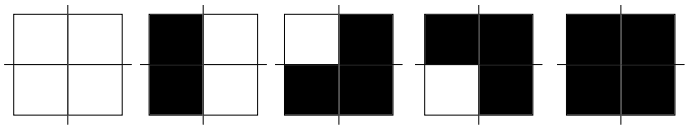

Figure 4.29: A possible geometric association of a matrix dot coded using a basis equal to 5.

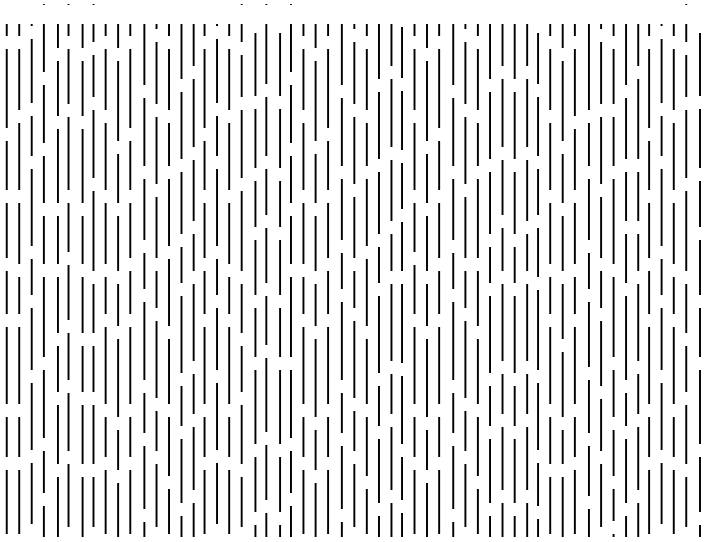

Figure 4.30: The pattern proposed by Maruyama et al. made by the projection of multiple slits with random cuts.

image point (x_{p2}, y_{p2}), the projector position point (x_{p1}, y_{p1}) from which it has been emitted, can be known. As shown in the mathematical section dedicated to surface measuring, it is not necessary to know both projector co-ordinates. Then, the pattern can be obviously simplified to obtain a single row coded or column coded pattern, but we could possibly obtain a pattern which uses the windows proposed by Vuylsteke et al. [Vuyl 90], or the coloured slits proposed by Boyer and Kak [Boye 87].

In 1996, Davies et al. [Davi 96] re-implemented the coloured dot projection system proposed by Griffin et al. However, as an improvement, Davies proposes the use of a specially developed formulation of the Hough Transform in order to extract the imaged dots of the projected pattern.

4.2.10 Maruyama-Abe

The method described by Maruyama et al. [Maru 93], in 1993, is based on the projection of multiple vertical slits. The slits are coded from the position of some random cuts, as shown in figure 4.30.

All the random cuts are placed to obtain short lines, satisfying the condition that its length is in the interval $[L_0 - D, L_0 + D]$, where L_0 is a standard length and D is given by a random number.

Segment matching is performed only based on the correspondence of the

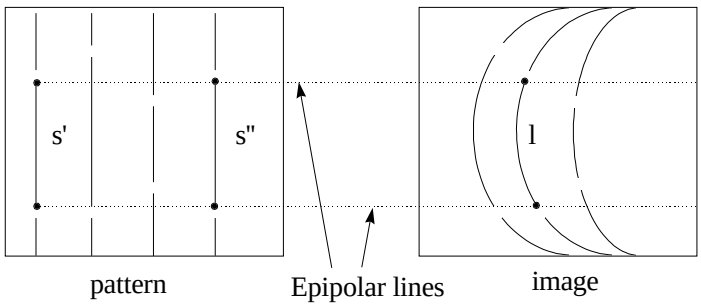

Figure 4.31: A multiple matching example. In this case the end points of two short lines, s' and s'' of the pattern, match the end points of a line projected on the image plane, along its epipolar lines.

short line end points along the epipolar lines. As the cuts are randomly distributed, more than one pattern line can be matched with a given image segment. An example of multiple matching is shown in figure 4.31. Then, the imaged segments are classified into three groups: 1.- Segments with no competitors, as they have only one matching. 2.- Segments that can be simply identified using adjacency relations between itself and its neighbours. In this case, the matching can not be done due to the noise on the image or due to a surface depth discontinuity, which has changed the line length. 3.- Otherwise.

Maruyama et al. consider that the epipolar lines can always be horizontally computed, so that only the x-axis has to be examined to find out all the lines on the pattern which match with the imaged segment. This is not always true, as epipolar lines depend on the projector position and orientation, with respect to the camera axis, and Maruyama et al. do not impose any restriction. But, it is true that the image can always be transformed to get horizontal epipolar lines, as they said in their work.

In order to remove erroneous matching, the information of neighbour segments is taken up. Finally, with the aim to obtain the correspondence of all the segments from groups 2 and 3, a region growing algorithm matching from adjacency constraints is used.

To improve the system, we could think of using a pattern with an intelligent cut distribution which matches the same imaged segment, along its epipolar lines (in order to not obtain more than one matching). So, a bijection between captured image and projected pattern is determined. But, as

Maruyama et al. said, if the adjacency constraints are used, the matching can be done using a region growing algorithm. However, surface depth discontinuity or noise problems in the image could provide segments which do not have a matching line. In this case, a region growing algorithm has to be used to match them.

The method is suitable for measuring 3D objects with relatively smooth surfaces. For objects with a lot of discontinuities and with highly textured surfaces, the method will have some difficulties. Obviously, line discontinuities produced in the captured segments complicate the matching process. However, the pattern is perfectly used to measure dynamic scenes, as only one projection is required.

4.2.11 Ito-Ishii

In 1995, Ito and Ishii [Ito 95] proposed a three-level checkerboard pattern. Each 2D pattern point or node has a square form and it is surrounded by four projected squares which can be coded using only three different grey levels. The grey levels used are: obscured ($intensity = 255$), half illuminated ($intensity = 128$) and illuminated ($intensity = 0$), which have been coded as 2,1 and 0 respectively. Ito et al. defined a subcode of a node as the clockwise combination of the codes of the four adjacent nodes. See figure 4.32. They assert that the quantity of codes is large enough to avoid false matchings between two spatially coded patterns. Although pattern generation without repeated appearance of the same feature code values is a further research subject, it is not a problem because the epipolar constraint eliminates confusion. Ito et al. affirm that the method provides high spatial resolution and makes it possible to take highly accurate measurements of stable and moving objects. Indeed, a unique pattern projection is needed to infer 3D scene information. However, it seems that the measuring of highly saturated surfaces may complicate the segmentation of the pattern, as it could be difficult to distinguish the three different grey levels.

4.2.12 Chen-Hung-Chiang-Wu

In 1997, Chen et al. [Chen 97] proposed the projection of a pattern made by coloured stripes. The approach is most similar to the one proposed by Boyer et al. [Boye 87]. However, Chen proposed to code each slit independently from the others. Firstly, Chen generates a sequence of coloured slits $C_1, C_2, C_3, ..., C_n$. Then, in order to increase the intensity variation between adjacent slits, a black slit B is inserted between them, obtaining the sequence $(C_1, B, C_2, B, C_3, ..., B, C_n)$. Each slit is code without repetition varying its

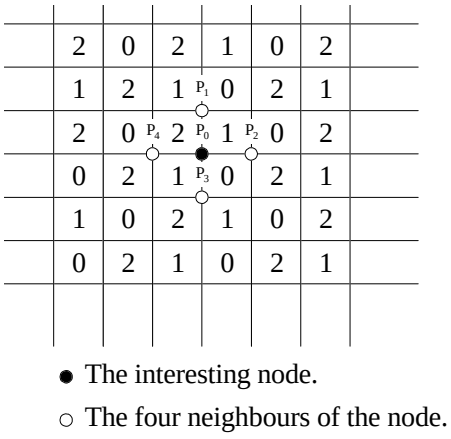

Figure 4.32: Spatial main codes specifying nodes P_0, P_1, P_2, P_3 and P_4 are (2101),(1012),(1020),(1012) and (0212). The subcode of P_0 is (1012,1020,1012,0212).

HSI representation. The intensity value are randomly generated in the range $[I_{min}, I_{max}]$. It has been fixed greater than I_{min} to ensure an easily edge extraction, and smaller than I_{max} in order that the hue value carries sufficient color informaton. Note that high intensity values reduce the coloured information going to white. Actually, the hue value is the most important and is generated applying equation 4.5,

$$\begin{aligned} H_o &= H_{ini} \\ H_{i+1} &= (H_i + H_{jmp}) \; mod \; 360^o \end{aligned} \tag{4.5}$$

where, Chen et al. proposed to fix $H_{ini} = 0$ and $H_{jmp} = 140$. The two values have been found by experimentation testing all the possible combinations and selecting the best set of parameters. Furthermore, in order to fix the saturation value for each slit, all the colours HI of the set $C = \{C_1, C_2, C_3, ..., C_n\}$ have been transformed to the RGB representation. The largest saturation value which satisfies that (I_i, H_i, S) is valid in the RGB space, is then selected as the saturation of C_i.

However, the major contribution of Chen et al. is the proposal of using the coloured pattern only to solve the correspondence problem between two imaging sensors. It is known that, if a single camera is used, it is necesary to find colour correspondence between the projected pattern and the grabbed

one. In general, due to different reflection properties of 3D surfaces, the colour recorded by the camera is different from the one projected by the light, which difficults to solve the correspondence. Chen et al. proposed to use two imaging sensors as a stereo vision system, the pattern of light is projected on the scene and captured at the same time by both cameras. The colour information of the imaged slits is only used in order to solve the correspondence problem between both imaging sensors.

This approach has the advantage that the projector does not need to be calibrated. However, one of the major constraints is that both cameras have to be synchronised in imaging. Furthermore, the position of the cameras in the scene and the optical behaviour of them, may give a different imaged colour between both cameras which rather constraints the principle proposed by Chen et al.

4.3 Summary

In order to provide an easier overview of the techniques presented, we summarise them in the following chronological list, relating also the authors who used or improved the techniques lately.

1982 Posdamer - Altschuler. A temporal space-encoded projected beam system [Alts 81] [Alts 87] [Posd 82].

Also proposed by: Mundy and Porter [Mund 87] and Minou et al. [Mino 81].

Lately used: Inokuchi et al.[Inok 84] in 1984, Sato et al.[Sato 86] [Sato 87] in 1986 and 1987, Hattori et al.[Hatt 95] and Müller [Mull 95] in 1995.

1985 Carrihill - Hummel. An intensity ratio pattern projection [Carr 85].

Lately used: Hung [Hung 93] in 1993.

1987 Boyer - Kak. A stripped pattern coded by coloured slits [Boye 87].

Lately used: Monks et al. [Monk 92] [Monk 93] in 1992 and 1993.

1988 Le Moigne - Waxman. A grid pattern with some landmark dots [Moig 84] [Moig 85] [Moig 88].

1988 Morita - Yakima - Sakata. An M-array pattern projection system [Mori 88].

Lately used: Lavoie [Lavo 96] in 1996.

1990 Vuylsteke - Oosterlinck. A single binary-encoded chess board pattern [Vuyl 90].

Lately used: Pajdla [Pajd 95b] in 1995.

1990 Tajima - Iwakawa. A rainbow pattern [Taji 90].

Lately used: Geng [Geng 96] and Smutny et al. [Smut 96] in 1996.

1991 Wust - Capson. A colour sinusoidal pattern [Wust 91].

1992 Griffin - Narasimhan - Yee. A mathematical study about encoded patterns [Grif 92].

Lately used: Davies et al. [Davi 96] in 1996.

1993 Maruyama - Abe. A multiple slits with random cuts pattern projection [Maru 93].

1995 Ito - Ishii. A Three-Level Checkerboard Pattern. [Ito 95].

1997 Chen-Hung-Chiang-Wu. A uniquely encoded coloured striped pattern [Chen 97].

Chapter 5

A New Coded Pattern Projection

This chapter presents a new robust coded pattern projection. 3D measurement is based on the projection of a coloured grid pattern on the scene. The pattern is coloured-coded in order to solve the correspondence problem by a single shot and without using geometrical constraints. The technique proposed permits a rapid and robust 3D scene measurement of dynamic scenes. The chapter also explains the calibration method we have used and shows some experimental results of 3D measurement.

5.1 Introduction

Our aim is to present a new approach in order to solve the correspondence problem in the measurement of 3D data in dynamic (i.e. moving) scenes. It is known that range information can be obtained by triangulation using a system based on two cameras. However, two problems are presented. Firstly, the problem of measuring mostly continuous surfaces, where few matching points are presented, and even the problem of measuring high textured surfaces, where it becomes nearly impossible to solve the matching (see the previous chapter for a detailed explanation). Secondly, the problem of synchronising both cameras to snap the moving image at the same time. We propose a new coded structured light technique to solve the correspondence problem using a single camera. The technique is based on obtaining 3D scene information from a single pattern shot projection, then it could be used to measure static and dynamic objects. We have taken into account that the projecting pattern must be segmented in an easy and robust way, allowing us to solve the correspondence problem efficiently without high computing

requirements.

5.2 Pattern Design

The pattern is composed by the single projection of a grid made by n x n orthogonal slits. The slits are coded in such a way that each slit with its two neighbours appears only once in the whole pattern. Vertical slits are coded using different colour primitives from the horizontal ones. This fact permits to segment the vertical slits and the horizontal ones separately, fact that reduces considerably the computing time. Obviously, vertical slits are column coded (X axe), and the horizontal ones are row coded (Y axe). This codification allows us to code: a) each slit on a single axe, and b) each crossing point on both axes. It is known that, with the goal to obtain a more robust codification, the projection of slits is more interesting than the projection of dots. Each imaged cross-point can be decoded obtaining the (X,Y) position from where it comes, and the imaged slit segments can be decoded knowing whether it is vertical or horizontal, and, what is more interesting, knowing one of its two co-ordinates.

The codification used allows us to place the pattern projection system in any orientation with respect to the camera. It is known that a column coded pattern has to be placed with an horizontal displacement from the camera. Note that row coded patterns have to be placed with a vertical displacement. In our case, we do not have to take care of the location of the camera and the projector to ensure a determined axe displacement. A schematic draft of the measuring system is shown in figure 5.1.

5.2.1 Colour Assignment

In this section we describe the algorithm used in order to assign colour primitives to a sequence of slits. We have used a codification similar to the one used by Griffin et al. [Grif 92]. Our interest is that each slit with its two neighbours exists only once in the whole sequence, such a grouping is called triplet. Firstly, we have to define the set of colour primitives we have in order to colour each slit. If P is the set of primitives, $P = \{1, 2, ..., p\}$, and for instance, 1 = red, 2 = green, 3 = blue, etc., then, we have to define the number of slits of the sequence to be coded. And S is the sequence of slits $S = (s_1, s_2, ..., s_n)$. However, if the number p of primitives has been fixed, we have to know the maximum sequence of triplets given by p primitives in order to know whether the whole sequence S can or can not be coloured. And the maximum sequence of P is defined as $V = (v_1, v_2, ..., v_m)$.

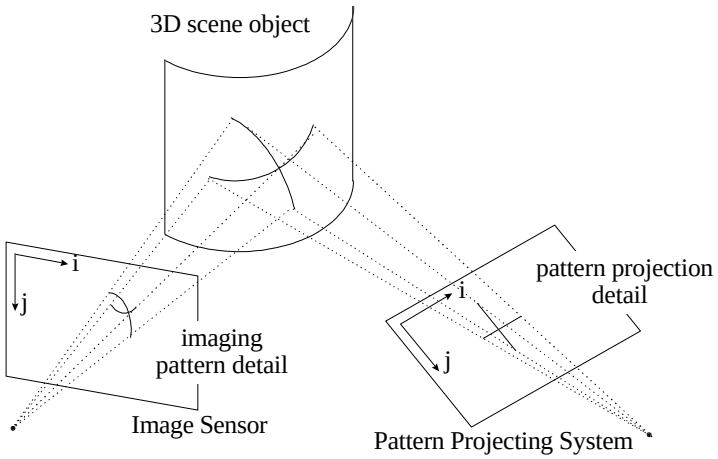

Figure 5.1: Schema of the 3D measuring system based on coded grid pattern projection.

Let us examine the problem more in detail. Given a set of colour primitives P, we wish to build a sequence of colours so that each triplet of adjacent colours in the sequence is different from all the other triplets in order to obtain a unique codification. Since there is an overlap of the triplets in the sequence, the triplets are dependent. The algorithm which allows us to obtain the sequence must take these requisites into account.

The problem can be solved using the graph theory. Given a p basis we can obtain a set nodes, where each node is defined by a triplet of three elements of P. As we have to look after the order and only three elements can be presented in the node, we will obtain $VR_p^3 = p^3$ nodes. A node is defined by a triplet $ijk \mid ijk \, \epsilon \, P$. At this point, we can construct the graph. Each ijk node will have p inputs and p outputs, except the nodes where $i = j = k$ which will have $p-1$ inputs and $p-1$ outputs. Then, the problem is reduced to construct the sequence of triplets by visiting all the nodes of the graph only once. Note that we can go from node $N = ijk$ to node $N' = i'j'k'$ only if $j = i'$ and $k = j'$.

Let us illustrate the problem using an example. Consider $p = 2$, which gives $VR_2^3 = 2^3 = 8$ nodes, that is (111), (112), (121), (122), (211), (212), (221), (222). The following figure 5.2 shows the relation among the nodes.

Then, the algorithm to be used is reduced to visit all the nodes of the graph only once. We can start at any node. If we consider as starting

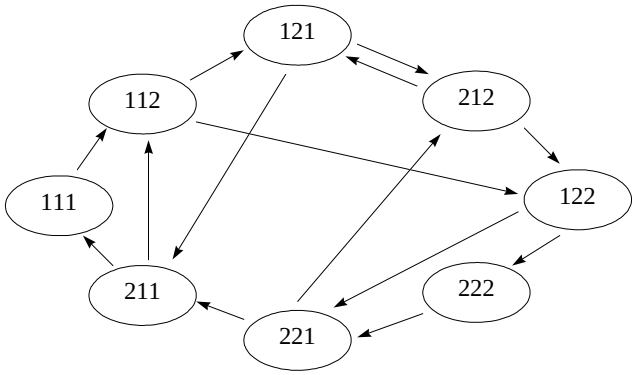

Figure 5.2: The graph obtained from $p = 2$.

node $N = (111)$, we obtain the path { (111), (112), (122), (222), (221), (212), (121), (211) }, which gives the maximum sequence of primitives $V = (1, 1, 1, 2, 2, 2, 1, 2, 1, 1)$. The problem can be solved by a simple algorithm based on the backtracking principle [Bras 90]. However, Griffin et al. studied the problem in detail and they proposed an iterative algorithm, which is the following,

$$\begin{aligned} V \;=\; & \{pp1, p(p-1)1, ...p11, \\ & (p-1)p1, ..., (p-1)11, \\ & , ..., \\ & 2p1, ..., 211, 1\}, \\ & \{pp2, p(p-1)2, ..., p22, \\ & (p-1)p2, ..., (p-1)22, \\ & , ..., \\ & 3p2, ..., 322, 2\} \\ & ... \\ & \{pp(p-1), p(p-1)(p-1), p-1\}, \\ & \{ppp\}. \end{aligned} \tag{5.1}$$

As an example of a coloured codification we have computed V varying p. Consider the following outputs: for $p = 1$ then $V = \{111\}$; for $p = 2$ then $V =$\{221, 211, 1, 222\}; for $p = 3$ then $V = \{331, 321, 311, 231, 221, 211, 1, 332, 322, 2, 333\}$ and $p = 4$ then $V = \{441, 431, 421, 411, 341, 331, 321,$

311, 241, 231, 221, 211, 1, 442, 432, 422, 342, 332, 322, 2, 443, 433, 3, 444 }. Removing the commas from V ,forms a sequence of colour primitives. The following algorithm allows us to obtain the triplets given by equation 5.1, we have only to fix p.

```
c=1;
for(k=1;k<=p-1;k++)
{
  for(j=p;j>=k+1;j–)
    for(i=p;i>=k;i–)
    {
    V[c++]=j; V[c++]=i; V[c++]=k;
    }
  V[c++]=k;
}
V[c++]=p;V[c++]=p;V[c++]=p;
```

As an example of coloured codification we have made a pattern using 6 colours. What is important is that we must choose well-spaced colours in the HSI cone, as they will be the best segmented ones. That is why we have decided to colour horizontal slits in red, green and blue ; and the vertical slits in magenta, cyan and yellow. The slit sequence is obtained by applying equation 5.1 using a p basis equal to 3, that is $P = \{1, 2, 3\}$. Then, the primitives have been substituted by 1 = red, 2 = green and 3 = blue to code horizontal slits; and 1 = magenta, 2 = cyan and 3 = yellow to code the vertical ones. This codification gives us a unique encoded grid of 29 x 29 orthogonal slits. The slits are constantly spaced in a 512 x 512 image with a pixel depth of 8 bits, like the one shown in figure 5.3.

The resolution of the pattern may be increased by just changing the p basis, i.e. using more colour primitives. We can even increase the resolution reducing the spacing between slits, producing also a reduction of the area illuminated in the scene. The size of the image can also be changed producing patterns such as 128 x 128 pixels, 256 x 256 pixels, 512 x 512 pixels, or even not square matrix such as 433 x 584. So, we can take advantadge varying the distance among slits or incerasing the size of the image in any axe. Even the wideness in pixels of the slits is an important constraint in order to measure the scene accurately without obtaining a lot of discontinuities.

As an example, we have decided to project a 512 x 512 image, with a pattern of 29 x 29 slits made using a p basis $p = 3$, and a constant spacing between segments of 16 pixels in both directions. Finally, the whole pattern is centred in the image, that is with a space from each side of 32 pixels.

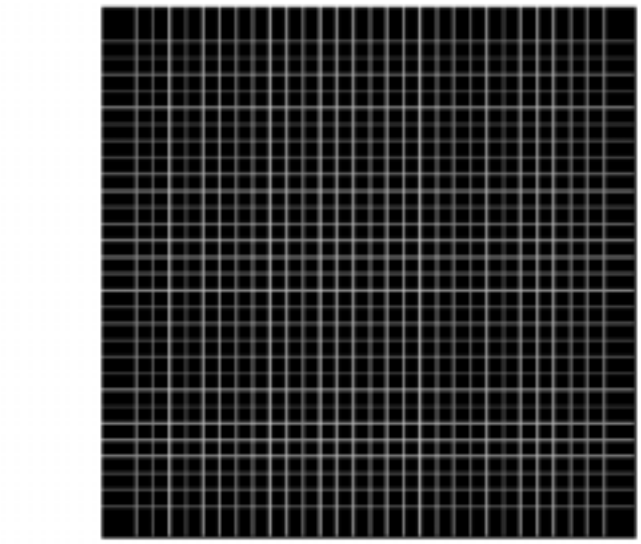

Figure 5.3: The coloured pattern.

In the aim of testing the proposed pattern, a lab scenario has been set (see figure 5.4). The system is composed by a RGB camera, a computer and an electronic slide projector. In figure 5.4, the camera calibrating pattern is also shown. A 512 x 512 pixel resolution RGB image is made by the computer. The image is projected on the measuring scene using the electronic slide projector which takes the image from the VGA computer output.

5.3 System Calibration

The calibration method is based on obtaining the intrinsic and extrinsic parameters of the camera model and the projecting system model knowing the co-ordinates of the 3D object points. In fact, the problem is reduced to the computation of the equations which model the relation between the 3D object points (X_w, Y_w, Z_w) and their 2D observable correspondence point (X_i, Y_i). Camera and projector devices are modelled using the same equations as both have the same geometrical behaviour. Then, we will describe the calibration algorithm without taking into account if it is used to calibrate the camera sensor or the projector system. However, the section will further explain how we have calibrated the camera and the projector separately. Note that both have used the same algorithm of calibration.

Figure 5.4: The scenario of the 3D measuring system.

5.3.1 Describing the calibration algorithm

The linear relation between a 3D object point and its 2D projection is modelled by a $3x4$ transformation matrix. This matrix contains the 6 extrinsic parameters defined by the three rotation angles (α, β, γ), expressed as a $3x3$ rotation matrix $\mathbf{R}$, and the translation vector $T = (t_X, t_Y, t_Z)$; and the 4 intrinsic parameters defined by the projection of the optical centre in the image plane (u_0, v_0) and the perspective parameters $(\alpha_u, \alpha_{v)}$ (the reader is referred to chapter two for a detailed explanation). However, as a result of some types of imperfections in the design and assembly of the lens composing the optical system, a linear relation does not hold true. These kinds of imperfections, known as lens distortion, can be modelled by a radial and tangential approximation. Radial distortion causes an inward or outward displacement of a given image point from its ideal location, and it has been demonstrated to be quite important in camera modelling (see chapter 2 for a detailed explanation). Radial lens distortion can be approximated by non-linear equations governed by the first coefficient k_1. The relation between a 3D object point and its observable projective point is then modelled by the following equations 5.2 and 5.3 (the reader should compare them with equation 2.27 - 2.30, note that we do not have to take care of the sign of the parameters as they will be inferred in calibration).

$$U = X_u - X_d - k_1 r^2 X_d = 0 \quad (5.2)$$

$$V = Y_u - Y_d - k_1 r^2 Y_d = 0 \quad (5.3)$$

where,

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} \mathbf{R} & T \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{5.4}$$

and,

$$X_u = f\frac{X_c}{Z_c} \tag{5.5}$$

$$Y_u = f\frac{Y_c}{Z_c} \tag{5.6}$$

$$X_d = \frac{(X_i - u_0)}{k_u} \tag{5.7}$$

$$Y_d = \frac{(Y_i - v_0)}{k_v} \tag{5.8}$$

$$r^2 = X_d^2 + Y_d^2 \tag{5.9}$$

Let us consider equation 5.2 and 5.3 as a function $G = \{GU, GV\}$ which depends on 11 unknowns. These 11 unknowns are arranged in a vector called X, that is $X = \{u0, u1, ..., u10\}$, for instance, we have fixed X as vector $X = \{\alpha, \beta, \gamma, t_X, t_Y, t_Z, k_u, k_v, u_0, v_0, k_1\}$. We have not included the focal distance f as an unknown parameter, due to the following relationship,

$$a_u = f\ k_u \tag{5.10}$$

$$a_v = f\ k_v \tag{5.11}$$

Note that infinite combinations of (f, k_u, k_v) parameters can be obtained without changing the (a_u, a_v) values. That is why we have to fix one of the three parameters, otherwise it becomes unstable as the system has infinite solutions. Thus we have decided to fix the focal distance f as a constant.

The obtaining of the X vector is the main objective of the calibrating method. As a result of the inclusion of the lens distortion in the camera model, G has become a non-linear equation. That is why we have to use an iterative calibrating method to minimise G in order to obtain X.

We want to calculate the values of X which solve the equation $G(X) = 0$ iterating from an initial solution X_0. The initial solution is computed using the linear method of Toscani explained in chapter 2, assuming no lens distortion ($k_1 = 0$). In order to calculate the model including lens distortion we have used the widely known Newton-Raphson iterative method of function

minimisation, also explained in chapter 2. Then, from a determined iteration k, we can approximate the next values of X from the last ones, using equation 5.12. Note that the solution we want to find is vector X of the unknowns which solves 5.13.

$$G(X_k) \approx G(X_{k-1}) + J(X_{k-1})\,\Delta X_k \tag{5.12}$$
$$G(X_k) = 0 \tag{5.13}$$

Then equation 5.14 is derived, which may be solved from the least-squares method expressed in equation 5.15.

$$\Delta X_k = -J^{-1}(X_{k-1})G(X_{k-1}) \tag{5.14}$$
$$\Delta X_k = -\left(J^t(X_{k-1})J(X_{k-1})\right)^{-1} J^t(X_{k-1})G(X_{k-1}) \tag{5.15}$$

If we consider that n different correspondence matchings are obtained, then $G(X_{k-1})$ is an $2*n$-dimensional vector. $J(X_{k-1})$ is an $(2*n)x11$ matrix in which each column is a partial derivative of G with respect to each unknown, and each row and evaluation of these derivatives from each correspondence couple of points. ΔX_k is a 11-dimensional vector which contains the estimated error of the 11 unknown parameters to be determined. Following, each component of equation 5.15 is expressed in more detail.

$$G(X_{k-1}) = \begin{pmatrix} GU_0(X_{k-1}) \\ GV_0(X_{k-1}) \\ .. \\ GU_i(X_{k-1}) \\ GV_i(X_{k-1}) \\ .. \\ GU_n(X_{k-1}) \\ GV_n(X_{k-1}) \end{pmatrix} \tag{5.16}$$

$$J(X_{k-1}) = \begin{pmatrix} \frac{\partial GU_0(X_{k-1})}{\partial \alpha} & \frac{\partial GU_0(X_{k-1})}{\partial \beta} & \frac{\partial GU_0(X_{k-1})}{\partial \gamma} & \frac{\partial GU_0(X_{k-1})}{\partial t_X} & \cdots & \frac{\partial GU_0(X_{k-1})}{\partial k_1} \\ \frac{\partial GV_0(X_{k-1})}{\partial \alpha} & \frac{\partial GV_0(X_{k-1})}{\partial \beta} & \frac{\partial GV_0(X_{k-1})}{\partial \gamma} & & & \frac{\partial GV_0(X_{k-1})}{\partial k_1} \\ \cdots & & & & & \cdots \\ \frac{\partial GU_i(X_{k-1})}{\partial \alpha} & & \cdots & & \cdots & \frac{\partial GU_i(X_{k-1})}{\partial k_1} \\ \frac{\partial GV_i(X_{k-1})}{\partial \alpha} & & \cdots & & \cdots & \frac{\partial GV_i(X_{k-1})}{\partial k_1} \\ \cdots & & & & & \cdots \\ \frac{\partial GU_n(X_{k-1})}{\partial \alpha} & & \cdots & & \cdots & \frac{\partial GU_n(X_{k-1})}{\partial k_1} \\ \frac{\partial GV_n(X_{k-1})}{\partial \alpha} & & \cdots & & \cdots & \frac{\partial GV_n(X_{k-1})}{\partial k_1} \end{pmatrix} \tag{5.17}$$

Equation 5.15 is computed by the iterative algorithm. After each iteration, the error found in each parameter is added to its value obtaining the new value for the next iteration. We must iterate until equation 5.18 is reached.

$$\Delta X_k < \varepsilon \tag{5.18}$$

5.3.2 Calibrating the camera and the projector model

In order to calibrate the system we only have to put the camera and the projector in the scenario taking into account that the camera can snap properly the scene illuminated by the projector. When the camera and the projector have been correctly placed and their lens have been properly focused, the calibration process starts. Firstly, we have to fix a 3D world reference co-ordinate system, then, we have to place a 3D known surface in the scene, usually two orthogonal planes (the so-called calibrating pattern). The planes of the calibrating pattern have to be orthogonals in order to obtain components of the 3D object points with respect to the three axes. This principle is one of the most commonly used in order to calibrate the system properly. See also figure 5.4 to view the space relation between the camera, the projector and the calibrating pattern. In order to simplify the measurement of the 3D object points with respect to the 3D world co-ordinate system we have placed it in the lower corner of the calibrating pattern as shown in the following schema 5.5.

Camera Calibration

In order to calibrate the camera sensor we have to fix a set of 3D object points on the surface defined by the calibrating pattern. We have decided to use a pattern made by equidistants black squares, which has been used by several authors, for instance [Faug 93]. Each square has a size of 30 mm. x 30 mm. and the distance between squares is 20 mm. in both directions. Each square defines four vertexes, that is four 3D object points. As our pattern is made by 2 planes with 20 squares for plane, we obtained a total of 160 object points, which is enough to calibrate the system. In order to simplify the measurement of the 3D object points we have moved the world co-ordinate system as shown in the following schema 5.6, showing also the position of the squares in the pattern. Figure 5.7 presents the real calibrating pattern used to calibrate the camera.

Once the 3D calibrating pattern has been fixed in the scene, we have to grab its image. The position of each square vertex in the 2D image has been hand-segmented in order to obtain its position accurately in the image plane.

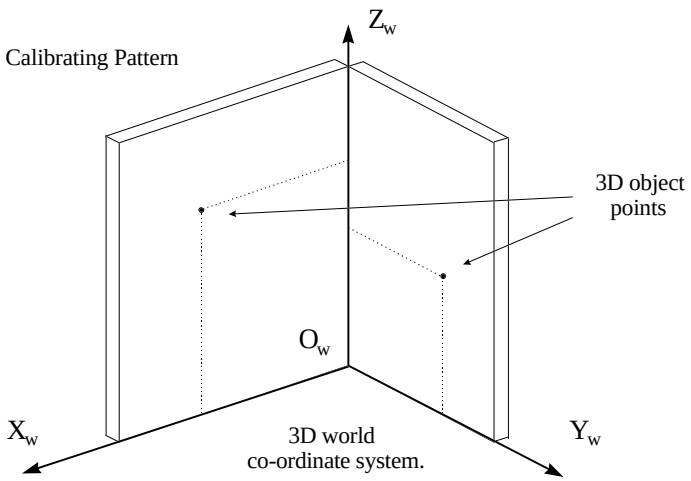

Figure 5.5: Schema of the calibrating pattern and its position and orientation with respect to the world co-ordinate system.

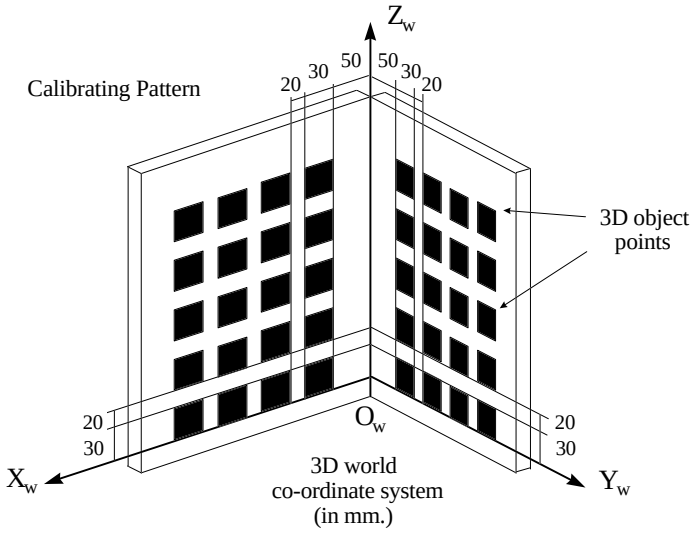

Figure 5.6: Schema of the pattern used in order to calibrate the camera.

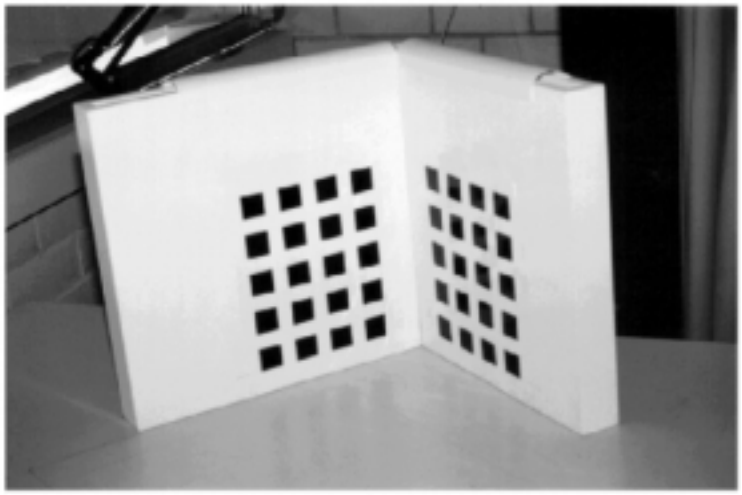

Figure 5.7: The calibrating pattern of squares.

Then, the association of the 2D object points and the 3D object points is obtained directly, defining a matrix **P** of 160 rows and 5 columns (the three co-ordinates of the 3D object point and the two co-ordinates of its imaged point). Figure 5.8 shows the image of the calibrating pattern captured by the camera sensor.

Figure 5.9 shows the position of the square vertexes detected in the image plane (image points), and figure 5.10 shows the position of the square vertexes measured with respect to the world co-ordinate system (object points).

The **P** matrix is the input of the calibrating algorithm defined in the previous section. We have computed intrinsic and extrinsic parameters using the method proposed by Toscani, explained in the linear calibration section of chapter 2. However, the maximum discrepancy between the modelled projections and the observable points is rather large as it is around 2.8 pixels. The main problem is due to the fact that Toscani does not take into account lens deformation and that it is a non-iterative method. Then an iterative algorithm is quite interesting to decrease this discrepancy, readjusting the intrinsic and extrinsic parameters of the system. We used the iterative method without modelling the lens deformation and obtained a maximum error around 1.2 pixels, which improves considerably Toscani's method. Then, the lens deformation was modelled. The iterative algorithm starts from the parameters obtained by Toscani and a null radial distortion. The initial solution was compared with the final parameters. The maximum discrepancy

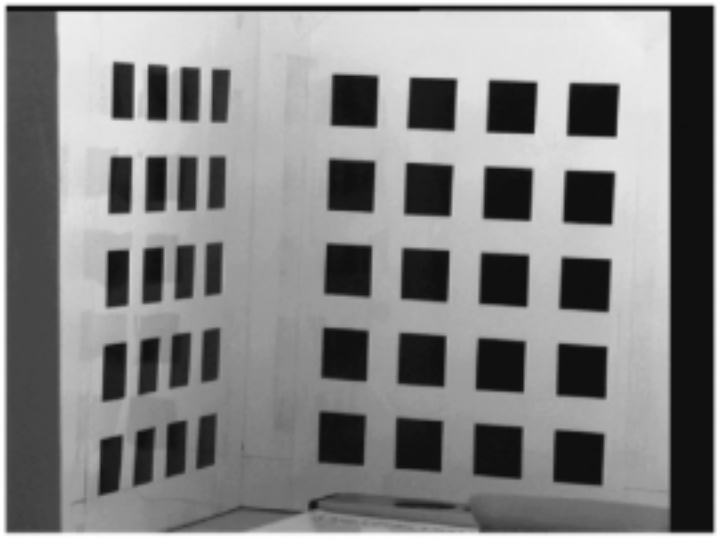

Figure 5.8: The image of the calibrating pattern grabbed by the camera.

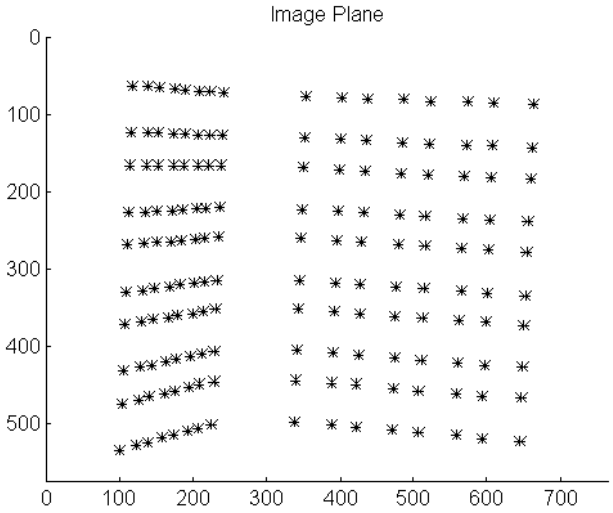

Figure 5.9: Position of the 2D image points of the camera.

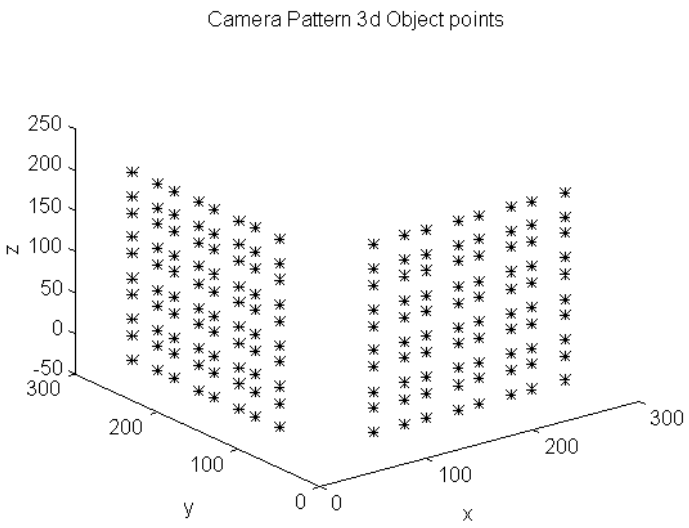

Figure 5.10: Position of the 3D object points of the camera.

between the modelled projections and the observable points with the obtained parameters was around 0.0053 pixels. The experiments proved that once the number of iterations exceeds 20, the accuracy of the system does not improve substantially. The following table 5.1 shows the initial solution and the resulting intrinsic and extrinsic parameters obtained by the iterative algorithm. However, in table 5.1 we have shown only the most significative decimals, in the model we have used until the 10^{th} decimal.

Projector Calibration

The calibration of the pattern light projector is similar to the calibration of the camera sensor. However, the obtaining of the matrix $\mathbf{P}$ differs a little. Of course, the measurement of the 2D image point is direct as we know how the coloured grid pattern has been constructed. The pattern is projected on the calibrating surface. We have placed two sheets of paper with a milimetrical grid on the calibrating surface (see figure 5.11).

This metric grid allows us to obtain the 3D position of the object points. We have used as object points all the crossing points of the coloured grid. In order to obtain the crossing points we have measured both 3D ending point positions of each slit, obtaining the equation of the slit in the space. Then, analysing each slit with the others, we obtain all the crossing points of each slit, and with this, the set of 3D object points. Associating each 2D object

Table 5.1: Intrinsic and extrinsic parameters of the camera sensor.

	X_0	X_1	X_5	X_{10}	X_{15}	X_{20}
α	1.6605	1.7292	1.7294	1.7294	1.7294	1.7294
β	-1.2051	-1.2537	-1.2537	-1.2537	-1.2537	-1.2537
γ	-3.0980	-3.0326	-3.0325	-3.0325	-3.0325	-3.0325
t_X	19.7	-84.5	-84.9	-84.9	-84.9	-84.9
t_Y	144.9	106.4	106.3	106.3	106.3	106.3
t_Z	1829.4	2414.9	2430.6	2432.1	2432.1	2432.1
f	10	10	10	10	10	10
k_u	316.05	418.07	420.81	421.06	421.07	421.07
k_v	332.57	439.28	442.16	442.42	442.43	442.43
u_0	223.43	404.47	405.13	405.19	405.19	405.19
v_0	227.25	297.56	297.78	297.80	297.80	297.80
k_1	0	-0.03399	-0.03496	-0.03505	-0.03505	-0.03505

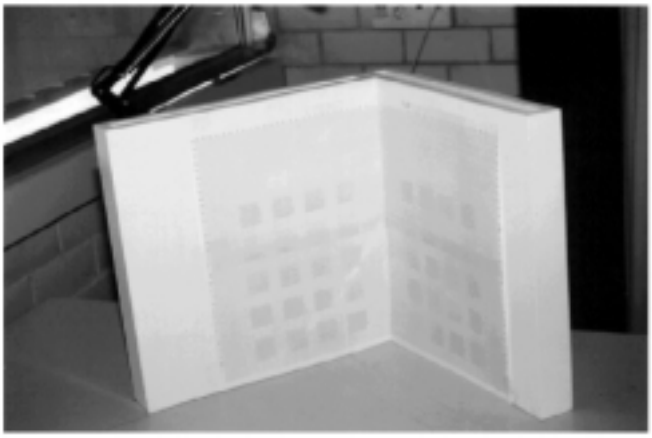

Figure 5.11: The metric grid used in projector calibration.

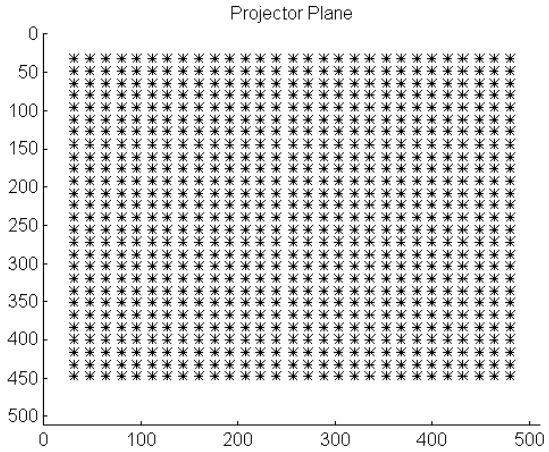

Figure 5.12: Position of the 2D crossing points of the projected pattern.

point with its 3D object point, we have obtained a matrix **P** of 783 rows and 5 columns. The following figures 5.12 and 5.13 show the 2D image points and the 3D objects points respectively. Note that we did not use the two last horizontal slits because they have been projected out of the metric grid, thus, the 783 crossing points have been obtained by crossing 29 vertical slits by 27 horitzontal slits.

Taking into account the same considerations already described in the previous section on camera calibration and using the same iterative algorithm of non-linear calibration with the 783x5 **P** matrix, we have obtained the set of intrinsic and extrinsic parameters which model the projector system. The following table shows the initial solution obtained also by the linear method proposed by Toscani, and the parameters obtained iterating the non-linear model. We have also used 20 iterations in order to minimise the function.

5.4 Accuracy of the method

In order to compare 3D geometric information inferred by the proposed coded structured light system and the real 3D information of the scene, we have projected the coloured pattern on the calibrating plane. The calibrating plane has a dimensions of 230 mm. with respect to each axe X_w, Y_w and Z_w. Several object points have been placed on the calibrating plane surface. We have measured the 3D real point, and we have inferred the 3D object point from both projective 2D points. The (X, Y, Z) error measured is

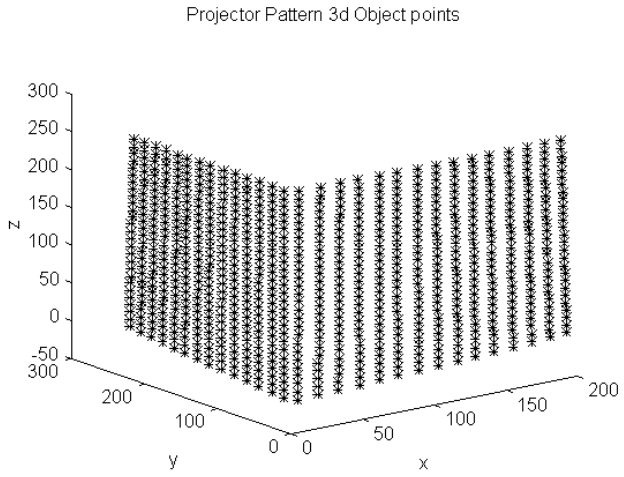

Figure 5.13: Position of the 3D crossing points of the projected pattern.

Table 5.2: Intrinsic and extrinsic parameters of the pattern projector.

	X_0	X_1	X_5	X_{10}	X_{15}	X_{20}
α	1.8091	1.7001	1.7507	1.7511	1.7511	1.7511
β	-0.7090	-0.6994	-0.7041	-0.7042	-0.7042	-0.7042
γ	-2.9961	-3.0648	-3.0326	-3.0324	-3.0324	-3.0324
t_X	-79.8	-42.7	-61.6	-61.8	-61.8	-61.8
t_Y	-82.7	49.9	-13.7	-14.3	-14.3	-14.3
t_Z	1772.1	1892.3	1911.4	1910.8	1910.8	1910.8
f	10	10	10	10	10	10
k_u	251.93	269.80	272.88	272.80	272.80	272.80
k_v	261.31	279.77	282.74	282.66	282.66	282.66
u_0	376.22	323.72	350.45	350.82	350.82	350.82
v_0	560.45	365.65	459.30	460.09	460.09	460.09
k_1	0	-0.00308	-0.00312	-0.00309	-0.00309	-0.00309

Table 5.3: A sample of the analysed object points.

u_c	v_c	u_p	v_p	x_r	y_r	z_r	x_m	y_m	z_m
210	69	144	96	102	0	229	101.99	1.21	228.39
530	344	416	320	0	159	92	0.78	159.62	91.31
430	78	352	96	0	93	230	1.15	94.43	229.84
202	466	160	416	90	0	22	89.02	−0.09	21.72

the discrepancy between real co-ordinates and the inferred ones. We have measured the error in approximately 100 points. Table 5.3 shows a sample of 4 object points with their 2D projections on the camera image plane (u_c, v_c) and in the pattern frame (u_p, v_p), their real 3D co-ordinates (x_r, y_r, z_r) and the 3D co-ordinates obtained by our method (x_m, y_m, z_m).

The average values of the error deviate 0.728 mm.(0.31 %) from the X_w axe, 0.624 mm. (0.27 %) from the Y_w axe, and 0.465 mm.(0.20 %) from the Z_w axe. However, the maximum error deviate 1.15 mm. (0.5 %) from the X_w axe, 1.43 mm. (0.62 %) from the Y_w axe and 0.68 mm. (0.29 %) from the Z_w axe. The results are quite interesting if we take into account that the deviation degree is highly influenced by the segmentation process, i.e by the image noise and the problem of grabbing the projected lines of the pattern with different thickness due to different depth projections.

5.5 Measuring three-dimensional objects

Once the camera and the projector have been calibrated, the system is allowed to reconstruct 3D information of an object point from its projective 2D points. The algorithm of reconstruction from non-linear calibration explained in section 2.4.2 has been used. Obviously, if the position and orientation of the camera or the projector in the space is modified, then, the system has to be re-calibrated, as the intrinsic and extrinsic parameters have changed.

The RGB camera snaps an image of the pattern projected on the scene that we want to measure. A RGB-PAL 768 x 576 x 24 bit image is grabbed. The image is captured by a Matrox Comet acquisition card into the memory of a PC Pentium. At this point straight lines are explored in order to detect cross-points. Cross-points are the tokens used as image points to obtain 3D information of the captured scene. The cross-points have been hand-segmented. Using the coloured codification of the slits which form the cross-point in the image plane, the matching, i.e. the determination of the

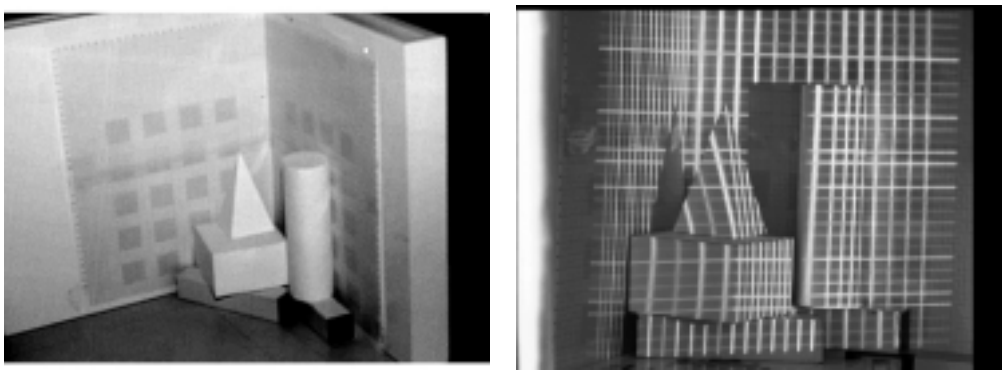

Figure 5.14: A scene composed by three geometric objects. a) The real scene (not used to infer 3D information); b) The scene illuminated by the coloured pattern.

position of the same point in the projector image plane, is carried out. Then, 3D object point co-ordinates are obtained from both 2D positions of their projective points in the image plane and in the projected pattern, respectively.

In the following, two examples of 3D measurement of static objects is shown and discussed. As examples we have chosen the measurement of a scene made by three geometrical objects, and the measurement of a face mask. Furthermore, two examples of dynamic scene measurement are shown. Both examples are based on the measurement of a human hand. In the first example, range information of the palm of the hand is obtained. The second example illustrates the obtaining of the range information from the back of the same hand after a movement.

5.5.1 Example : Measuring geometrical objects

The first example is composed by a scene with three geometric and achromatic objects illuminated by the coloured pattern, as shown in figure 5.13. Figure 5.14 shows the measured correspondence points in the projector and camera frame, and figure 5.15 shows the 3D reconstruction.

5.5.2 Example : Measuring a face mask

The second example is based on the 3D reconstruction of a face mask, see figure 5.16. Figure 5.17 shows the measured correspondence points in both

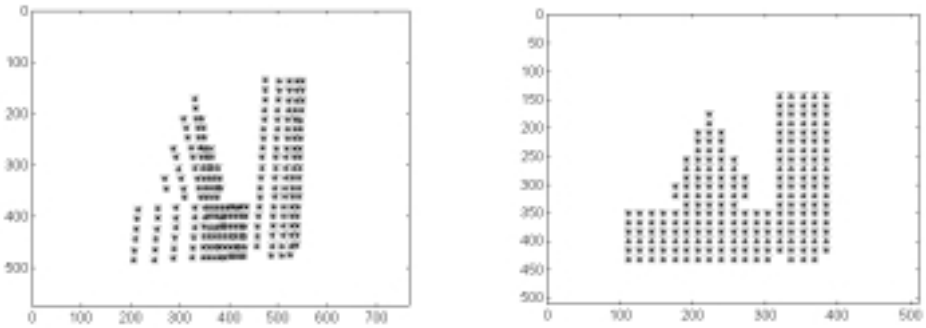

Figure 5.15: The matching obtained. a) The Matching points from the 576 x 768 camera image plane, b) The Matching points from the 512 x 512 projected pattern frame.

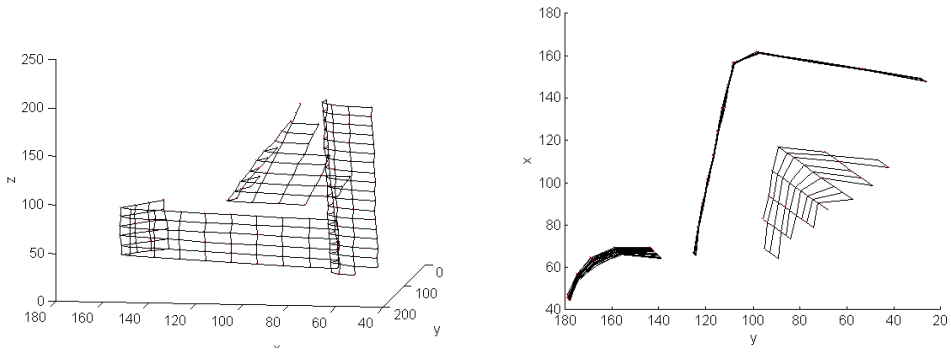

Figure 5.16: The reconstruction of the 3D objects. a) A perspective view from the right side. b) A top view.

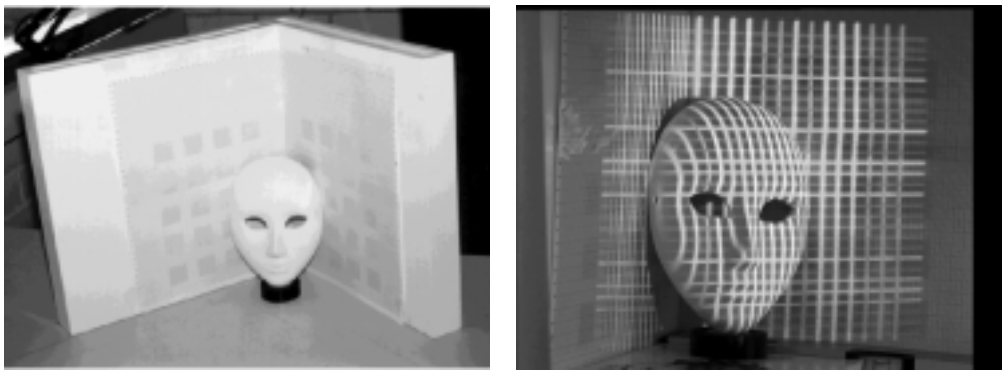

Figure 5.17: A scene composed by a face mask. a) The real scene (not used to infer 3D information); b) The scene illuminated by the coloured pattern.

frames and figure 5.18 the 3D reconstruction.

5.5.3 Example : Measuring the palm of a hand

The third example is based on the 3D reconstruction of a human palm, see figure 5.19. Figure 5.20 shows the measured correspondence points in both frames and figure 5.21 the 3D reconstruction. In this case, we have reduced the distances among slits in order to increase the resolution of the system, then, we have projected the same coloured grid pattern made in a 256 x 256 image.

5.5.4 Example : Measuring the back of a hand

The fourth example is based on the 3D reconstruction of a human back hand, see figure 5.22. Figure 5.23 shows the measured correspondence points in both frames and figure 5.24 the 3D reconstruction. In this case, we have also reduced the distances among slits in order to increase the resolution of the system, then, we have projected the same coloured grid pattern made in a 256 x 256 image.

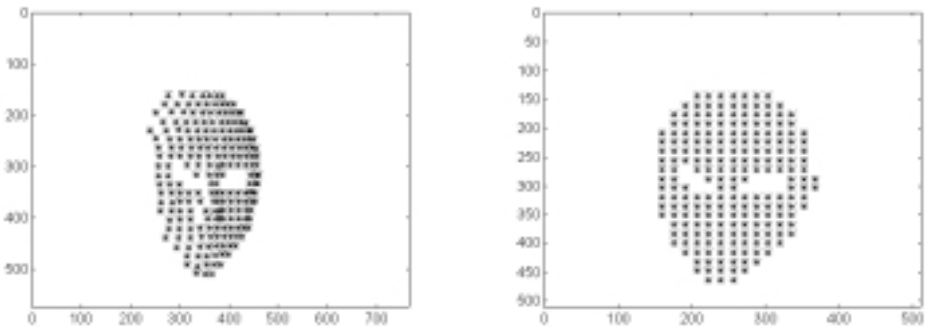

Figure 5.18: The matching obtained. a) The Matching points from the 576 x 768 camera image plane, b) The Matching points from the 512 x 512 projected pattern frame.

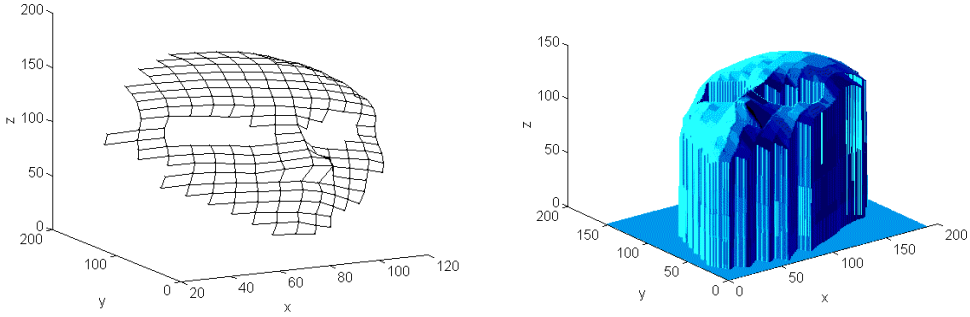

Figure 5.19: The reconstruction of the face mask.

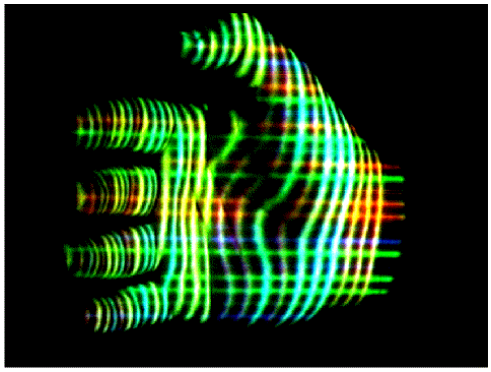

Figure 5.20: An illuminated human palm.

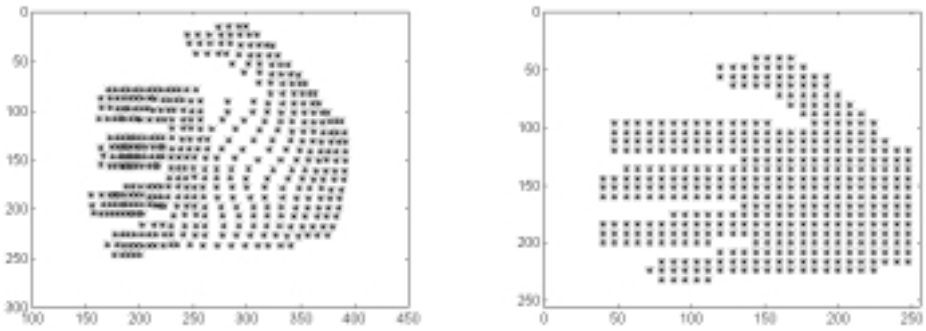

Figure 5.21: The matching obtained. a) The matching points from the 576x768 camera image plane. b) The matching points from the 256x256 projector pattern frame.

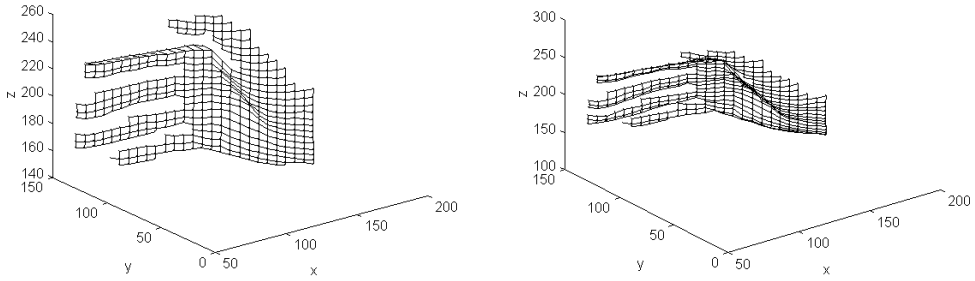

Figure 5.22: Reconstruction of the human palm.

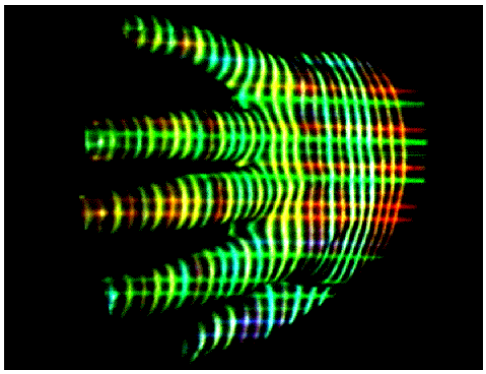

Figure 5.23: An illuminated back hand.

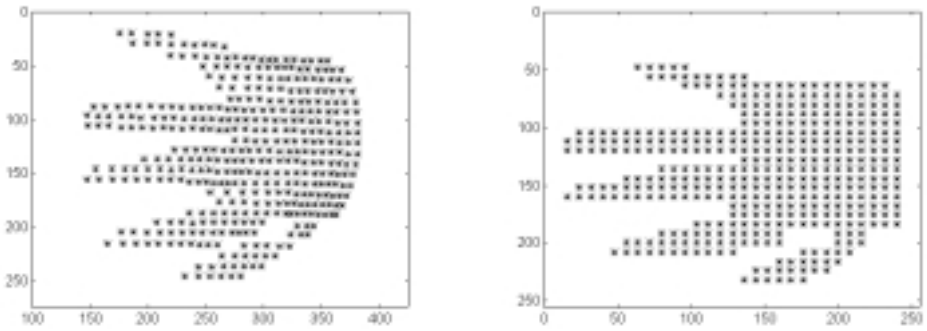

Figure 5.24: The matching obtained. a) The matching points from the 576x768 camera image plane. b) The matching points from the 256x256 projector pattern frame.

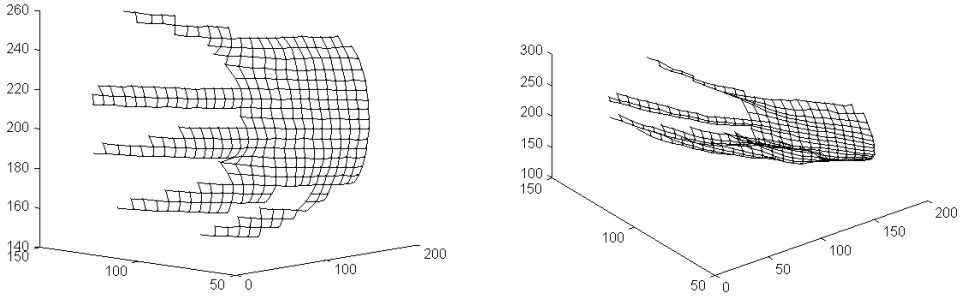

Figure 5.25: Reconstruction of the human back hand.

Chapter 6

Conclusions

This chapter presents the conclusions of this work. Further work is also analysed considering some ways through which the research can continue. Finally, the chapter lists the articles directly related with the thesis, and other research contributions that have been published.

6.1 Conclusions

As conclusions of the work done in this thesis, it can be affirmed that:

The problem of camera calibration has been described in detail, analysing the contribution of each intrinsic and extrinsic parameter in the whole camera model. Two models have been discussed, the linear model without considering the lens distortion, and the non-linear model which takes into account the lens distortion. The process used to calibrate the camera has also been analysed. With respect hereto, we have described the Toscani method of linear camera calibration, and the iterative method of Newton-Raphson which has been used in non-linear camera calibration.

The problem involved with stereo vision systems of solving the matching between two correspondence points has also been described. We have analysed the correspondence problem and one of the most powerful tools that are used in the scientific community to reduce it. In fact, the epipolar geometry has been described in detail, and we have presented the equations which permit to reduce the two-dimensional search of the matching point to a one-dimensional one.

It has been demonstrated that an easy way to solve the correspondence problem, is substituting a camera by a pattern projector which is known as a structured light system. Structured light can solve the problem of points without match, but it can not solve the problem of multiple matching. Then,

we have presented coded structured light which has been used in order to solve completely the correspondence problem. A survey of the most interesting techniques which are based on coded pattern projection has been presented and each technique has been analysed and discussed in detail.

Furthermore, we have presented a new coded structured light pattern, which can be used in order to solve the correspondence problem in 3D surface measurement. Most of the techniques presented in the survey are based on single axe codification. The proposed pattern is coded with respect to both axes, which has been demonstrated to obtain more accurate results. 3D information is obtained by a single pattern shot. This principle allows the system to be used in the measurement of moving objects, which is another big challenge in coded structured light. The pattern design has been explained. The system is made by grid pattern projection based on coloured coded slits. The principle of codification is based on the mathematical study proposed by Griffin et al. [Grif 92]. However, we are only interested in the maximal sequence of triplets that can be obtained from a determined number of colour primitives. Vertical and horizontal slits are coloured using three different colours, obtaining a unique-encoded grid pattern of 29 x 29 orthogonal slits. Then, a unique match is obtained between the grabbed pattern and the projected one, solving the correspondence problem.

The whole system made by an imaging sensor and the pattern projector has been calibrated using a non-linear calibration algorithm, taking into account lens distortion. Finally, four examples of 3D object measurement are shown. We have tested the system measuring two static scenes: one scene consisting of geometric objects and another one consisting of a face mask. Furthermore, we have tested the system measuring two dynamic scenes consisting of a human hand.

The projection of coloured patterns may not be used in the measurement of coloured surfaces. However, using our pattern, the measurement of low saturated or pale surfaces is allowed. As we project a reduced number of well-defined colours, small colour discrepancies between the projected colour and the captured one are permitted.

The proposed measuring system can be used in several applications of 3D object measurement. For instance, it can be used in the measurement of objects which move due to the displacement of a conveyor as an industrial application. Although the same problem can also be solved using a single laser slit projection, our system can measure the object without scanning, which may reduce the computing time.

The pattern can also be used as a system which takes range information of the surrounded scene of a mobile robot. In such a way, the resolution of the pattern may vary dynamically in order to grab always the best image,

which can be done re-programming the digital projector and, what is more important, without re-calibrating the system. Of course, we have to adjust the number of slits as a result of the complexity of the scene and the permitted computing time. However, one of the big challenges is to self-calibrate the measuring system in order to grab always a focused image of the scene. This is a hard problem which involves one of the most interesting topics in computer vision and it has been proposed as a further research in the following section.

6.2 Further Work

The work presented in this thesis has two main subjects of further research:

Firstly, we have not studied the behaviour of the coloured grid pattern when it is projected on a scene made by coloured objects. Obviously, a frequency shifting is suffered by the colour captured by the camera with respect to the colour projected by the pattern due to the intrinsic colour of the measuring objects. If the pattern colours are well chosen, this discrepancy may allow us to obtain the intrinsic colour of the scene objects. Then, the proposed system could be used to obtain 3D information and coloured information of any scene by a single pattern shot projection.

Secondly, we are also interested in the possibility of using coded structured light techniques to measure a wide range of depth distances, which is essential in order to use the proposed pattern in mobile robot navigation. Such large range of depths forces us to self-calibrate the system dynamically, that is, without human interaction, which means that the co-ordinates of the 3D object points are unknown in calibration. In this case, we are thinking of using a stereo vision system based on two cameras and a pattern projector in order to obtain 3D information without a calibration of the projector but a self-calibration of both cameras. It is known that a pattern projector can not be self-calibrated as it does not fulfil the principle of static 3D objects points while the system is calibrated. Then, the projecting pattern can be used as a simple tool which permits to solve the matching, along the epipolar lines of both cameras, by colour comparing between both imaged scenes.

6.3 Publications related with the thesis

The publications obtained from the work of this thesis are the following:

1. J. Salvi and J. Batlle. Contribution of the Mobile Navigation in Structured Indoor Environments using Laser Beam Patterns. Int. Symp. on

Lasers, Optics, and Vision for Productivity in Manufacturing I. SPIE - The International Society for Optical Engineering, **EUROPTO '96**, Vol. 2785, Besançon (France), June 1996.

2. E. Mouaddib, J. Salvi and J. Batlle. A Survey: Coded Structured Light. Research Report No. LSA / CSL / 96, Laboratoire des Systemes Automatiques, Universite de Picardie - Jules Verne, Amiens (France), July 1996.

3. E. Mouaddib, J. Batlle and J. Salvi. Recent Progress in Structured Light in order to Solve the Correspondence Problem in StereoVision. IEEE Int. Conf. on Robotics and Automation, **ICRA '97**, Vol. I, pp 130-136. Albuquerque (USA), April 1997.

4. J. Salvi, E. Mouaddib and J. Batlle. An Overview on Self-Calibration: Application to Coded Pattern Projection. Research Report No. LSA / OCC / 97, Laboratoire des Systemes Automatiques, Universite de Picardie - Jules Verne, Amiens (France), April 1997.

5. J. Salvi, E. Mouaddib and J. Batlle. An Overview of the Advantages and Constraints of Coded Pattern Projection Techniques for Autonomous Navigation. IEEE Int. Conf. on Intelligent Robots and Systems, **IROS '97**, Vol. III, pp 1264-1271. Grenoble (France), September 1997.

6. J. Batlle, E. Mouaddib and J. Salvi. A Survey: Recent Progress in Coded Structured Light as a Technique to Solve the Correspondence Problem. Int. Journal of Pattern Recognition, **PR**. Pergamon Press, Oxford. Accepted to be published.

7. J. Batlle, J. Salvi and E. Mouaddib. Unique-Encoded Coloured Grid Pattern Projection for Dynamic Measurement of Moving Scenes. IEEE Int. Conf. on Robotics and Automation, **ICRA '98**. Leuven (Belgium), May 1998. Submitted.

8. J. Salvi, J. Batlle and E. Mouaddib. A Robust-Coded Pattern Projection for Dynamic 3D Scene Measurement. Int. Journal of Pattern Recognition Letters, **PRL**. Elsevier Science, The Netherlands. Sent in September 1997. Submitted.

6.4 Other Research Contributions

Following, a list of other research contributions of the author is related.

1. J. Salvi, Ll. Pacheco and R. Garcia-Campos. A Mobile Robot for Research Experimentation. In Proc. World Automation Congress, Robotics and Manufacturing Vol. 6, pp 745-750. Montpellier (France), May 1996.

2. J. Forest, R. Garcia-Campos and J. Salvi. Position, Orientation Detection and Control of Mobile Robots through Real Time Processing. In Proc. World Automation Congress, Robotic and Manufacturing Systems Vol. 3, pp 307-312. Monpellier (France), May 1996.

3. J. Salvi. Co-Chair of the ThR8 - 'Experimental Robot Design' Session. ISRAM Sixth International Symposium on Robotics and Manufacturing. World Automation Congress. Montpellier (France), May 1996.

4. R. García-Campos, J. Salvi, L.Pacheco" Una Plataforma para la Experimentación", SAAEI - Seminario Anual de Automática y Electrónica Industrial Vol. I, pp 467-470. Zaragoza (Spain), September 1996.

5. J. Salvi, P. Ridao and J. Batlle. Accurate Measurement of Mechanical Deviation using Laser Beams and Computer Vision. 30th. Int. Symp. on Automotive Technology and Automation. Vol. Laser Applications in the Automative Industries, pp 415-422. Florence (Italy), June 1997.

6. J. Batlle, P. Ridao and J. Salvi. Integration of a Teleoperated Robotic Arm with Vision Systems using CORBA Compatible Software. 30th. Int. Symp. on Automotive Technology and Automation. Vol. Robotics, Motion and Machine Vision in the Automative Industries, pp 371-378. Florence (Italy), June 1997.

Bibliography

[Agin 73] G.J. Agin and T.O. Binford. Computer Description of Curved Objects. In Proceedings International Joint Conference on Artificial Intelligence, pp 629-640, 1973.

[Ahle 89] R. Ahlers and J. Lu. Stereoscopic Vision - An Application Oriented Overview. In Proceedings SPIE, Vol. 1194, pp 298-307, 1989.

[Alts 81] M.D. Altschuler, B.R. Altschuler and J. Taboada. Laser Electro-Optic System for Rapid Three-Dimensional (3-D) Topographic Mapping of Surfaces. International Journal on Optical Engineering, Vol. 20, No. 6, pp 953-961, 1981.

[Alts 87] M.D. Altschuler, B.R. Altschuler, J.T. Dijak, L.A. Tamburino and B. Woolford. Robot Vision by Encoded Light Beams. In Takeo Kanade Three-Dimensional Machine Vision. Kluwer Academic Publishers, 1987.

[Alve 89] N. Alvertos, D. Brzakovic and R.C. Gonzalez. Camera Geometries for Image Matching in 3-D Machine Vision. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, No. 9, pp 897-915, 1989.

[Asad 85] M. Asada and S. Tsuji. Utilization of a Stripe Pattern for Dynamic Scene Analysis. In Proceedings International Joint Conference on Artificial Intelligence, pp 895-897, 1985.

[Asad 86] M. Asada, H. Ichikawa and S. Tsuji. Determining of Surface Properties by Projecting a Stripe Pattern. In Proceedings International Conference on Pattern Recognition, pp 1162-1164, 1986.

[Asad 88] M. Asada, H. Ichikawa and S. Tsuji. Determining Surface Orientation by Projecting a Stripe Pattern. IEEE Transactions on

Pattern Analysis and Machine Intelligence, Vol. 10, No. 5, pp 749-754, 1988.

[Ayac 91] N. Ayache and F. Lustman. Trinocular Stereo Vision for Robotics. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 13, No. 1, pp 73-84, 1991.

[Ayac 91b] N. Ayache. Artificial Vision for Mobile Robots: Stereo Vision and Multisensory Perception. The MIT Press, 1991.

[Barn 82] S.T. Barnard and M.A. Fischler. Computational Stereo. International Journal on Computing Surveys, Vol. 14, No. 4, pp 553-572, 1982.

[Besl 88] P.J. Besl. Active, Optical Range Imaging Sensors. International Journal on Machine Vision and Applications, Vol. 1, pp 127-152, 1988.

[Beye 92] H.A. Beyer. Accurate Calibration of CCD-Cameras. International Conference on Computer Vision and Pattern Recognition, pp 96-101, 1992.

[Bort 91] F. Bortolozzi. Vision Trinoculaire, Une Solution Geometrique. Thèse Université de Technologie de Compiègne. 1991.

[Boye 87] K.L. Boyer and A.C. Kak. Color-Encoded Structured Light for Rapid Active Ranging. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 9, No. 1, pp 14-28, 1987.

[Brad 88] J.P. Brady, N. Nandhakumar and J.K. Aggarwal. Recent Progress in the Recognition of Objects from Range Data. In Proceedings International Conference on Pattern Recognition, Vol. I, pp 85-92, 1988.

[Bras 90] G. Brassard and P. Bratley. Algorítmica: Concepción y Análisis. Ed. Masson, 1990.

[Bras 95] E. Brassart. Localisation Absolue d'un Robot Mobile Autonome par des Balises Actives et un Systeme de Vision Monoculaire. Thèse Université de Technologie de Compiègne. 1995.

[Carr 85] B. Carrihill and R. Hummel. Experiments with the Intensity Ratio Depth Sensor. International Journal on Computer Vision Graphics and Image Processing, Vol 32, pp 337-358, 1985.

[Chen 97] C.S. Chen, Y.P. Hung, C.C. Chiang and J.L. Wu. Range Data Acquisition Using Color Structured Lighting and Stereo Vision. International Journal on Image and Vision Computing, Vol. 15, pp 445-456, 1997.

[Davi 96] C.J. Davies and M.S. Nixon. Sensing Surface Discontinuities via Coloured Spots. In Proceedings Third International Workshop on Image and Signal Processing, pp 573-576, 1996.

[Faug 86] O.D. Faugeras and G. Toscani. The Calibration Problem for Stereo. In Proceedings International Conference on Computer Vision and Pattern Recognition, pp 15-20, 1986.

[Faug 93] O. Faugeras. Three-Dimensional Computer Vision: A Geometric Viewpoint. The MIT Press. 1993.

[Geng 96] Z.J. Geng. Rainbow Three-Dimensional Camera: New Concept of High-Speed Three-Dimensional Vision Systems. International Journal on Optical Engineering, Vol. 35, No. 2, pp 376-383, 1996.

[Gosh 89] A. Goshtasby. Correction of Image Deformation from Lens Distortion Using Bezier Patches. International Journal on Computer Vision Graphics and Image Processing, Vol. 47, pp 385-394, 1989.

[Grif 92] P.M. Griffin, L.S. Narasimhan and S.R. Yee. Generation of Uniquely Encoded Light Patterns for Range Data Acquisition. International Journal on Pattern Recognition, Vol. 25, No. 6, pp 609-616,1992.

[Hall 82] E.L. Hall, J.B.K. Tio, C.A. McPherson, C.S. Draper and F.A. Sadjadi. Measuring Curved Surfaces for Robot Vision. International Journal on Computer, Vol. Decenber, pp 42-54, 1982.

[Hara 93] R.M. Haralick and L.G. Shapiro. Computer and Robot Vision, Vol II. Addison-Wesley Publishing Company. 1993.

[Hatt 95] K. Hattori and Y. Sato. Handy Rangefinder for Active Robot Vision. In Proceedings International Conference on Robotics and Automation, pp 1423-1428, 1995.

[Hosh 90] J. Hoshino, T. Uemura and I. Masuda. Region-Based Reconstruction of an Indoor Scene Using and Integration of Active and Passive Sensing Techniques. In Proceedings International Conference on Computer Vision, pp 568-572, 1990.

[Hu 86] G. Hu, A.K. Jain and G. Stockman. Shape from Light Stripe Texture. In Proceedings International Conference on Computer Vision and Pattern Recognition, pp 412-414, 1986.

[Hu 89] G. Hu and G. Stockman. 3-D Surface Solution Using Structured Light and Constraint Propagation. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, No. 4, pp 390-402, 1989.

[Hu 89b] G. Hu and G. Stockman. Representation and Segmentation of a Cluttered Scene Using Fused Edge and Surface Data. In Proceedings International Conference on Computer Vision and Pattern Recognition, pp 313-318, 1989.

[Hung 93] D.C.D. Hung. 3D Scene Modelling by Sinusoid Encoded Illumination. International Journal on Image and Vision Computing, Vol. 11, No. 5, pp 251-256, 1993.

[Ikeu 91] K. Ikeuchi and K. Sato. Determining Reflectance Properties of an Object Using Range and Brightness Images. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 13, No. 1, pp 1139-1153, 1991.

[Inok 72] S. Inokuchi, Y. Morita and Y. Sakurai. Optical Pattern Processing Utilitzing Nematic Liquid Crystals. International Journal on Applied Optics, Vol. 11, No. 10, pp 2223-2227, 1972.

[Inok 84] S. Inokuchi, K. Sato and F. Matsuda. Range-Imaging System for 3-D Object Recognition. In Proceedings International Conference on Pattern Recogniton, pp 806-808, 1984.

[Ito 91] M. Ito. Robot Vision Modelling - Camera Modelling and Camera Calibration. International Journal on Advanced Robotics, Vol. 5, No. 3, pp 321-335, 1991.

[Ito 95] M. Ito and A. Ishii. A Three-Level Checkerboard Pattern (TCP) Projection Method for Curved Surface Measurement. International Journal on Pattern Recognition, Vol. 28, No. 1, pp 27-40, 1995.

[Jarv 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, pp 122-139, 1983.

[Jarv 93] R.A. Jarvis. Range Sensing for Computer Vision. In A.K. Jain and P.J. Flynn Three-Dimensional Object Recognition Systems, Elsevier Science Publishers B.V. 1993.

[Kana 91] K. Kanatani. Computational Projective Geometry. International Journal on Computer Vision Graphics and Image Processing Vol.54, No.3, pp 333-348, 1991.

[Kemm 94] K. Kemmotsu and T. Kanade. Sensor Placement Design for Object Pose Determination with Three Light-Stripe Range Finders. Technical Report No. CMU-CS-94-152, School of Computer Science, Carnegie Mellon University, 1994.

[Kemm 95] K. Kemmotsu and T. Kanade. Uncertainty in Object Pose Determination with Three Light-Stripe Range Measurements. IEEE International Journal of Robotics and Automation, Vol 11, No. 5, pp 741-747, 1995.

[Lavo 96] P. Lavoie, D. Ionescu and E. Petriu. 3D Reconstruction Using an Uncalibrated Stereo Pair of Encoded Images. In Proceedings International Conference on Image Processing 1996, Vol. 2, pp 859-862.

[Lenz 88] R.K. Lenz and R.Y. Tsai. Techniques for Calibration of the Scale Factor and Image Center for High Accuracy 3-D Machine Vision Metrology. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 10, No. 5, pp 713-720, 1988.

[Luon 92] Q-T. Luong. Matrice fondamentale et calibration visuelle sur l'environnement vers une plus grande autonomie des systèmes robotiques. Thèse Université Paris Sud. 1992.

[Luon 93b] Q. Luong, R. Deriche, O. Faugeras and T. Papadopoulo. On Determining the Fundamental Matrix: Analysis of Different Methods and Experimental Results. INRIA Research Report No. 1894, 1993.

[Maru 93] M. Maruyama and S. Abe. Range Sensing by Projecting Multiple Slits with Random Cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 15, No. 6, pp 647-651, 1993.

[Mino 81] M. Minou, T. Kanade and T. Sakai. A Method of Time-Coded Parallel Planes of Light for Depth Measurement. The Transactions of the IECE of Japan, Vol E64, No. 8, pp 521-528, 1981.

[Mohr 96] R. Mohr and B. Triggs. Projective Geometry for Image Analysis. A Tutorial at ISPRS. INRIA Rhône-Alpes. 1996.

[Moig 84] J. Le Moigne and A.M. Waxman. Projected Light Grids for Short Range Navigation of Autonomous Robots. In Proceedings International Conference on Pattern Recognition, Vol 1, pp 203-206, 1984.

[Moig 85] J. Le Moigne and A.M. Waxman. Multi-Resolution Grid Patterns for Building Range Maps. In Proceedings Vision, Vol. 8, pp 22-39, 1985.

[Moig 88] J. Le Moigne and A.M. Waxman. Structured Light Patterns for Robot Mobility. IEEE International Journal of Robotics and Automation, Vol. 4, No. 5, pp 541-548, 1988.

[Monk 92] T.P. Monks, J.N. Carter and C.H. Shadle. A real-time 3D digitisation system for speech research. In Proceedings SPIE, Vol. 1660, pp 819-829, 1992.

[Monk 93] T.P. Monks and J.N. Carter. Improved Stripe Matching for Colour Encoded Structured Light. In Proceedings International Conference on Computer Analysis of Images and Patterns, pp 476-485, 1993.

[Mori 88] H. Morita, K. Yajima and S. Sakata. Reconstruction of Surfaces of 3-D Objects by M-Array Pattern Projection Method. In Proceedigns International Conference on Computer Vision, pp 468-473, 1988.

[Mull 95] E. Muller. Fast Three Dimensional Form Measurement System. International Journal on Optical Engineering, Vol. 34, No. 9, pp 2754-2756, 1995.

[Mund 87] J.L. Mundy and G.B. Porter. A Three-Dimensional Sensor Based on Structured Light. In T. Kanade Three-Dimensional Machine Vision. Kluwer Academic Publishers, 1987.

[Ozek 86] O. Ozeki, T. Nakano and S. Yamamoto. Real-Time Range Measurement Device for Three-Dimensional Object Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, No. 4, pp 550-554, 1986.

[Pajd 95b] T. Pajdla. BCRF - Binary-Coded Illumination Range Finder Reimplementation. Technical Report No. KUL/ESAT/MI2/9502, Katholieke Universiteit Leuven, 1995.

[Popp 75] R.J. Popplestone, C.M. Brown, A.P. Ambler and G.F. Crawford. Forming Models of Plane-and-Cylinder Faceted Bodies from Light Stripes. In Proceedings International Joint Conference on Artificial Intelligence, pp 664-668, 1975.

[Posd 82] J.L. Posdamer and M.D. Altschuler. Surface Measurement by Space-Encoded Projected Beam Systems. International Journal on Computer Graphics and Image Processing, Vol. 18, pp 1-17, 1982.

[Reid 86] G.T. Reid. Automatic Fringe Pattern Analysis: A Review. International Journal on Optics and Lasers in Engineering, Vol. 7, pp 37-68, 1986.

[Rock 75] F. Rocker and A. Kiessling. Methods for Analyzing Three Dimensional Scenes. In Proceedings International Joint Conference on Artificial Intelligence, pp 669-673, 1975.

[Rote 95] F. Rotella and P. Borne. Théorie et Practique du Calcul Matriciel. Collection Méthodes et Practiques de l'Ingénieur. Editions Technip, 1995.

[Sato 82] Y. Sato, H. Kitagawa and H. Fujita. Shape Measurement of Curved Objects Using Multiple Slit-Ray Projections. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 4, No. 6, pp 641-646, 1982.

[Sato 86] K. Sato, H. Yamamoto and S. Inokuchi. Tuned Range Finder for High Precision 3D Data. In Proceedings International Conference on Pattern Recognition, pp 1168-1171, 1986.

[Sato 87] K. Sato and S. Inokuchi. Range-Imaging System Utilizing Nematic Liquid Crystal Mask. In Proceedings International Conference on Computer Vision, pp 657-661, 1987.

[Sato 96] K. Sato. Range Imaging Based on Moving Pattern Light and Spatio-Temporal Matched Filter. In Proceedings International Conference on Image Proceessing, Vol. 1, pp 33-36, 1996.

[Shir 71b] Y. Shirai and M. Suwa. Recognition of Polyhedrons with a Range Finder. In Proceedings International Joint Conference on Artificial Intelligence, pp 80-87, 1971.

[Shri 89] N. Shrikhande and G. Stockman. Surface Orientation from a Projected Grid. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, No.6, pp 650-655, 1989.

[Smut 96] V. Smutny and T. Pajdla. Rainbow Range Finder and its Implementation at the CVL. Technical Report No. K335-1996, Computer Vision Lab, Czech Technical University. 1996.

[Srin 89] V. Srinivasan and R. Lumia. A Pseudo-Interferometric Laser Range Finder for Robot Applications. IEEE International Journal of Robotics and Automation Vol.5, No.1, pp 98-105, 1989.

[Stoc 86] G. Stockman and G. Hu. Sensing 3-D Surface Patches Using a Projected Grid. In Proceedings International Conference on Computer Vision and Pattern Recognition, pp 602-607, 1986.

[Stoe 80] J. Stoer and R. Bulirsch. Introduction to Numerical Analysis. Springer-Verlag, 1980.

[Taji 90] J. Tajima and M. Iwakawa. 3-D Data Acquisition by Rainbow Range Finder. In Proceedings International Conference on Pattern Recognition, pp 309-313, 1990.

[Tosc 87] G. Toscani. Systèmes de Calibration et Perception du Mouvement en Vision Artificielle. Thèse Université Paris Sud. 1987.

[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 of Robotics and Automation, Vol. 3, No. 4, pp 323-344, 1987.

[Umed 96] K. Umeda and N. Suzuki. Gesture Recognition of Head Motion Using Range Images. In Proceedings International Conference on Intelligent Robots and Systems, pp 1594-1599, 1996.

[Vemu 86] B.C. Vemuri and J.K. Aggarwal. 3-D Model Construction from Multiple Views Using Range and Intensity Data. In Proceedings International Conference on Computer Vision and Pattern Recognition, pp 435-437, 1986.

[Vuyl 90] P. Vuylsteke and A. Oosterlinck. Range Image Acquisition with a Single Binary-Encoded Light Pattern. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.12, No. 2, pp 148-164, 1990.

[Wang 87] Y.F. Wang, A. Mitiche and J.K. Aggarwal. Computation of Surface Orientation and Structure of Objects Using Grid Coding. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 9, No. 1, pp 129-137, 1987.

[Wang 89] Y.F. Wang and P. Liang. A New Method for Computing Intrinsic Surface Properties. In Proceedings International Conference on Computer Vision and Pattern Recognition, pp 235-240, 1989.

[Wang 91] Y.F. Wang. Characterizing Three-Dimensional Surface Structures from Visual Images. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 13, No. 1, pp 52-60, 1991.

[Wang 92] Y.F. Wang and D.I. Cheng. Three-Dimensional Shape Construction and Recognition by Fusing Intensity and Structured Lighting. International Journal on Pattern Recognition, Vol. 25, No. 12, pp 1411-1425, 1992.

[Weng 92] 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, pp 965-980, 1992.

[Will 71] P.M. Will and K.S. Pennington. Grid Coding: A Preprocessing Technique for Robot and Machine Vision. In Proceedings International Joint Conference on Artificial Intelligence, pp 66-70, 1971.

[Wust 91] C. Wust and D.W. Capson. Surface Profile Measurement Using Color Fringe Projection. International Journal on Machine Vision and Applications, Vol. 4, pp 193-203, 1991.

[Yama 86] H. Yamamoto, K. Sato and S. Inokuchi. Range Imaging System Based on Binary Image Accumulation. In Proceedings International Conference on Pattern Recognition, pp 233-235, 1986.

[Yoko 94] A. Yokoyama, K. Sato, T. Yoshigahara and S. Inokuchi. Realtime Range Imaging Using Adjustment-Free Photo-VLSI -Silicon Range Finder-. In Proceedings of International Conference on Intelligent Robots and Systems, pp 1751-1758, 1994.

[Zhan 93b] Z. Zhang. Le problème de la mise en correspondance: L'état de l'art. INRIA Resarch Report N^o 2146, 1993.

[Zhan 96c] Z. Zhang. Determining the Epipolar Geometry and its Uncertainty: A Review. INRIA Research Report No. 2927, 1996.