Visual methods for three-dimensional modeling

Thesis by

Jean-Yves Bouguet

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

California Institute of Technology

Pasadena, California

1999

(Submitted May 25, 1999)

A mes parents,

à mon frère Florent,

à papy et mamie "de Deuil",

à mamie Bouguet.

Acknowledgements

I would like to express my deepest gratitude to my advisor, Professor Pietro Perona, who has been an invaluable teacher and friend during the past five years at Caltech. His insightful guidance and advice have made substantial differences in my work, as well as in my life. Pietro has always been present to support me when I was in doubt, criticize me when I erred, and encourage me when I needed it, in both professional and personal matters. I cannot thank him enough for all he did for me.

I would also like to thank the other members of my committee, Professors Erik K. Antonsson, James R. Arvo, Alan Barr and Demetri Psaltis for their support.

Caltech has been for me a fantastic environment to work and mature scientifically, and that, to a great part, has been due to the interaction and collaboration with many scientists. I am grateful to Professors James R. Arvo, Peter Schröder, Yaser S. Abu-Mostafa, P.P. Vaidyanathan and Christof Koch for sharing with me their knowledge and thinking. In particular, I thank Peter and Jim for their stimulating interaction in the teaching of the class on 3D photography that led to many of the ideas presented in chapters 5 and 6 of this thesis.

I am grateful to Dr. Larry Matthies and Dr. Andrew E. Johnson from the Jet Propulsion Laboratory for their very close collaboration on the "motion" project. A significant part of the work presented in chapter 4 would not have existed without their participation on this project.

The whole Caltech vision group has been a fantastic environment both at work and at play. I want to thank Silvio Savarese, Luis Goncalves, Dr. Enrico Di Bernardo, Mario Munich, Arrigo Benedetti, Enrico Ursella, Markus Weber, Dr. Alan Bond, Xiaolin Feng, Yang Song, Dr. Max Welling, Dr. Stefano Soatto, Dr. Jennifer Sun and Dr. Michael C. Burl for their support and exchange. Special thanks go to Silvio for his outstanding work on the real-time implementation of the shadow scanner. Markus Weber provided the original idea that gave birth to chapter 7.

I also want to thank other people at Caltech with whom I had very close interactions, on both a professional and a personal basis: Dr. George Barbastathis, Christophe Moser and Dr. Dean Schonfeld. I will never forget these long working nights spent in the vision lab with George on the tune of the "fifth element" and the "sweetest steak." George gave me a tremendous inspiration in my work, gave me advice, proof-read many of my writings, encouraged me when I needed it. He is a fantastic friend, and I cannot possibly thank him enough for all that he did for me.

Of course, many thanks go to my other close friends at Caltech, Wayez Ahmad, Alberto Pesavento, Francesco Bullo, Diego Dugatkin, Gudrun Socher, Dieter Koller, Melissa Saenz, Laurent Chognard, Pili Munich, Barbara Di Bernardo for all the fantastic moments we spent together. Special congratulations go to my roommate Wayez, "The Commander," for having put up with me for over three years at the beginning of my time at Caltech.

Throughout my graduate studies, I have also had the great privilege to interact with a number of prestigious research figures outside Caltech. My deepest gratitude goes to Dr. Paul Debevec, and Professors Brian Curless, Marc Levoy, Steven Seitz and Wolfgang Stuerzlinger.

And of course, none of this could have happened without my parents, brother, grandparents and family and their unflagging love and faith in me. They were always present to support me when tough decisions were demanded. For their unassuming love, I cannot thank them enough.

Abstract

Most animals use vision as a primary sensor to interact with their environment. Navigation or manipulation of objects are among the tasks that can be better achieved while understanding the thre- dimensional structure of the scene.

In this thesis, we present a variety of computational techniques for estimating 3D shape from 2D images, based on both passive and active technologies.

The first proposed method is purely passive. In this technique, a single camera is moved in an unconstrained manner around the scene to model as it acquires a sequence of images. The reconstruction process consists then of retrieving the trajectory of the camera, as well as the 3D structure of the scene using only the information contained in the images.

The second method is based on active lighting technology. In the philosophy of standard 3D scanning methods, a projector is used to project light patterns in the scene. The shape of the scene is then inferred from the way the patterns deform on the objects. The main novelty of our scheme compared to traditional methods is in the nature of the patterns, and the type of image processing associated to them. Instead of using standard binary patterns made out of black and white stripes, our scheme uses a sequence of grayscale patterns with a sinusoidal profile in brightness intensity. This choice allows us to establish correspondence (between camera image, and projector image) in a dense fashion, leading to depth computation at (almost) every pixel in the image.

The last reconstruction method that we propose in this thesis is an alternative 3D scanning scheme that does not require any other device besides a camera. The main idea is to substitute the projector by a standard light source (such as a desk lamp), and use a pencil (or any other object with a straight edge) to cast planar shadows in the scene. The 3D geometry of the scene is then inferred from the way the shadow naturally deforms on the objects in the scene. Since this technology is largely inspired

from structured lighting techniques, we call it 'weakly structured lighting.'

Contents

List of Figures

Chapter 1 Introduction

1.1 Different approaches to 3D reconstruction

For more than two decades, the problem of reconstructing the three-dimensional structure of the surrounding scene from a set of 2D images has been subject to a lot of attention in the computational vision research community. Navigation or manipulation of objects are among the tasks that can be better achieved while understanding the three-dimensional structure of the scene.

This process of "understanding"* the three-dimensional structure of the world from 2D visual observations (e.g., pictures) is one of the most valuable functions of our visual system. It is believed that this task is achieved by integrating a number of visual cues (perspective deformation, stereo, shading, shadows, occluding contours, (de)focus, texture, motion parallax, highlights) that naturally exist in most image observations of the world in conjunction with higher level cues (such as prior knowledge about the scene). In many circumstances, even a single image of an object is sufficient to a human subject for extracting a good mental representation of its three-dimensional shape as well as the material it is made of. It has also been shown that such a task may still be achieved in absence of prior knowledge† about the observed object. See [1, 2, 3].

Figure 1.1 illustrates the remarkable ability of the human visual system to infer 3D shape from a number of "pictorial" cues.

In spite of the large research effort devoted for more than forty years in trying to understand the fundamental neurobiological building blocks constituting our visual system (from the retina to the visual cortex), very little is known to this day. From

*The term "understanding" has to be taken in the broad sense. It can be substituted by "extracting a representation of."

†Sometimes aside from symmetry or smoothness assumptions.

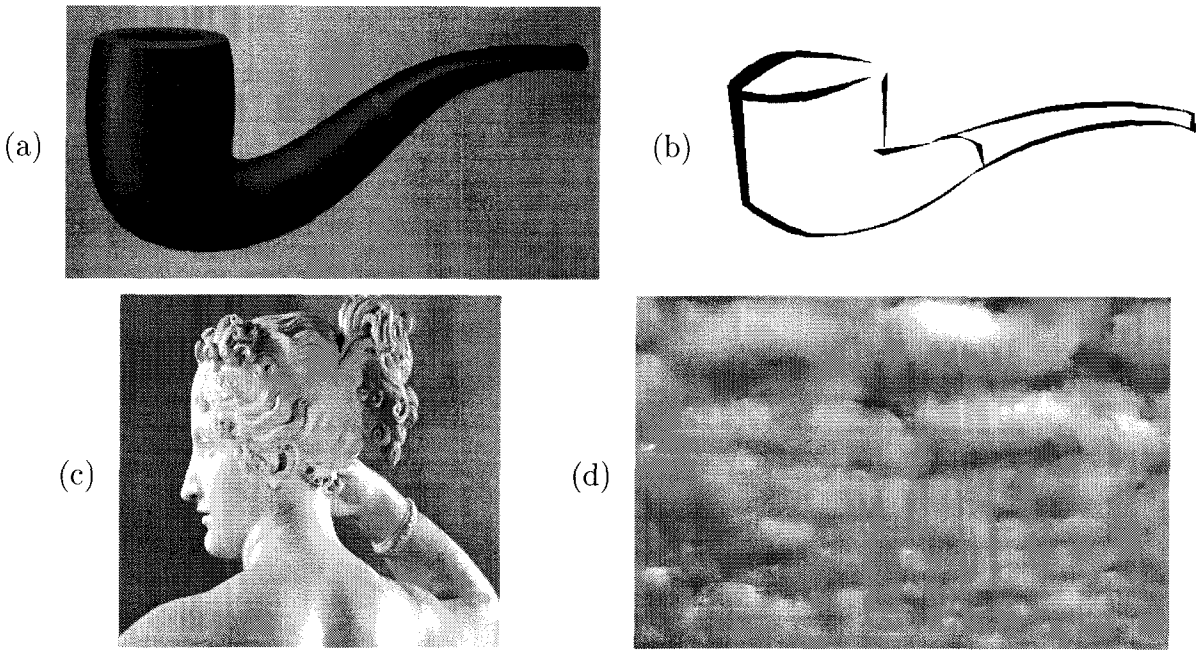


Figure 1.1: Our visual system uses many "pictorial" cues for inferring 3D shape. Among those are shading (a,c), contours (d), and texture (d).

a biological point of view, vision is probably the sense that is the most complex, in architectural sense as well as in functional sense [1, 2, 3].

From a computational (or engineering) point of view, the main goal of computer vision researchers is to design artificial systems that would replicate the function of the human visual system. In relation to the problem we are targeting, this ideal system would be able to automatically extract the three-dimensional geometry and surface properties of the world using only the information contained in a set of 2D pictures of the world. Ideally, the complete 3D model of an object would be computed (3D geometry + surface properties) as it is presented in front of a camera (possibly with several poses).

The applications of such a 3D modeling device are numerous. Perhaps the most important ones are animation and entertainment, industrial design, archiving, virtual visits to museums, and commercial on-line catalogues (e-commerce).

Other techniques have been proposed to serve similar applications. Among those, image-based approaches consist of generating novel views of a scene from an initial set of acquired images without explicitly computing the scene geometry. Those tech-

niques are quite successful in dealing with very complex scenes where geometry is difficult to reconstruct[‡] but often require a very large set of images densely acquired throughout the entire viewing sphere. Consequently, the amount of data required to be stored in memory is very often extremely large making these approaches difficult to be used in internet applications[§]. In contrast, model-based approaches allow to synthesize any novel view of the scene (by direct rendering) from a unique 3D model. In comparison to image-based techniques, model-based techniques may be viewed as ways to perform "smart" image stream compression. The information contained in an entire set of images is summarized into a single compact 3D model that is then rendered very efficiently using existing rendering platforms.

Then, the central goal is designing a system that would automatically acquire the three-dimensional model of an object from a set of pictures.

Unfortunately, we are still very far from such an ideal system. To this day, the only practical systems for 3D modeling are based on active lighting technology. Most of these systems consist of a combination of passive imaging devices (one or more camera(s)) and active devices (laser/LCD projectors) calibrated one with respect to each other. The principle of these systems is quite intuitive: the projector emits light patterns that are reflected by the scene and detected on the image acquired by the camera. The three-dimensional structure of the scene is then computed by geometrical triangulation (this is also known as optical triangulation). Aside from being quite insensitive to variations in texture within the scenes, this technology has the advantage of yielding very good accuracies (errors in reconstruction can be as small as a part in one or two thousand). The main drawback of standard active lighting systems is the cost: motorized transport of the object and active (laser, LCD projector) lighting of the scene makes them very accurate, but unfortunately expensive [4, 5, 6, 7, 8, 9].

An interesting challenge for vision scientists is to design systems that would use only images acquired in natural light for computing 3D geometry. In contrast to

[‡]For example fur material.

[§]For internet applications, download time is a crucial factor.

active techniques that use an external projecting device, this class of techniques is also called passive. Among all the passive cues that contain information about 3D shape (stereoscopic disparity, texture, motion parallax, (de)focus, shadows, shading and specularities, occluding contours and other surface discontinuities), at the current state of vision research, stereoscopic disparity is the single passive cue that reliably gives reasonable accuracy. Unfortunately it has two major drawbacks: it requires two cameras thus increasing complexity and cost, and it cannot be used on untextured surfaces, which are common for industrially manufactured objects.

An extension of stereo techniques consists of substituting the pair of cameras with a single moving camera. In that case, a single camera takes two snapshots of the world from different locations in space at two different instants in time. The reconstruction procedure is then identical to traditional stereo scenarios: the structure of the world is triangulated using the two camera images used as a stereo pair. The first advantage of such an approach is in the cost: one camera is necessary instead of two. The second advantage is in ergonomy: the camera can be moved freely by the user in the scene, and no specific exterior calibration is required[¶]. This class of technique is also known as Structure From Motion. There exists, however, two major limitations of such an approach. First, as the camera is moved in a unconstrained manner, the motion disparity between the two camera positions is not known, and therefore it must be computed as well (that is a required step for enabling geometrical triangulation). Second, since the two pictures are taken at different times, the world must remain rigid between those two acquisition times. That is also known as the *rigidity assumption.* Although there exist partial answers in the vision literature to the issue regarding computing the motion disparity between the two camera positions from the two images alone (see for example [10, 11]), this problem is still largely regarded as an open research issue. The extent of the work that needs to be done is even greater when considering a scenario where a longer stream of images (more than two) is acquired as the camera explores the entire surface of the scene (in order to achieve

[¶]We are referring here to the measurement of the location and orientation of two cameras one with respect to each other that is required in any stereo system.

a complete 3D reconstruction). In that case, camera position needs to be computed at every image prior to 3D structure reconstruction. Autonomous navigation is one straightforward application of such systems.

In the context of this thesis, we mainly focus on 3D modeling applications, and more specifically on the problem of estimating the 3D geometry (estimating the surface properties is another important task in modeling).

With that specific goal in mind, we will present a variety of techniques for estimating 3D shape, based on both passive and active technologies.

1.2 Outline of the thesis

Chapter 2 introduces the fundamental notation used in the thesis. The basic geometrical elements that are used for reconstruction are presented (points, lines, planes) together with their mathematical representations. In this chapter we also introduce a new mathematical formalism that we call 'B-dual-space geometry' that enables us to explore and compute geometrical properties of three-dimensional scenes with simple and compact algebra. The main contribution of this work is a new parameterization of planes in space that leads to a set of useful properties.

Chapter 3 presents a direct application of the dual-space formalism to the problem of camera calibration. The essential results are closed-form solutions for calibration in the case of several calibration models. In this chapter, we show how dual-space geometry enables us to study the observability of the different camera models in a very intuitive manner.

The four following chapters (4, 5, 6 and 7) describe four different techniques for 3D reconstruction. These schemes are in most parts independent from each other. Therefore, a reader only interested in one particular approach for shape estimation may read only the associated chapter describing it while skipping the other chapters.

Let us give a brief description of the content of each of those chapters.

Chapter 4 describes passive visual techniques for 3D shape estimation. The fundamental mathematical tools are presented, in the case of image sequences consisting

of 2, 3 or more than three camera views. As mentioned earlier, when the camera is moved freely within the environment, its position must be computed at every image to enable structure estimation. This "camera position computation" is particularly challenging as the number of views in the sequence is large (and the camera trajectory is long). In that chapter, we propose a technique for performing this computation that is sufficiently reliable and consistent for enabling accurate shape estimation.

Chapter 5 introduces a novel active lighting technique for 3D scanning. In the philosophy of standard scanning methods, a projector is used to project light patterns in the scene. The three-dimensional shape of the scene is then inferred from the way the patterns deform on the objects. The main difference of our scheme compared to traditional methods is in the nature of the patterns, and the type of image processing associated to them. Traditional techniques use binary patterns consisting of stripes, or other sharp boundaries (e.g., a laser sheet) and depth is computed along those boundaries through optical triangulation. Our approach uses a sequence of grayscale patterns with a sinusoidal profile in brightness intensity. This choice of patterns allows us to establish correspondence (between camera image, and projector image) in a dense fashion in the image plane. This is done through a processing based on temporal analysis. This new scheme leads to depth information at (almost) every pixel in the image.

The following chapter 6 describes a new technique for capturing 3D surfaces that is based on using planar shadows. As mentioned earlier, standard structured lighting techniques use an additional computer-controlled active device (e.g., an LCD projector) to project light patterns in the scene (see chapter 5). This device makes most systems expensive and bulky. The main idea underlining the new method is in substituting a simple desk lamp for the complex active device. Then, instead of projecting stripe patterns (or grayscale patterns as described in chapter 5), the user casts a planar shadow in the scene by holding a regular pencil (or anything else with a straight edge) between the light source and the scene. The three-dimensional shape of the scene is then computed from the way the shadow deforms in the scene. Since this technology is largely inspired from structured light lighting techniques, we call it

'weakly structured lighting.' Once again, the processing is based on temporal analysis. This enables scene depth computation at (almost) every pixel in the image. In this chapter, we also demonstrate that this reconstruction scheme may be used in outdoor scenes (for scanning large objects) by substituting the desk lamp with the sun.

As described in chapter 6, our shadow scanning method requires the presence of a background plane used as a reference surface. In the following chapter 7 we provide a solution for 3D reconstruction (still with planar shadows) in the case where there is no such plane in the scene. Once again, the dual-space formalism is used as a fundamental tool for all mathematical derivations[‖].

Each chapter contains an experimental section to address the accuracies of each of those reconstruction schemes.

Finally, in chapter 8 we conclude, and discuss some directions for future developments.

[‖]This chapter addresses mostly theoretical aspects of Euclidean reconstructions from planar shadows only. The reader interested in practical solutions for 3D scanning could concentrate on implementing the shadow scanning method described in the previous chapter 6.

Chapter 2 B-Dual-Space geometry

This chapter introduces the fundamental notation used in the thesis. Section 2.1 defines the basic geometrical elements that are used for reconstruction: points, lines and planes in space and point and lines on the image plane. This section also defines perspective projection as the fundamental image projection operator (from 3D to 2D) as well as rigid body motion transformation. All definitions are given in Euclidean space as well as in projective geometry. The following section 2.2 defines a new mathematical formalism called B-dual-space geometry derived from projective geometry. This formalism enables us to explore and compute geometrical properties of three-dimensional scenes with simple and compact notation. This will be illustrated in the following chapter when applying that formalism to the problem of camera calibration.

2.1 Standard notation

2.1.1 Euclidean space - Camera reference frame

Let (E) be the 3D Euclidean space. For a given position of a camera in space, we define $\mathcal{F} = (0_c, X_c, Y_c, Z_c)$ as the standard frame of reference (called "camera reference frame") where O_c is the camera center of projection, and the three axes (O_c, X_c), (O_c, Y_c) and (O_c, Z_c) are mutually orthogonal and right-handed ((O_c, X_c) and (O_c, Y_c) are chosen parallel to the image plane). See figure 2.1.

We may then refer to a point P in space by its corresponding Euclidean coordinate vector $\overline{X} = [X \;\; Y \;\; Z]^T$ in that reference frame $\mathcal{F}$. The Euclidean space may also be viewed as a three-dimensional projective space $\mathcal{P}^3$. In that representation, the point P is alternatively represented by the homogeneous 4-vector $\overline{\mathbf{X}} \simeq [X \;\; Y \;\; Z \;\; 1]^T$. The sign $\simeq$ denotes a vector equality up to a non-zero scalar. Therefore, any scaled version of $[X \;\; Y \;\; Z \;\; 1]^T$ represents the same point in space.

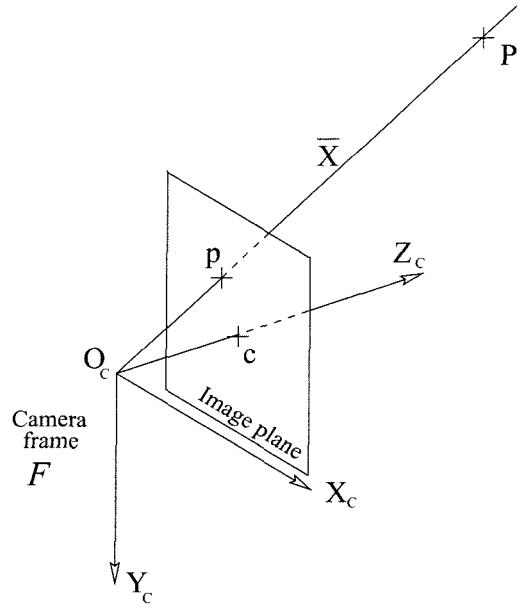


Figure 2.1: In the reference frame attached to the camera $\mathcal{F} = (O_c, X_c, Y_c, Z_c)$, a point P in space has coordinates $\overline{X} = [X \;\; Y \;\; Z]^T$. Its perspective projection p on the image plane has coordinates $\overline{x} = [X/Z \;\; Y/Z]^T$.

A plane Π in space is defined as the set of points P of homogeneous coordinate vector $\overline{\mathbf{X}}$ that satisfy:

$$\langle \overline{\pi}, \overline{\mathbf{X}} \rangle = 0 \tag{2.1}$$

where $\overline{\pi} \simeq [\pi_x \;\; \pi_y \;\; \pi_z \;\; \pi_t]^T$ is the homogeneous 4-vector parameterizing the plane Π ($\langle . \rangle$ is the standard scalar product operator). Observe that if $\overline{\pi}$ is normalized such that $\pi_x^2 + \pi_y^2 + \pi_z^2 = 1$, then $\overline{n}_\pi = [\pi_x \;\; \pi_y \;\; \pi_z]^T$ is the normal vector of the plane Π (in the camera reference frame $\mathcal{F}$) and $d_\pi = -\pi_t$ its orthogonal (algebraic) distance to the camera center O_c.

2.1.2 Image plane and perspective projection

Let (I) be the 2D image plane. The image reference frame is defined as (c, x_c, y_c) where c is the intersection point between (O_c, Z_c) (*optical axis*) and the image plane, and (c, x_c) and (c, y_c) are the two main image coordinate axes (parallel to (O_c, X_c) and (O_c, Y_c)). See figure 2.1. The point c is also called *optical center* or *principal*

point.

Let p be the projection on the image plane of a given point P of coordinates $\overline{X} = [X \quad Y \quad Z]^T$, and denote $\overline{x} = [x \quad y]^T$ its coordinate vector on the image plane. Then, the two vectors $\overline{X}$ and $\overline{x}$ are related through the perspective projection equation:

$$\overline{x} = \begin{bmatrix} x \\ y \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} X \\ Y \end{bmatrix} \tag{2.2}$$

This projection model is also referred to as a "pinhole" camera model.

In analogy to Euclidean space, it is sometimes useful to view the image plane as a two-dimensional projective space $\mathcal{P}^2$. In that representation, a point p on the image plane has homogeneous coordinate vector $\overline{\mathbf{x}} \simeq [x \quad y \quad 1]^T$. Similarly to $\mathcal{P}^3$, any scaled version of $[x \quad y \quad 1]^T$ describes the same point on the image plane.

One advantage of using projective geometry is that the projection operator defined in equation 2.2 becomes a linear operator from $\mathcal{P}^3$ to $\mathcal{P}^2$:

$$\overline{\mathbf{x}} \simeq \mathbf{P}\,\overline{\mathbf{X}} \qquad \text{with} \qquad \mathbf{P} = \begin{bmatrix} I_{3\times 3} & 0_{3\times 1} \end{bmatrix} \tag{2.3}$$

where $\overline{\mathbf{X}}$ and $\overline{\mathbf{x}}$ are the homogeneous coordinates of P and p respectively, $I_{3\times 3}$ is the 3×3 identity matrix and $0_{3\times 1}$ is the 3×1 zero-vector. Observe from equation 2.3 that $\overline{\mathbf{x}}$ is equal (up to a scale) to the Euclidean coordinate vector $\overline{X} = [X \quad Y \quad Z]$ of P:

$$\overline{\mathbf{x}} \simeq \overline{X} \tag{2.4}$$

Therefore $\overline{\mathbf{x}}$ is also referred to as the optical ray direction associated to P.

A line λ on the image plane is defined as the set of points p of homogeneous coordinate vectors $\overline{\mathbf{x}}$ that satisfy:

$$\langle \overline{\lambda}, \overline{\mathbf{x}} \rangle = 0 \tag{2.5}$$

where $\overline{\lambda} = [\lambda_x \ \ \lambda_y \ \ \lambda_z]^T$ is the homogeneous 3-vector defining the line λ. Observe that if $\overline{\lambda}$ is normalized such that $\lambda_x^2 + \lambda_y^2 = 1$, then $\overline{n}_\lambda = [\lambda_x \ \ \lambda_y]^T$ is the normal vector of the line λ (in the image reference frame) and $d_\lambda = -\lambda_z$ its orthogonal (algebraic) distance to the principal point c.

Claim 1: Let p_1 and p_2 be two distinct points on the image plane with respective homogeneous coordinate vectors $\overline{\mathbf{x}}_1$ and $\overline{\mathbf{x}}_2$. Then, it is straightforward to show that the line λ connecting p_1 and p_2 has homogeneous coordinate vector $\overline{\lambda} \simeq \overline{\mathbf{x}}_1 \times \overline{\mathbf{x}}_2$, where $\times$ is the standard vector product operator in $\mathbb{R}^3$.

Claim 2: Let λ_1 and λ_2 be two distinct lines on the image plane with respective homogeneous coordinate vectors $\overline{\lambda}_1$ and $\overline{\lambda}_2$. Then, the point of intersection p between the two lines λ_1 and λ_2 has homogeneous coordinate vector $\overline{\mathbf{x}} \simeq \overline{\lambda}_1 \times \overline{\lambda}_2$. If the two lines are parallel then the last coordinate of $\overline{\mathbf{x}}$ is zero. In that case p is a point at infinity.

There exists useful relations between lines on the image plane and planes in space, as illustrated by the two following examples.

Example 1: Consider a line λ on the image plane of coordinate vector $\overline{\lambda} = [\lambda_x \ \ \lambda_y \ \ \lambda_z]^T$. Then, the set of points P in space that project onto λ is precisely the plane Π_λ spanned by λ and the camera center O_c (see figure 2.2). Let $\overline{\pi}_\lambda$ be the coordinate vector of Π_λ. Let us compute $\overline{\pi}_\lambda$ as a function of $\overline{\lambda}$. According to equations 2.3 and 2.5, a point P of homogeneous coordinate vector $\overline{\mathbf{X}}$ will lie on Π_λ if and only if:

$$\left\langle \overline{\lambda}, \mathbf{P}\,\overline{\mathbf{X}} \right\rangle = 0 \tag{2.6}$$

this relation enforces the projection of P to lie on λ. This may be alternatively written:

$$\left\langle \mathbf{P}^T\,\overline{\lambda}, \overline{\mathbf{X}} \right\rangle = 0. \tag{2.7}$$

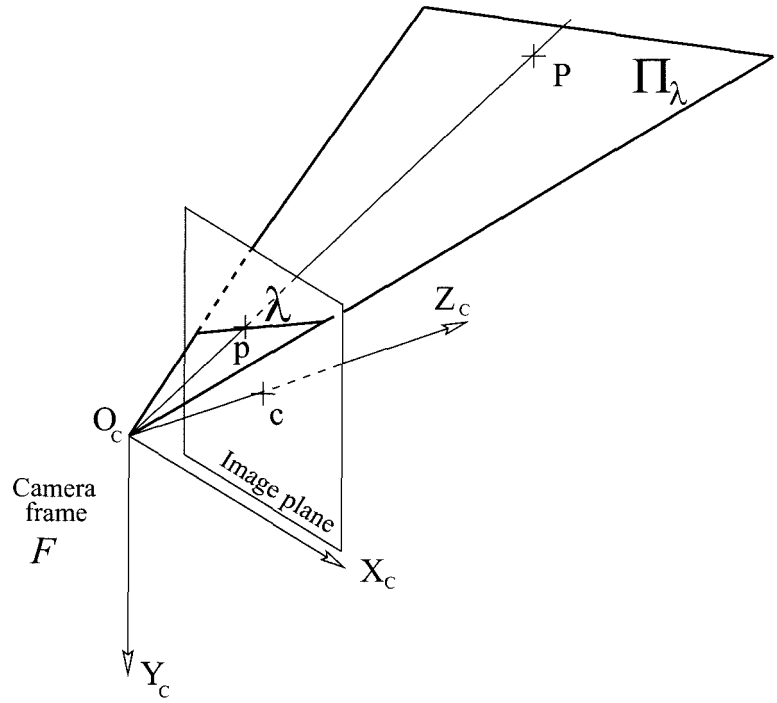


Figure 2.2: Any point P on Π_λ projects onto the line λ on the image plane. We say that the plane Π_λ is spanned by λ.

Therefore the plane coordinates $\overline{\pi}_\lambda$ has the following expression:

$$\overline{\pi}_\lambda \simeq \mathbf{P}^T \overline{\lambda} = \begin{bmatrix} \overline{\lambda} \\ 0 \end{bmatrix} = \begin{bmatrix} \lambda_x \\ \lambda_y \\ \lambda_z \\ 0 \end{bmatrix} \tag{2.8}$$

Example 2: Consider two planes in space Π_1 and Π_2 of respective coordinate vectors $\overline{\pi}_1 \simeq [\pi_{x_1} \quad \pi_{y_1} \quad \pi_{z_1} \quad \pi_{t_1}]^T$ and $\overline{\pi}_2 \simeq [\pi_{x_2} \quad \pi_{y_2} \quad \pi_{z_2} \quad \pi_{t_2}]^T$. Assume the two planes intersect along line Λ in space, and call λ the resulting image line after projection of Λ onto the image plane. Let us compute the coordinate vector $\overline{\lambda}$ of λ as a function of $\overline{\pi}_1$ and $\overline{\pi}_2$. Consider a point P on Λ and denote p its projection on the image plane. Since P lies on Π_1 and Π_2, its homogeneous coordinate vector $\overline{\mathbf{X}} \simeq [X \quad Y \quad Z \quad 1]^T$

must satisfy the following system:

$$\begin{cases} \pi_{x_1} X + \pi_{y_1} Y + \pi_{z_1} Z + \pi_{t_1} = 0 & (L_1) \\ \pi_{x_2} X + \pi_{y_2} Y + \pi_{z_2} Z + \pi_{t_2} = 0 & (L_2) \end{cases} \tag{2.9}$$

This system yields:

$$(\pi_{t_2}\,\pi_{x_1} - \pi_{t_1}\,\pi_{x_2})\,X + (\pi_{t_2}\,\pi_{y_1} - \pi_{t_1}\,\pi_{y_2})\,Y + (\pi_{t_2}\,\pi_{z_1} - \pi_{t_1}\,\pi_{z_2})\,Z = 0. \tag{2.10}$$

Since the homogeneous coordinate vector of p is $\overline{\mathbf{x}} \simeq \overline{X} = [X \;\; Y \;\; Z]^T$, equation 2.10 reduces to a standard image line equation:

$$\langle \overline{\lambda}, \overline{\mathbf{x}} \rangle = 0 \tag{2.11}$$

where

$$\overline{\lambda} \simeq \begin{bmatrix} \pi_{t_2}\,\pi_{x_1} - \pi_{t_1}\,\pi_{x_2} \\ \pi_{t_2}\,\pi_{y_1} - \pi_{t_1}\,\pi_{y_2} \\ \pi_{t_2}\,\pi_{z_1} - \pi_{t_1}\,\pi_{z_2} \end{bmatrix} \tag{2.12}$$

that is the coordinate vector of λ, projection of $\Lambda = \Pi_1 \cap \Pi_2$.

2.1.3 Rigid body motion transformation

Consider a set of N points P_i in space ($i = 1, \ldots, N$), and let $\overline{X}_i = [X_i \;\; Y_i \;\; Z_i]^T$ be their respective coordinate vectors in the camera reference frame $\mathcal{F}$. Suppose the camera moves to a new location in space, and let $\overline{X}'_i = [X'_i \;\; Y'_i \;\; Z'_i]^T$ be the coordinate vectors of the same points P_i in the new camera reference frame $\mathcal{F}'$ (see figure 2.4). Then $\overline{X}_i$ and $\overline{X}'_i$ are related to each other through a rigid body motion transformation:

$$\forall i = (1, \ldots, N), \qquad \overline{X}'_i = R\,\overline{X}_i + T \tag{2.13}$$

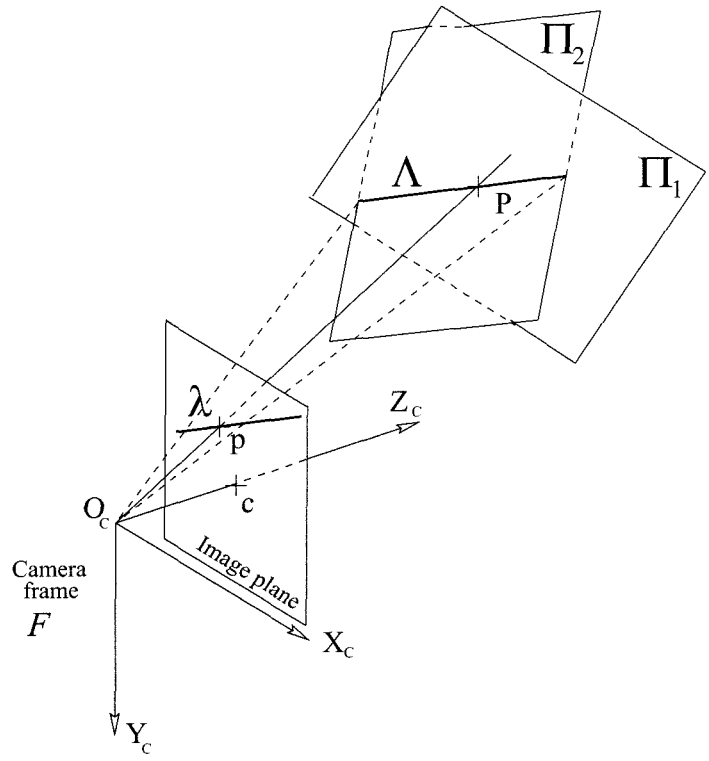


Figure 2.3: The two planes Π_1 and Π_2 intersect along the line Λ in space. The line λ is the projection of Λ on the image plane.

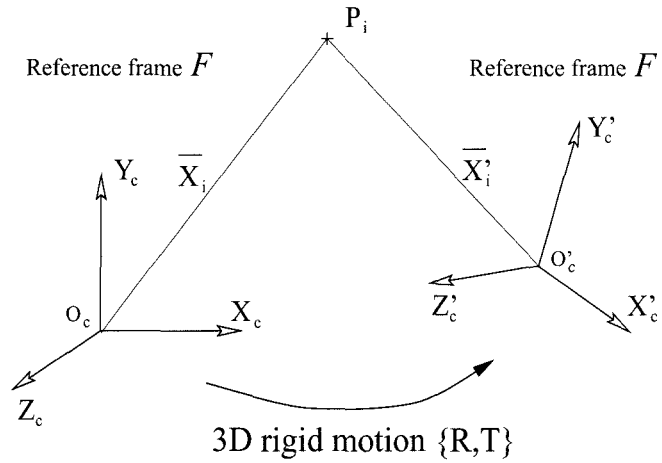


Figure 2.4: Rigid body motion transformation between camera frames $\mathcal{F} = (O_c, X_c, Y_c, Z_c)$ and $\mathcal{F}' = (O'_c, X'_c, Y'_c, Z'_c)$. The two coordinate vectors $\overline{X}_i$ and $\overline{X}'_i$ of P_i in $\mathcal{F}$ and $\mathcal{F}'$ are related to each other through the rigid body motion equation $\overline{X}'_i = R\overline{X}_i + T$.

where $R \in SO(3)^*$ and T are respectively a 3×3 rotation matrix and a 3-vector that uniquely define the rigid motion between the two camera positions. The matrix R is defined by a rotation vector $\overline{\Omega} = [\Omega_x \;\; \Omega_y \;\; \Omega_z]^T$ such that:

$$R = e^{\overline{\Omega}\wedge} \tag{2.14}$$

where $\overline{\Omega}\wedge$ is the following skew-symmetric matrix:

$$\overline{\Omega}\wedge = \begin{bmatrix} 0 & -\Omega_z & \Omega_y \\ \Omega_z & 0 & -\Omega_x \\ -\Omega_y & \Omega_x & 0 \end{bmatrix} \tag{2.15}$$

Equation 2.14 may also be written in a compact form using the Rodrigues' formula [10]:

$$R = I_{3\times 3}\,\cos(\theta) + \left[\overline{\Omega}\wedge\right]\,\frac{\sin(\theta)}{\theta} + \left[\overline{\Omega}\,\overline{\Omega}^T\right]\,\frac{1-\cos(\theta)}{\theta^2} \tag{2.16}$$

where $\theta = \|\overline{\Omega}\|$, and $\overline{\Omega}\,\overline{\Omega}^T$ is the following semi-positive definite matrix:

$$\overline{\Omega}\,\overline{\Omega}^T = \begin{bmatrix} \Omega_x^2 & \Omega_x\,\Omega_y & \Omega_x\,\Omega_z \\ \Omega_y\,\Omega_x & \Omega_y^2 & \Omega_y\,\Omega_z \\ \Omega_z\,\Omega_x & \Omega_z\,\Omega_y & \Omega_z^2 \end{bmatrix} \tag{2.17}$$

The fundamental rigid body motion equation 2.13 may also be written in projective space $\mathcal{P}^3$. In $\mathcal{P}^3$, the point P_i has homogeneous coordinate vectors $\overline{\mathbf{X}}_i \simeq [X_i \;\; Y_i \;\; Z_i \;\; 1]^T$ and $\overline{\mathbf{X}}'_i \simeq [X'_i \;\; Y'_i \;\; Z'_i \;\; 1]^T$ in the first ($\mathcal{F}$) and second ($\mathcal{F}'$) reference frames respectively. Then, equation 2.13 may be written:

$$\overline{\mathbf{X}}'_i \simeq D\,\overline{\mathbf{X}}_i \qquad \text{with} \qquad D = \begin{bmatrix} R & T \\ 0_{1\times 3} & 1 \end{bmatrix} \tag{2.18}$$

where $0_{1\times 3}$ is a 1×3 zero row vector. Observe that the inverse relation may also be

*Special Orthogonal 3×3 matrices.

written as follows:

$$\overline{\mathbf{X}}_i \simeq D^{-1}\,\overline{\mathbf{X}}'_i \qquad \text{with} \qquad D^{-1} = \begin{bmatrix} R^T & -R^T T \\ 0_{1\times 3} & 1 \end{bmatrix} \tag{2.19}$$

Let p'_i be the projection of P_i onto the second camera image plane, and let $\overline{\mathbf{x}}'_i$ be the homogeneous coordinate vector. Then, following the equation 2.3, we have:

$$\overline{\mathbf{x}}'_i \simeq \mathbf{P}\,\overline{\mathbf{X}}'_i \tag{2.20}$$

which may be also written:

$$\overline{\mathbf{x}}'_i \simeq \mathbf{P}'\,\overline{\mathbf{X}}_i \tag{2.21}$$

where: which may be also written:

$$\mathbf{P}' = \mathbf{P}\,D = \begin{bmatrix} R & T \end{bmatrix} \tag{2.22}$$

The matrix $\mathbf{P}'$ is the projection matrix associated to the second camera location.

Consider now a plane Π of homogeneous coordinate vectors $\overline{\pi}$ and $\overline{\pi}'$ in both camera reference frames $\mathcal{F}$ and $\mathcal{F}'$. How do $\overline{\pi}$ and $\overline{\pi}'$ relate to each other? Consider a generic point P on Π with homogeneous coordinate vectors $\overline{\mathbf{X}}$ and $\overline{\mathbf{X}}'$ in both reference frames. According to equation 2.1, we have:

$$\langle \overline{\pi}, \overline{\mathbf{X}} \rangle = 0 \tag{2.23}$$

which successively implies:

$$\langle \overline{\pi}, D^{-1}\,\overline{\mathbf{X}}'_i \rangle = 0 \tag{2.24}$$

$$\langle D^{-T}\,\overline{\pi}, \overline{\mathbf{X}}'_i \rangle = 0. \tag{2.25}$$

Therefore:

$$\overline{\pi}' \simeq D^{-T}\,\overline{\pi} = \begin{bmatrix} R & 0_{3\times 1} \\ -T^T R & 1 \end{bmatrix} \overline{\pi} \tag{2.26}$$

Similarly, the plane coordinate vector before motion $\overline{\pi}$ may be retrieved from $\overline{\pi}'$ through the inverse expression:

$$\overline{\pi} \simeq D^{T}\,\overline{\pi}' = \begin{bmatrix} R^T & 0_{3\times 1} \\ T^T & 1 \end{bmatrix} \overline{\pi}' \tag{2.27}$$

In order to put in practice these concepts, let us go through the following example:

Example 3: In the second reference frame $\mathcal{F}'$ (after camera motion), consider a line λ' on the image plane, and the plane Π that this line spans with the camera center (similarly to example 1 - see figure 2.2). Let $\overline{\lambda}'$ and $\overline{\pi}'$ be the homogeneous coordinate vectors of λ' and Π in $\mathcal{F}'$. See figure 2.5. Let us compute $\overline{\pi}$, the coordinate vector of Π in the initial camera reference frame $\mathcal{F}$ (before motion) as a function of $\overline{\lambda}'$, R and T.

According to equation 2.8, $\overline{\pi}'$ and $\overline{\lambda}'$ are related through the following expression:

$$\overline{\pi}' \simeq \begin{bmatrix} \overline{\lambda}' \\ 0 \end{bmatrix} \tag{2.28}$$

Then, $\overline{\pi}$ may be calculated from $\overline{\pi}'$ using equation 2.27:

$$\overline{\pi} \simeq D^{T}\,\overline{\pi}' = \begin{bmatrix} R^T & 0_{3\times 1} \\ T^T & 1 \end{bmatrix} \begin{bmatrix} \overline{\lambda}' \\ 0 \end{bmatrix} = \begin{bmatrix} R^T\,\overline{\lambda}' \\ \langle T, \overline{\lambda}' \rangle \end{bmatrix} = \mathbf{P}'^{T}\,\overline{\lambda}' \tag{2.29}$$

where $\mathbf{P}'$ is the projection matrix associated to the second camera location (eq. 2.22). Observe the similarity between equations 2.8 and 2.29.

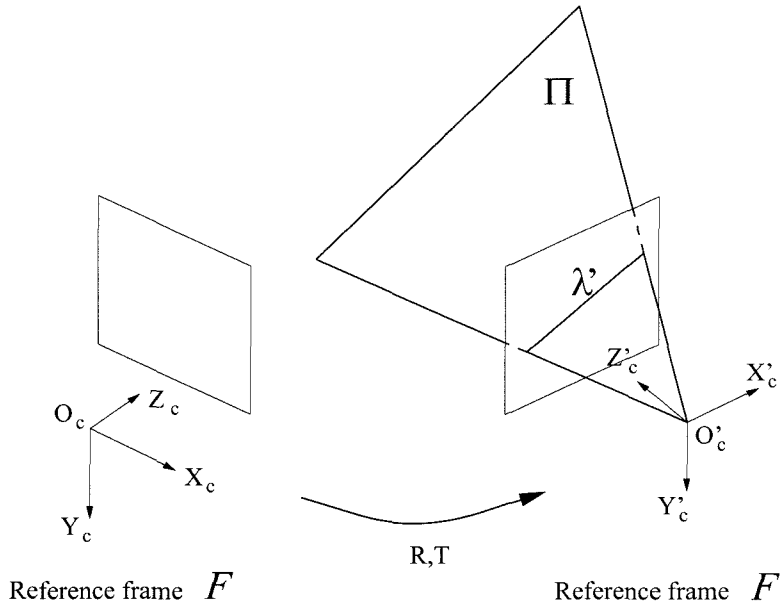


Figure 2.5: The plane Π is spanned by the line λ' observed on the image plane after camera motion (in frame $\mathcal{F}'$).

2.2 B-dual-space geometry

2.2.1 Definition of B-dual-space

As presented in the previous section, a plane Π in space is represented by an homogeneous 4-vector $\overline{\pi} \simeq [\pi_x \ \ \pi_y \ \ \pi_z \ \ \pi_t]^T$ in the camera reference frame $\mathcal{F} = (0_c, X_c, Y_c, Z_c)$ (see equation 2.1). Alternatively, if Π does not contain the camera center O_c (origin of $\mathcal{F}$) then it may be represented by a 3-vector $\overline{\omega} = [\omega_x \ \ \omega_y \ \ \omega_z]^T$, such that:

$$\langle \overline{\omega}, \overline{X} \rangle = 1 \tag{2.30}$$

for any point $P \in \Pi$ of coordinate vector $\overline{X} = [X \ \ Y \ \ Z]^T$ in $\mathcal{F}$. Notice that $\overline{\omega} \doteq \overline{n}_\pi / d_\pi$ where $\overline{n}_\pi$ is the unitary normal vector of the plane and $d_\pi \neq 0$ its distance to the origin. Let $(\Omega) = \mathbb{R}^3$. Since every point $\overline{\omega} \in (\Omega)$ corresponds to a unique plane Π in Euclidean space (E), we refer to (Ω) as the 'plane space' or 'B-dual-space'. For brevity in notation, we will often refer to this space as the *dual-space*. There exists a simple relationship between plane coordinates in projective geometry and dual-space

geometry:

$$\overline{\omega} = -\frac{1}{\pi_t}\begin{bmatrix} \pi_x \\ \pi_y \\ \pi_z \end{bmatrix} \qquad \text{if } \pi_t \neq 0 \tag{2.31}$$

In that sense, dual-space geometry is not a new concept in computational geometry. Originally, the dual of a given vector space (E) is defined as the set of linear forms on (E) (linear functions of (E) into the reals $\mathbb{R}$). See [12]. In the case where (E) is the three dimensional Euclidean space, each linear form may be interpreted as a plane Π in space that is typically parameterized by a homogeneous 4-vector $\overline{\pi} \simeq [\pi_x \ \ \pi_y \ \ \pi_z \ \ \pi_t]^T$. A point P of homogeneous coordinates $\overline{\mathbf{X}} = [X \ \ Y \ \ Z \ \ 1]^T$ lies on a generic plane Π of coordinates $\overline{\pi}$ if and only if $\langle \overline{\pi}, \overline{\mathbf{X}} \rangle = 0$ (see [13]). Our contribution is mainly the new $\overline{\omega}-$parameterization. We will show that this representation exhibits useful properties allowing us to naturally relate objects in Euclidean space (planes, lines and points) to their perspective projections on the image plane (lines and points). One clear limitation of that representation is that plane crossing the camera origin cannot be parameterized using that formalism (for such planes $\pi_t = 0$). However, this will be shown not to be a critical issue in all geometrical problems addressed in this thesis (as most planes of interest do not contain the camera center).

2.2.2 Properties of B-dual-space

This section presents the fundamental properties attached to dual-space geometry.

The following proposition constitutes the major property associated to our choice of parameterization:

Proposition 1: Consider two planes Π_a and Π_b in space, with respective coordinate vectors $\overline{\omega}_a$ and $\overline{\omega}_b$ $(\overline{\omega}_a \neq \overline{\omega}_b)$ in dual-space, and let $\Lambda = \Pi_a \cap \Pi_b$ be the line of intersection between them. Let λ be the perspective projection of Λ on the image plane, and $\overline{\lambda}$ its homogeneous coordinate vector. Then $\overline{\lambda}$ is parallel to $\overline{\omega}_a - \overline{\omega}_b$ (see figure 2.6). In other words, $\overline{\omega}_a - \overline{\omega}_b$ is a valid coordinate vector of the line λ.

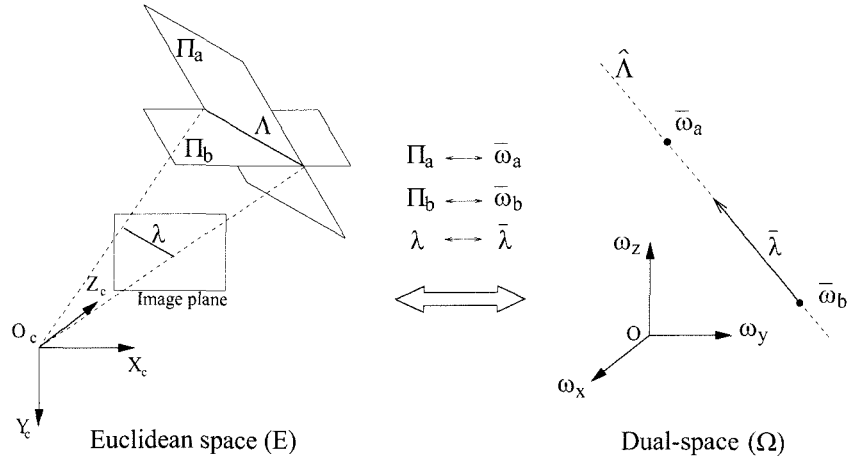


Figure 2.6: Proposition 1 - Intersecting planes: The direction of the line connecting two planes vectors $\overline{\omega}_a$ and $\overline{\omega}_b$ in plane space (Ω) is precisely $\overline{\lambda}$, the coordinate vector of the perspective projection λ of the line of intersection Λ between the two planes Π_a and Π_b in Euclidean space (E).

Proof: Let $P \in \Lambda$ and let p be the projection of P on the image plane. Call $\overline{X} = [X \;\; Y \;\; Z]^T$ and $\overline{\mathbf{x}} \simeq \frac{1}{Z}\overline{X}$ the respective coordinates of P and p. We successively have:

$$
\begin{aligned}
P \in \Lambda \iff & \begin{cases} P \in \Pi_a \\ P \in \Pi_b \end{cases} \\
\iff & \begin{cases} \langle \overline{\omega}_a, \overline{X} \rangle = 1 \\ \langle \overline{\omega}_b, \overline{X} \rangle = 1 \end{cases} \\
\implies & \langle \overline{\omega}_a - \overline{\omega}_b, \overline{X} \rangle = 0 \\
\implies & \langle \overline{\omega}_a - \overline{\omega}_b, \overline{\mathbf{x}} \rangle = 0 \qquad (\text{since } Z \neq 0) \\
\implies & \overline{\lambda} \simeq \overline{\omega}_a - \overline{\omega}_b. \; \blacksquare
\end{aligned}
$$

Notice that this relation is significantly simpler than that derived using standard projective geometry (equation 2.12).

In addition, observe that the coordinate vector $\overline{\omega}$ of any plane Π containing the line Λ lies on the line connecting $\overline{\omega}_a$ and $\overline{\omega}_b$ in dual-space (Ω). We denote that line by $\hat{\Lambda}$ and call it the *dual image* of Λ. The following definition generalizes that concept of dual image to other geometrical objects:

Definition: Let $\mathcal{A}$ be a sub-manifold of (E) (e.g., a point, line, plane, surface or

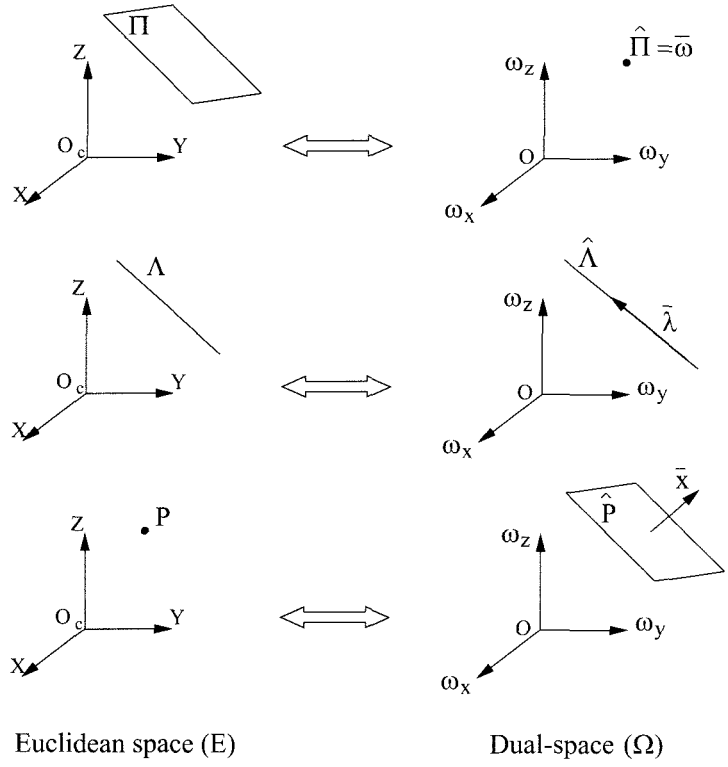


Figure 2.7: Duality principle: The dual images of a plane Π, a line Λ and a point P are respectively a point, a line and a plane. Notice that the perspective projection $\overline{\lambda}$ of the line Λ is directly observable in dual-space as the direction vector of its dual image $\hat{\Lambda}$. Similarly, the normal vector of the plane $\hat{P}$ (dual image of P) is precisely the homogeneous coordinate vector $\overline{\mathbf{x}}$ of the projection of P on the image plane. If P is a point at infinity (vanishing point), then its dual image $\hat{P}$ is a plane containing the origin O of the plane space reference frame.

curve). The *dual image* $\hat{\mathcal{A}}$ of $\mathcal{A}$ is defined as the set of coordinates vectors $\overline{\omega}$ in dual-space (Ω) representing the tangent planes to $\mathcal{A}$. Following that standard definition (see [13, 14]), the dual images of points, lines and planes in (E) may be shown to be respectively planes, lines and points in dual-space (Ω), as illustrated in figure 2.7. Further properties regarding non-linear sub-manifolds may be observed, such as for quadric surfaces in [15] or for general apparent contours in space in [16].

The following five propositions cover the principal properties attached to the dual-space formalism.

Proposition 2 - Parallel Planes - Horizon Line: Let Π_a and Π_b be two parallel planes of coordinates $\overline{\omega}_a$ and $\overline{\omega}_b$. Then $\overline{\omega}_a$ is parallel to $\overline{\omega_b}$.

Proof: The planes have the same surface normals $\overline{n}_a = \overline{n}_b$. Therefore, the propo-

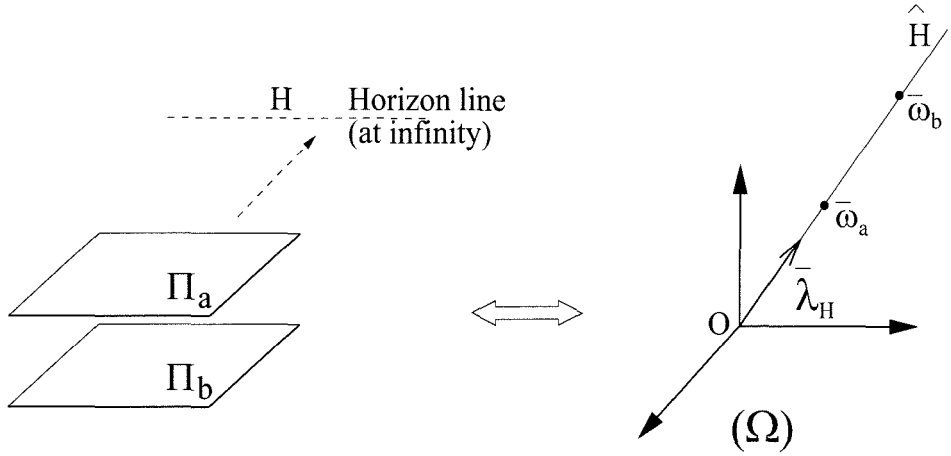


Figure 2.8: Proposition 2 - Parallel planes - Horizon line: The projection of the horizon line $\overline{\lambda}_H$ is precisely the orientation of the planes Π_a and Π_b.

sition follows from definition of $\overline{\omega}$.

The horizon line H represents the "intersection" of two planes at infinity. The dual image of H is the line $\hat{H}$ connecting $\overline{\omega}_a$ and $\overline{\omega}_b$ and crossing the origin of the (Ω) space. The direction of that line is not only the normal vector of the two planes $\overline{n}_a = \overline{n}_b$, but also the representative vector $\overline{\lambda}_H$ of the projection λ_H of H (horizon line) on the image plane (according to proposition 1). Although H is not a well-defined line in Euclidean space (being a line at infinity), under perspective projection, it may give rise to a perfectly well defined line λ_H on the image plane (for example a picture of the ocean). Once that line is extracted, the orientation of the plane is known:

$$\overline{\omega}_a \simeq \overline{\omega}_b \simeq \overline{\lambda}_H \tag{2.32}$$

Figure 2.8 gives a geometrical illustration of that proposition.

Proposition 3 - Orthogonal Planes: If two planes Π_a and Π_b are two orthogonal, then so are their coordinate vectors $\overline{\omega}_a$ and $\overline{\omega}_b$ in dual-space. Consequently, once one of the plane $\overline{\omega}_a$ is known, then $\overline{\omega}_b$ is constrained to lie in the sub-space orthogonal to $\overline{\omega}_a$, a plane in dual-space.

Proposition 4 - Intersecting lines: Consider two lines Λ_a and Λ_b intersecting at a point P, and call Π the plane that contains them. In dual-space, the two dual

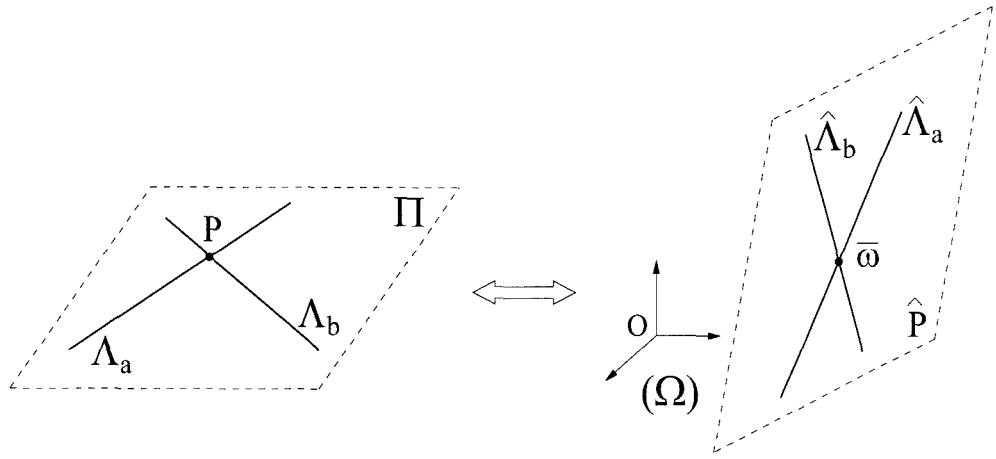


Figure 2.9: Proposition 4 - Intersecting lines: The dual-images of two intersecting lines Λ_a and Λ_b (defining the plane Π) are two lines $\hat{\Lambda}_a$ and $\hat{\Lambda}_b$ in (Ω) that intersect at $\overline{\omega}$ the coordinate vector of Π.

lines $\hat{\Lambda}_a$ and $\hat{\Lambda}_b$ necessarily intersect at $\overline{\omega}$ the coordinate vector of Π (since $\overline{\omega}$ is the plane that contains both lines). Similarly, the dual image $\hat{P}$ of P is the plane in dual-space that contains both dual lines $\hat{\Lambda}_a$ and $\hat{\Lambda}_b$. Notice that $\hat{P}$ does not cross the origin of (Ω). See figure 2.9.

Proposition 5 - Parallel lines - Vanishing Point: Consider two parallel lines Λ_a and Λ_b belonging to the plane Π of coordinates $\overline{\omega}$. Then $\overline{\omega}$ is at the intersection of the two dual lines $\hat{\Lambda}_a$ and $\hat{\Lambda}_b$. In dual-space, the plane containing both dual lines $\hat{\Lambda}_a$ and $\hat{\Lambda}_b$ is the dual image $\hat{V}$ of the *vanishing point* V, i.e., the intersection point of Λ_a and Λ_b in Euclidean space. If H is the horizon line associated with Π, then $V \in H$, which translates in dual-space into $\hat{H} \subset \hat{V}$. Since $\hat{H}$ contains the origin, so does $\hat{V}$. Notice that once the perspective projection v of V is observable on the image plane, the plane $\hat{V}$ is entirely known (since its orientation is the coordinate vector of v). See figure 2.10.

Proposition 6 - Orthogonal lines: Let Λ_1 and Λ_2 be two orthogonal lines contained in the plane Π of coordinates $\overline{\omega}$) and let $\overline{\omega} = \hat{\Lambda}_1 \cap \hat{\Lambda}_2$. Consider the set of planes orthogonal to Π. In the dual-space, that set is represented by a plane containing the origin, and orthogonal to $\overline{\omega}$ (see proposition 3). Call that plane $\hat{V}$ (it can be shown to be the dual image of a vanishing point). In that set, consider the two specific planes Π_1 and Π_2 that contain the lines Λ_1 and Λ_2 (see figure 2.11). In the

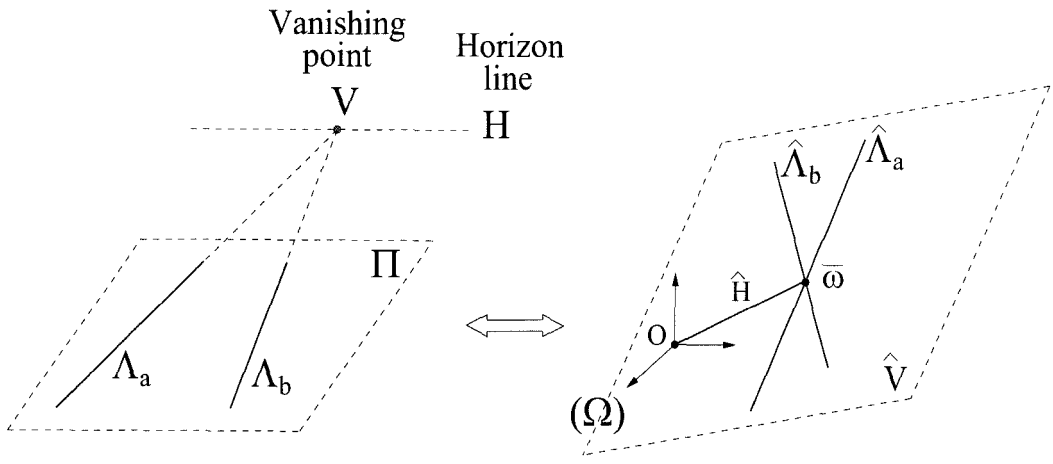


Figure 2.10: Proposition 5 - Parallel lines - Vanishing point

dual-space, the representative vectors $\overline{\omega}_1$ and $\overline{\omega}_2$ of those two planes are defined as the respective intersections between $\hat{V}$ and the two lines $\hat{\Lambda}_1$ and $\hat{\Lambda}_2$. Then, since the two lines Λ_1 and Λ_2 are orthogonal, the two vectors $\overline{\omega}_1$ and $\overline{\omega}_2$ are also orthogonal. See figure 2.11-top. This implies that the images of the two vanishing points $\hat{V}_1$ and $\hat{V}_2$ associated to the lines Λ_1 and Λ_2 are orthogonal in the dual-space. See figure 2.11-bottom.

Two vanishing points are enough to recover the horizon line H associated with a given plane Π in space. Therefore, observing two sets of parallel lines belonging to the same plane under perspective projection allows us to recover the horizon line, and therefore the orientation of the plane in space (from proposition 2). This process is illustrated in figure 2.12: The horizon line H corresponding to the ground floor Π is recovered from the two vanishing points V_1 and V_2.

2.2.3 Geometrical problems solved in B-dual-space

This section presents several useful geometrical problems solved using dual-space geometry.

Example 4: Let Π be a plane in space of coordinate vector $\overline{\omega}$. Let P be a point on Π with coordinate vector $\overline{X} = [X \quad Y \quad Z]^T$. Let p be the projection of P onto the image plane, and denote $\overline{\mathbf{x}} \simeq [x \quad y \quad 1]^T$ its homogeneous coordinate vector. The

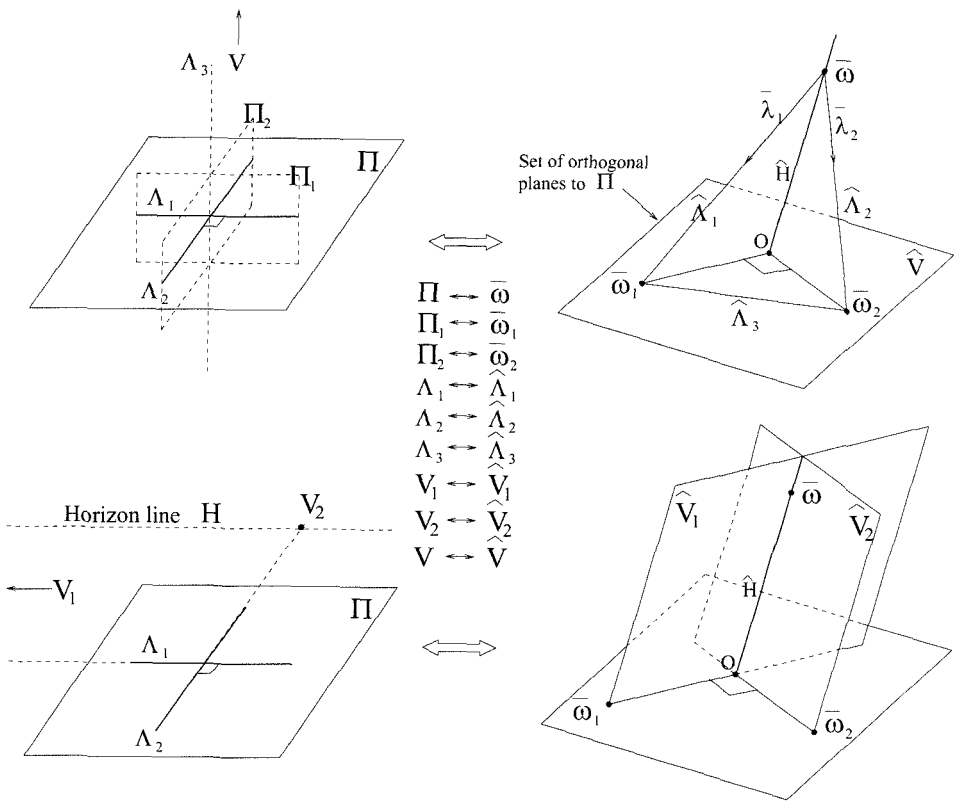


Figure 2.11: Proposition 6 - Orthogonal lines: Two lines Λ_1 and Λ_2 are orthogonal if and only if their corresponding vanishing points' images $\hat{V}_1$ and $\hat{V}_2$ are orthogonal in dual-space.

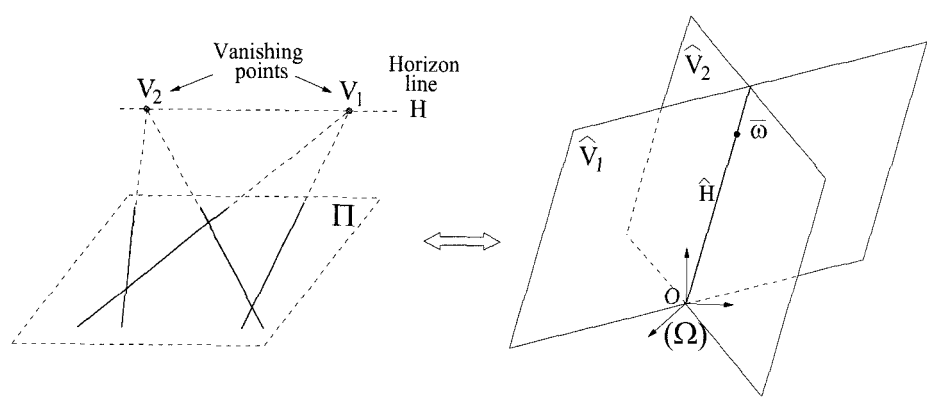


Figure 2.12: Recovery of the horizon line (plane orientation): Two sets of parallel lines lying on a given plane Π provide two vanishing points V_1 and V_2. The line connecting them is the horizon line H attached to the plane. In dual-space, $\hat{H} = \hat{V}_1 \cap \hat{V}_2$. Notice that if the two sets of lines are orthogonal then the two planes $\hat{V}_1$ and $\hat{V}_2$ must be orthogonal in the reciprocal space, or equivalently the coordinate vectors of the projections of the vanishing points are mutually orthogonal (see figure 2.11).

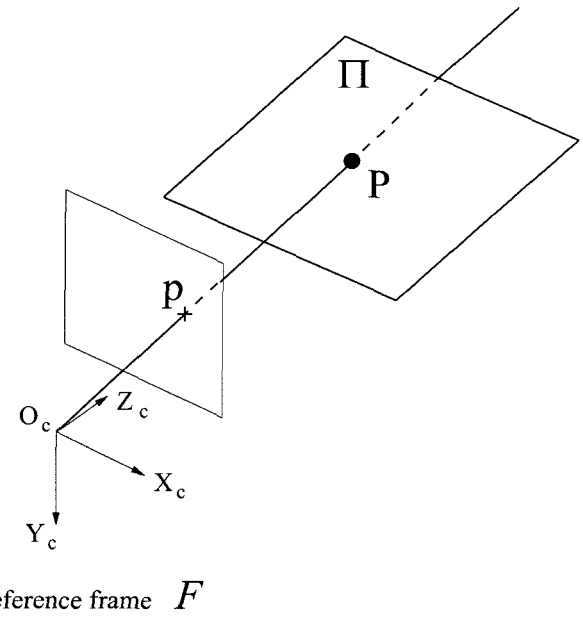


Figure 2.13: The triangulation problem consists of finding P from its image projection p and the plane Π.

triangulation problem consists of finding the point P from its projection p and the plane Π, or calculating $\overline{X}$ from $\overline{\mathbf{x}}$ and $\overline{\omega}$. See figure 2.13. Since P lies on the optical ray (O_c, p), its coordinate vector satisfies $\overline{X} \simeq \overline{\mathbf{x}}$, or equivalently, $\overline{X} = Z\,\overline{x}$, with $\overline{x} = [x \quad y \quad 1]^T$. In addition, since P lies on the plane Π, we have $\langle \overline{\omega}, \overline{X} \rangle = 1$. This implies:

$$Z = \frac{1}{\langle \overline{\omega}, \overline{x} \rangle} \qquad \Longrightarrow \qquad \overline{X} = \frac{\overline{x}}{\langle \overline{\omega}, \overline{x} \rangle} \tag{2.33}$$

This is the fundamental triangulation equation between a ray and a plane in space.

Example 5: Consider two camera frames $\mathcal{F}$ and $\mathcal{F}'$ and let $\{R, T\}$ be the rigid motion parameters between $\mathcal{F}$ and $\mathcal{F}'$ (defined in figure 2.4). Let Π be a plane in space of coordinate vectors $\overline{\omega}$ and $\overline{\omega}'$ in $\mathcal{F}$ and $\mathcal{F}'$ respectively. How do $\overline{\omega}$ and $\overline{\omega}'$ relate to each other?

Consider a generic point P on Π of coordinate vectors $\overline{X}$ and $\overline{X}'$ in $\mathcal{F}$ and $\mathcal{F}'$

respectively. Then, $\overline{X}' = R\overline{X} + T$. Since $P \in \Pi$, we may write:

$$\left\langle \overline{\omega}', \overline{X}' \right\rangle = 1 \tag{2.34}$$

$$\left\langle \overline{\omega}', R\overline{X} + T \right\rangle = 1 \tag{2.35}$$

$$\left\langle R^T \overline{\omega}', \overline{X} \right\rangle = 1 - \left\langle \overline{\omega}', T \right\rangle \tag{2.36}$$

$$\left\langle \frac{R^T \overline{\omega}'}{1 - \left\langle \overline{\omega}', T \right\rangle}, \overline{X} \right\rangle = 1 \qquad \text{if } \left\langle \overline{\omega}', T \right\rangle \neq 1 \tag{2.37}$$

Therefore:

$$\overline{\omega} = \frac{R^T \overline{\omega}'}{1 - \left\langle \overline{\omega}', T \right\rangle}. \tag{2.38}$$

This expression is the equivalent of equation 2.27 in dual-space geometry. Notice that the condition $\langle \overline{\omega}', T \rangle \neq 1$ is equivalent to enforcing the plane Π not to contain the origin of the first camera reference frame $\mathcal{F}$. That is a necessary condition in order to have a well defined plane vector $\overline{\omega}$. The inverse expression may also be derived in a similar way:

$$\overline{\omega}' = \frac{R\overline{\omega}}{1 + \left\langle \overline{\omega}, R^T T \right\rangle}. \tag{2.39}$$

In that case, the condition $\langle \overline{\omega}, -R^T T \rangle \neq 1$ constrains the plane Π not to contain the origin of the second reference frame $\mathcal{F}'$ (in order to have a well defined vector $\overline{\omega}'$).

In some cases, only one of the two plane vectors $\overline{\omega}$ or $\overline{\omega}'$ is well-defined. The following example is one illustration of such a phenomenon.

Example 6: Consider the geometrical scenario of example 5 where the plane Π is now spanned by a line λ' on the image plane of the second camera reference frame $\mathcal{F}'$ (after motion). This case is illustrated in figure 2.5. In that case, the coordinate vector $\overline{\omega}'$ is not well defined (since by construction, the plane Π contains the origin of $\mathcal{F}'$). However, the plane vector $\overline{\omega}$ may very well be defined since Π does not necessarily contain the origin of the first reference frame $\mathcal{F}$. Indeed, according to equation 2.29,

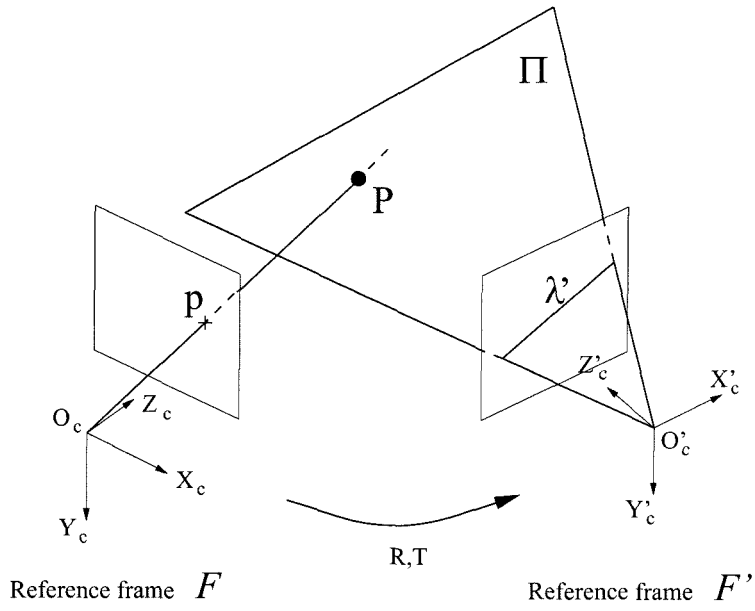


Figure 2.14: Triangulation between the optical ray (O_c, p) and the plane Π spanned by λ'.

the homogeneous coordinate vector $\overline{\pi}$ of Π in $\mathcal{F}$ is given by:

$$\overline{\pi} \simeq \begin{bmatrix} R^T \overline{\lambda}' \\ \langle T, \overline{\lambda}' \rangle \end{bmatrix} \tag{2.40}$$

where $\overline{\lambda}'$ is the homogeneous coordinate vector of the image line λ' in $\mathcal{F}'$. Then, according to expression 2.31, the corresponding dual-space vector $\overline{\omega}$ is given by:

$$\overline{\omega} = -\frac{R^T \overline{\lambda}'}{\langle T, \overline{\lambda}' \rangle} \tag{2.41}$$

which is perfectly well-defined as long as Π does not contain O_c, or equivalently if $\langle T, \overline{\lambda}' \rangle \neq 0$ [†].

Example 7: Figure 2.14 illustrates a mix of examples 4 and 6. In that case, the problem consists of triangulating the optical ray (O_c, p) with the plane Π spanned by the line λ' in the other camera reference frame $\mathcal{F}'$. Let $\overline{X} = [X \;\; Y \;\; Z]^T$ be the

[†]This condition is equivalent to enforcing the line λ' not to contain the epipole $e' \simeq T$ on the image plane attached to camera frame $\mathcal{F}'$. The point e' is the projection of O_c onto the image plane attached to the second camera reference frame $\mathcal{F}'$ (see figure 4.1).

coordinates of P in space and $\overline{x} = [x \;\; y \;\; 1]^T$ the coordinates of its known projection p on the image plane. Equation 2.41 provides then an expression for the coordinate vector $\overline{\omega}$ of Π in frame $\mathcal{F}$:

$$\overline{\omega} = -\frac{R^T \overline{\lambda}'}{\langle T, \overline{\lambda}' \rangle} \tag{2.42}$$

where $\overline{\lambda}'$ is the homogeneous coordinate vector of the image line λ' in $\mathcal{F}'$. The triangulation expression given by equation 2.40 returns then the final coordinate vector of P:

$$\overline{X} = \frac{\overline{x}}{\langle \overline{\omega}, \overline{x} \rangle} = -\frac{\langle T, \overline{\lambda}' \rangle \, \overline{x}}{\langle R^T \overline{\lambda}', \overline{x} \rangle} \tag{2.43}$$

Observe that the plane Π is not allowed to cross the origin of the initial reference frame $\mathcal{F}$, otherwise, triangulation is impossible. Therefore the plane vector $\overline{\omega}$ is perfectly well defined (i.e., $\langle T, \overline{\lambda}' \rangle \neq 0$).

Chapter 3 Camera Calibration in B-dual-space geometry

In this chapter, we propose to apply the B-dual-space formalism to the problem of camera calibration. In section 3.1, the problem of camera calibration is first defined, followed in section 3.2 by the complete derivation of closed-form solutions for intrinsic and extrinsic camera parameters using dual-space geometry as fundamental mathematical tool. Section 3.3 closes the chapter with some conclusions.

3.1 Definition of camera calibration

3.1.1 Pixel coordinates - intrinsic camera parameters

The position of a point p in a real image is originally expressed in pixel units. One can only say that a point p is at the intersection of column $p_x = 150$ and row $p_y = 50$ on a given digitized image. So far, we have been denoting $\overline{x} = [x \;\; y \;\; 1]^T$ the homogeneous coordinate vector of a generic point p on the image plane. This vector (also called *normalized coordinate vector*) is directly related to the 3D coordinates $\overline{X} = [X \;\; Y \;\; Z]^T$ of the corresponding point P is space through the perspective projection operator (eq. 2.2). Since in practice we only have access to pixel coordinates $\overline{p} = [p_x \;\; p_y \;\; 1]^T$, we need to establish a correspondence between $\overline{p}$ and $\overline{x}$ (from pixel coordinates to optical ray in space).

Since the origin of the image reference frame is at the optical center c (or principal point), it is necessary to know the location of that point in the image: $\overline{c} = [c_x \;\; c_y]^T$ (in pixels). Let f_o be the focal distance (in meters) of the camera optics (distance of the lens focal point to the imaging sensor), and denote by d_x and d_y the x and y dimensions of the pixels in the imaging sensor (in meters). Let $f_x = f_o/d_x$ and

$f_y = f_o/d_y$ (in pixels). Notice that for most imaging sensors currently manufactured, pixels may be assumed perfectly square, implying $d_x = d_y$ or equivalently $f_x = f_y$. In the general case f_x and f_y may be different.

Then, the pixel coordinates $\overline{p} = [p_x \quad p_y \quad 1]^T$ of a point on the image may be computed from its normalized homogeneous coordinates $\overline{x} = [x \quad y \quad 1]^T$ through the following expression:

$$\begin{cases} p_x &= f_x\, x + c_x \\ p_y &= f_y\, y + c_y \end{cases} \tag{3.1}$$

That model assumes that the two axes of the imaging sensor are orthogonal. In the case where they are not orthogonal, the pixel mapping function may be generalized to:

$$\begin{cases} p_x &= f_x\, x - \alpha\, f_y\, y + c_x \\ p_y &= f_y\, y + c_y \end{cases} \tag{3.2}$$

where α is a scalar coefficient that controls the amount of skew between the two main sensor axes (if $\alpha = 0$, there is no skew). For now, let us consider the simple model without skew (equation 3.1). If p is the image projection of the point P in space (of coordinates $\overline{X} = [X \quad Y \quad Z]^T$), the global projection map may be written in pixel units:

$$\begin{cases} p_x &= f_x\, (X/Z) + c_x \\ p_y &= f_y\, (Y/Z) + c_y \end{cases} \tag{3.3}$$

This equation returns the coordinates of a point projected onto the image (in pixels) in the case of an ideal pinhole camera. Real cameras do not have pinholes, but lenses. Unfortunately a lens will introduce some amount of distortion (also called aberration) in the image. That makes the projected point to appear at a slightly different position on the image. The following expression is a simple first-order model

that captures the distortions introduced by the lens:

$$\begin{cases} \overline{a} = \begin{bmatrix} X/Z \\ Y/Z \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} & \text{pinhole projection} \\ \overline{b} = \begin{bmatrix} b_x \\ b_y \end{bmatrix} = \overline{x}(1 + k_c\|\overline{a}\|^2) & \text{radial distortion} \\ \overline{p} = \begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} f_x b_x + c_x \\ f_y b_y + c_y \end{bmatrix} & \text{pixel coordinates} \end{cases} \tag{3.4}$$

where k_c is called the radial distortion factor. This model is also called first-order symmetric radial distortion model [17, 18, 19] ("symmetric" because the amount of distortion is directly related to the distance of the point to the optical center c). Observe that the systems (3.4) and (3.3) are equivalent when $k_c = 0$ (no distortion).

Therefore, if the position of the point P is known in *camera reference frame*, one may calculate its projection onto the image plane given the intrinsic camera parameters f_x, f_y, c_x, c_y and k_c. That is known as the *direct projection operation* and may be denoted $\overline{p} = \Pi(\overline{X})$. However, most 3D vision applications require to solve the "inverse problem" that is mapping pixel coordinates $\overline{p}$ to 3D world coordinates $[X \quad Y \quad Z]^T$. In particular, one necessary step is to compute normalized image coordinates $\overline{x} = [x \quad y \quad 1]^T$ (3D ray direction) from pixel coordinates $\overline{p}$ (refer to equation 3.4). The only non-trivial aspect of that inverse map computation is in computing the vector $\overline{a}$ from $\overline{b}$. This is the distortion compensation step. It may be shown that for relatively small distortions, this inverse map may be very well approximated by the following equation:

$$\overline{a} \approx \frac{\overline{b}}{1 + k_c \left\| \frac{\overline{b}}{1+k_c\|\overline{b}\|^2} \right\|^2} \tag{3.5}$$

Experimentally, this expression is sufficiently accurate.

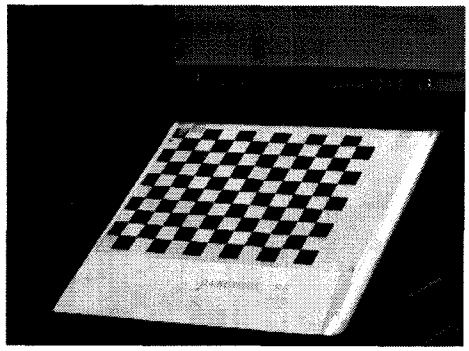
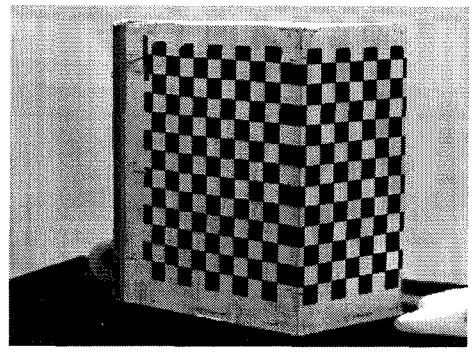
(a)  (b) 

Figure 3.1: Two examples of camera calibration image: (a) with a planar calibration rig (checker board pattern) or (b) a 3D calibration rig (a corner).

3.1.2 Camera calibration

The camera calibration procedure consists of identifying the intrinsic camera parameters f_x, f_y, c_x, c_y and k_c (and possibly α). A standard method is to acquire an image a known 3D object (a checker board pattern, a box with known geometry...) and look for the set of parameters that best match the computed projection of the structure with the observed projection on the image. The reference object is also called *calibration rig.* Since the camera parameters are inferred from image measurements, this approach is also called *visual calibration.* This technique was originally presented by Tsai in [18, 19] and Brown in [17]. An algorithm for estimation was proposed by Abdel-Aziz and Karara in [20] (for an overview on camera calibration, the reader may also refer to the book Klette, Schluns and Koschan [9]).

Figure 3.1 shows two examples of calibration images when using a planar rig (checker board pattern) and a 3D rig (two orthogonal checker board patterns).

Note that although the geometry of the calibration rig is known (i.e., the mutual position of the grid corners in space), its absolute location with respect to the camera is unknown. In other words, the *pose* of the calibration pattern is unknown. Therefore, before applying the set of equations (3.4) to compute the image projection of every corner in the structure, one needs to find their 3D coordinates in the camera reference frame. We first choose a reference frame attached to the rig (called the object frame) in which we express the known coordinates $\overline{X}_o^i$ of all the corners P_i, $(i = 1 \ldots N)$. This set of vectors is known since the intrinsic rig structure is known. Then, the

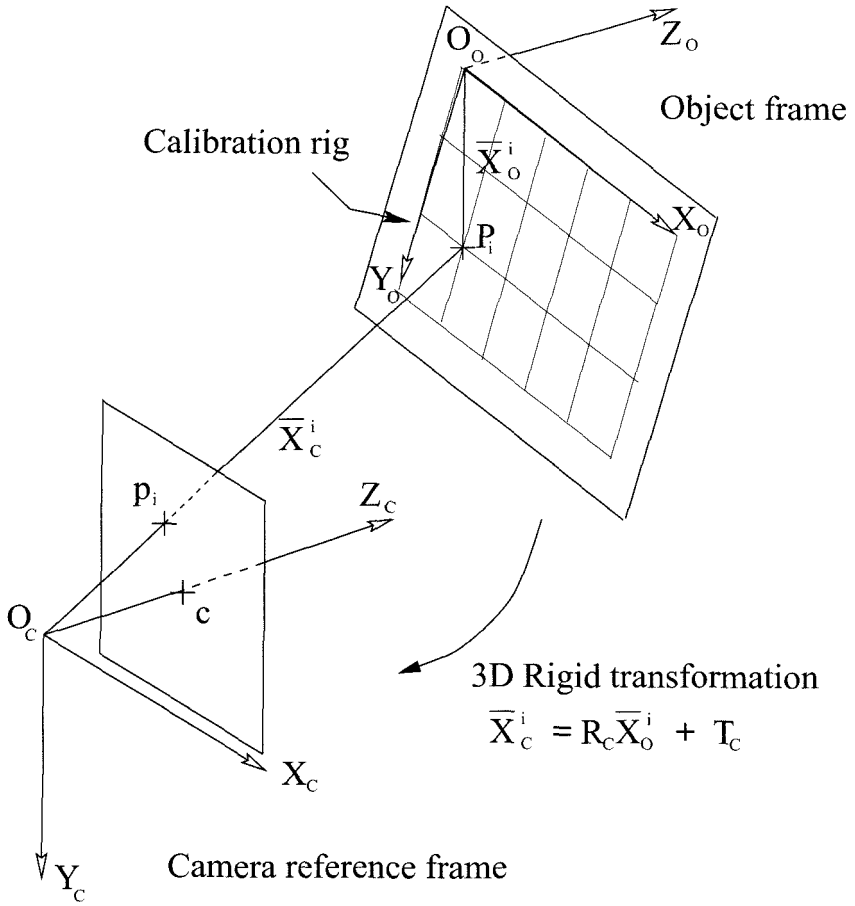


Figure 3.2: Camera calibration system. This figure illustrates the case where a planar rig is used for calibration. In general, any 3D structure may be used such as a box or a corner.

coordinate vector $\overline{X}^i_c$ of P_i in the camera frame is related to $\overline{X}^i_o$ through a rigid motion transformation:

$$\forall i = 1, \ldots, N, \qquad \overline{X}^i_c = R_c \overline{X}^i_o + T_c \tag{3.6}$$

where R_c and T_c define the pose of the calibration rig with respect to the camera (similarly to equation 2.13). See figure 3.2.

Notice that by adding the calibration object in the scene, more unknowns have been added to the problem: R_c and T_c. Those parameters are called *extrinsic camera parameters* since they are dependent upon the pose of the calibration pattern with respect to the camera (unlike the intrinsic parameters that remain constant as the

rig is moved in front of the camera).

Let $\overline{\Omega}_c$ be the rotation vector associated to the rotation matrix R_c (see equation 2.13). Then, the complete set of unknowns to solve for is:

- Focal length: f_x, f_y (2 DOF),
- Principal point coordinates: c_x, c_y (2 DOF),
- Radial distortion factor: k_c (1 DOF),
- Calibration rig pose: $\overline{\Omega}_c$, T_c (6 DOF).

Therefore, the global calibration problem consists of solving for a total of 11 scalar parameters (adding the skew coefficient α would bring the number of unknowns to 12).

Let p_i $(i = 1, \ldots, N)$ be the observed image projections of the rig points P_i and let $\overline{p}_i = [p_x^i \;\; p_y^i]^T$ be their respective pixel coordinates (see figure 3.2). Experimentally, the points p_i are detected using the standard Harris corner finders [21].

The estimation process consists then of finding the set of calibration unknowns (extrinsic and intrinsic) that minimizes the reprojection error. Therefore, the solution to that problem may be written as follows:

$$\{f_x, f_y, c_x, c_y, k_c, \overline{\Omega}_c, T_c\} = \text{Argmin} \sum_{i=1}^{N} \left\| \overline{p}_i - \Pi(R_c \overline{X}_o^i + T_c) \right\|^2 \tag{3.7}$$

where $R_c = e^{\overline{\Omega}_c \wedge}$, $\Pi(.)$ is the image projection operator defined in equation 3.4 (function of the intrinsic parameters f_x, f_y, c_x, c_y and k_c) and $\|.\|$ is the standard distance norm is pixel units. This non-linear optimization problem may be solved using standard gradient descent techniques. However, it is required to have a good initial guess before starting the iterative refinement. The purpose of the next section 3.2 is to present a method to derive closed form expressions for calibration parameters that may be used for initialization.

Apart from numerical implementation details, it is also important to study the observability of the model. In other words, under which conditions (type of the

calibration rig and its position in space) can the full camera model (eq. 3.4) be estimated from a single image projection. For example, it is worth noticing that if the calibration rig is planar (as shown on figure 3.1-a) the optical center c cannot be estimated (the two coordinates c_x and c_y). Therefore, in such cases, it is necessary to reduce the camera model to fewer intrinsic parameters and fix the optical center in the center of the image. Further discussions on the camera model observability may be found in the next section 3.2.

3.2 Closed-form solution in B-dual-space geometry

This section demonstrates how one may easily retrieve closed-form expressions for intrinsic and extrinsic camera parameters using the dual-space formalism as a fundamental mathematical tool. The method is based on using vanishing points and vanishing lines. The concept of using vanishing points for camera calibration is not new (most of the related work on this topic may probably be found in references [22, 23, 24, 25, 26, 27, 28, 29]). Therefore, the ambition of this work is not to state new concepts or theories on calibration, but rather illustrate the convenience of the dual-space formalism by applying it to the problem of calibration. We show here that this formalism enables us to keep the algebra simple and compact while exploiting all the geometrical constraints present in the scene (in the calibration rig). That will also lead us to derive properties regarding the observability of several camera models under different geometrical configurations of the setup, and types of calibration rig used (2D or 3D). Most related work on that topic only deal with simple camera model (unique focal length) [22, 26] and extract the extrinsic parameters through complex 3D parameterization (using Euler angles) [26, 28, 29]. Other standard methods for deriving explicit solutions for camera calibration were presented by Abdel-Aziz and Karara [20] and Tsai [18]. These methods are based on estimating, in a semi-linear way, a set of parameters that is larger than the real original set of unknowns and do

not explicitly make use of all geometrical properties of the calibration rig. For an overview of those techniques, refer to [9].

The method that we propose here involves very compact algebra, uses intuitive and minimal parameterizations, and naturally allows to exploit all geometrical properties present in the observed three-dimensional scene (calibration rig). In addition, our approach may be directly applied to natural images that do not contain a special calibration grid (such as pictures of buildings, walls, furniture...).

Once it is computed, the closed-form solution is then fed to the non-linear iterative optimizer as an initial guess for the calibration parameters (see previous section 3.1). This final optimization algorithm is inspired from the method originally presented by Tsai in [18, 19] including lens distortion (see equation 3.7). The purpose of that analysis is to provide a good initial guess to the non-linear optimizer, to better insure convergence, and check for the consistency of the results.

We will first consider the case of a calibration when using a planar rig (a 2D grid), and then generalize the results to 3D rigs (such as a cube). In those two cases, different camera models will be used.

3.2.1 When using a planar calibration rig

Consider the calibration image shown in figure 3.1-a. Assuming no lens distortion ($k_c = 0$) and no image noise, the grid may be summarized by its four extreme corners on the image (intuitively, one may localize all the inside grid corners from those four points by simple perspective warping). In practice, all points will be used in order to be less sensitive to image noise, however the principle remains the same. Then, the basic observed pattern is a perspective view of a rectangle of known dimensions $L \times W$. Without loss of generality, we can also assume that this rectangle is a square. The reason for that is that through a similar perspective image warping, it is always possible to convert a perspective view of a rectangle into a perspective view of a square, given that the dimensions of the original rectangle are known (actually, only the ratio W/L is necessary).

Figure 3.3 shows a perspective image of a square $ABCD$. The four points $\overline{x}_1$, $\overline{x}_2$, $\overline{x}_3$ and $\overline{x}_4$ are the coordinate vectors of the detected corners of the square on the image plane *after normalization*. This means that the $\overline{x}$ vectors are computed from the pixel coordinates of the points after subtraction of the optical center coordinates (c_x, c_y) (in pixel) and scaling by the inverse of the focal length (in pixel as well). To model the aspect ratio in x and y, one can assume two distinct focal lengths f_x and f_y in both image directions (to account for non-square CCD pixels).

In the case of calibration from planar rigs, it is known that the optical center position (c_x, c_y) cannot be estimated (see [18, 19, 29]). Therefore, we will keep it fixed at the center of the image, and take it out of the set of unknowns. The resulting intrinsic parameters to be estimated are therefore f_x and f_y. Let $\overline{p}_i \simeq [p_{x_i} \;\; p_{y_i} \;\; 1]^T$ $(i = 1, \ldots, 4)$ be the pixel locations of the corners after subtraction of the optical center (in homogeneous coordinates). Then one can extract the $\overline{x}$ vectors through a linear operation involving the focal lengths f_x and f_y: for $i = 1, \ldots, 4$,

$$\overline{x}_i \simeq \begin{bmatrix} 1/f_x & 0 & 0 \\ 0 & 1/f_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \overline{p}_i = K\,\overline{p}_i \tag{3.8}$$

where K is the intrinsic camera matrix containing the intrinsic parameters (f_x and f_y). Let us now extract the set of independent constraints attached to the observation in order to estimate the focal lengths (hence the camera matrix K).

Figure 3.3 shows the set of corner points $\overline{x}_i$ on the image plane. Following proposition 5 of section 2.2.2, the lines $\overline{\lambda}_i$ $(i = 1, \ldots, 4)$ are used to infer the two vanishing points V_1 and V_2 in order to recover the projection $\overline{\lambda}_H$ of the horizon line H associated to the plane Π_d. The derivation is as follows:

$$\left.\begin{array}{l} \left.\begin{array}{l} \overline{\lambda}_1 \simeq \overline{x}_1 \times \overline{x}_2 \\ \overline{\lambda}_2 \simeq \overline{x}_3 \times \overline{x}_4 \end{array}\right\} V_1 \simeq \overline{\lambda}_1 \times \overline{\lambda}_2 \\ \left.\begin{array}{l} \overline{\lambda}_3 \simeq \overline{x}_2 \times \overline{x}_3 \\ \overline{\lambda}_4 \simeq \overline{x}_4 \times \overline{x}_1 \end{array}\right\} V_2 \simeq \overline{\lambda}_3 \times \overline{\lambda}_4 \end{array}\right\} \overline{\lambda}_H \simeq V_1 \times V_2 \tag{3.9}$$

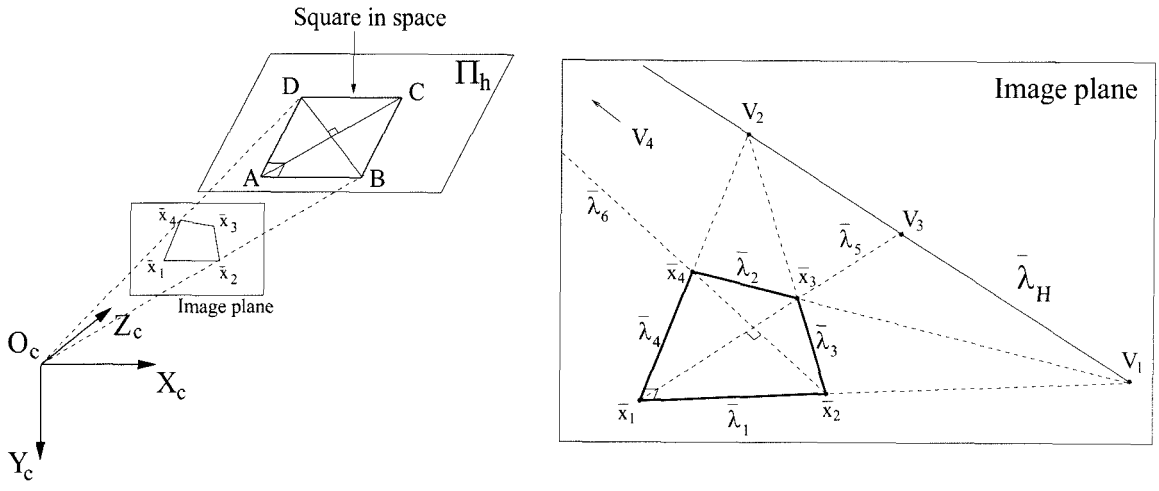


Figure 3.3: Camera calibration using a square (planar) rig: Perspective view of a square in space contained in the plane Π_h. The right figure shows the corners of the square measured on the image plane, together with the four vanishing points $\{V_1, V_2, V_3, V_4\}$ and the horizon line $\bar{\lambda}_H$ associated to Π_h.

where $\times$ is the standard vector product in $\mathbb{R}^3$. Notice that in order to keep the notation clear, we abusively used V_1 and V_2 to refer to the homogeneous coordinates of the vanishing points on the image plane (quantities similar to the $\bar{x}_i$'s using homogeneous coordinates). It is important to keep in mind that all equalities are defined "up to scale." For example, any vector proportional to $\bar{x}_1 \times \bar{x}_2$ would be a good representative for the same line $\bar{\lambda}_1$. The same observation holds for the coordinate vectors of the vanishing points and that of the horizon line.

Yet, the normalized coordinates $\bar{x}_i$ of the corners are not directly available, only the pixel coordinates $\bar{p}_i$. However, all $\bar{x}_i$'s can be retrieved from the $\bar{p}_i$'s through the linear equation 3.8. We will make use of the following statement whose proof may be found in [30]:

Claim 1: Let K be any 3×3 matrix, and $\bar{u}$ and $\bar{v}$ any two 3-vectors. Then the following relation holds:

$$(K\,\bar{u}) \times (K\,\bar{v}) = K^*\,(\bar{u} \times \bar{v})$$

where K^* is the adjoint of K (or the matrix of cofactors of K). Note that if K is invertible (which is the case here), then $K^* = \det(K)\,(K^T)^{-1}$, and consequently

$K^{**} \propto K$.

Using that claim, the camera matrix K (or K^*) may be factored out of the successive vector products of equations 3.9, yielding:

$$\left.\begin{array}{l} \overline{\lambda}_1 \simeq K^* \, \overline{\lambda}_1^p \\ \overline{\lambda}_2 \simeq K^* \, \overline{\lambda}_2^p \end{array}\right\} V_1 \simeq K \, V_1^p \qquad \left.\begin{array}{l} \\ \\ \overline{\lambda}_H \simeq K^* \, \overline{\lambda}_H^p, \\ \\ \end{array}\right\}$$
$$\left.\begin{array}{l} \overline{\lambda}_3 \simeq K^* \, \overline{\lambda}_3^p \\ \overline{\lambda}_4 \simeq K^* \, \overline{\lambda}_4^p \end{array}\right\} V_2 \simeq K \, V_2^p$$

where $\overline{\lambda}_1^p$, $\overline{\lambda}_2^p$, $\overline{\lambda}_3^p$, $\overline{\lambda}_4^p$, V_1^p, V_2^p and $\overline{\lambda}_H^p$ are line and point coordinate vectors on the image plane *in pixel* (directly computed from the pixel coordinates $\overline{p}_1$, $\overline{p}_2$, $\overline{p}_3$ and $\overline{p}_4$):

$$\left.\begin{array}{l} \overline{\lambda}_1^p \simeq \overline{p}_1 \times \overline{p}_2 \\ \overline{\lambda}_2^p \simeq \overline{p}_3 \times \overline{p}_4 \end{array}\right\} V_1^p \simeq \overline{\lambda}_1^p \times \overline{\lambda}_2^p \qquad \left.\begin{array}{l} \\ \\ \overline{\lambda}_H^p \simeq V_1^p \times V_2^p. \\ \\ \end{array}\right\}$$
$$\left.\begin{array}{l} \overline{\lambda}_3^p \simeq \overline{p}_2 \times \overline{p}_3 \\ \overline{\lambda}_4^p \simeq \overline{p}_4 \times \overline{p}_1 \end{array}\right\} V_2^p \simeq \overline{\lambda}_3^p \times \overline{\lambda}_4^p$$

The step of inferring the vanishing points V_1 and V_2 from the pairs of lines $\{\overline{\lambda}_1, \overline{\lambda}_2\}$ and $\{\overline{\lambda}_3, \overline{\lambda}_4\}$ made use of the fact that $ABCD$ is a parallelogram (proposition 5). Using proposition 6 (in section 2.2.2), one naturally enforce orthogonality of the pattern by stating that the two vanishing points V_1 and V_2 are mutually orthogonal (see figure 2.11):

$$\begin{aligned} V_1 \perp V_2 &\iff (K \, V_1^p) \perp (K \, V_2^p) \\ &\iff (V_1^p)^T \left(K^T K\right) (V_2^p) = 0. \end{aligned} \tag{3.10}$$

That provides one scalar constraint in the focal lengths f_x and f_y:

$$\frac{a_1 \, a_2}{f_x^2} + \frac{b_1 \, b_2}{f_y^2} + c_1 \, c_2 = 0 \tag{3.11}$$

where a_1, a_2, b_1, b_2, c_1 and c_2 are the known pixel coordinates of the vanishing points V_1^p and V_2^p: $V_1^p \simeq [a_1 \quad b_1 \quad c_1]^T$ and $V_2^p \simeq [a_2 \quad b_2 \quad c_2]^T$. Notice that equation 3.11

constraints the two square focals (f_x^2, f_y^2) to lie on a fixed hyperbola. Finally, the parallelogram $ABCD$ is not only a rectangle, but also a square. This means that its diagonals (AC) and (BD) are also orthogonal (see figure 3.3). This constraint is exploited by enforcing the two vanishing points attached to the diagonal V_3 and V_4 to be mutually orthogonal (proposition 6). Those points are extracted from intersecting the two diagonal lines $\overline{\lambda}_5$ and $\overline{\lambda}_6$ with the horizon line $\overline{\lambda}_H$ (see figure 3.3). Following the same process of factoring the K matrix (or K^*) out of every successive vector product, one obtains:

$$\begin{aligned} \overline{\lambda}_5 \simeq K^* \overline{\lambda}_5^p \quad &\Longrightarrow \quad V_3 \simeq K\, V_3^p \\ \overline{\lambda}_6 \simeq K^* \overline{\lambda}_6^p \quad &\Longrightarrow \quad V_4 \simeq K\, V_4^p \end{aligned}$$

where V_3^p and V_4^p are the two pixel coordinates of the vanishing points V_3 and V_4 (pre-computed from the pixel coordinates of the corner points):

$$\begin{aligned} \overline{\lambda}_5^p \simeq \overline{p}_1 \times \overline{p}_3 \quad &\Longrightarrow \quad V_3^p \simeq \overline{\lambda}_5^p \times \overline{\lambda}_H^p \\ \overline{\lambda}_6^p \simeq \overline{p}_2 \times \overline{p}_4 \quad &\Longrightarrow \quad V_4^p \simeq \overline{\lambda}_6^p \times \overline{\lambda}_H^p \end{aligned}$$

Then, the orthogonality of V_3 and V_4 yields $(V_3^p)^T \left(K^T K\right) (V_4^p) = 0$, or:

$$\frac{a_3\, a_4}{f_x^2} + \frac{b_3\, b_4}{f_y^2} + c_3\, c_4 = 0 \tag{3.12}$$

where a_3, a_4, b_3, b_4, c_3 and c_4 are the known pixel coordinates of V_3^p and V_4^p: $V_3^p \simeq [a_3 \;\; b_3 \;\; c_3]^T$ and $V_4^p \simeq [a_4 \;\; b_4 \;\; c_4]^T$. This constitutes a second constraint on f_x and f_y (a second hyperbola in the (f_x^2, f_y^2) plane), which can be written together with equation 3.11 in a form of a linear equation in $\overline{u} = [u_1 \;\; u_2]^T = [1/f_x^2 \;\; 1/f_y^2]^T$:

$$\mathcal{A}\,\overline{u} = \overline{b} \quad \text{with} \quad \mathcal{A} = \begin{bmatrix} a_1 a_2 & b_1 b_2 \\ a_3 a_4 & b_3 b_4 \end{bmatrix} \quad \text{and} \quad \overline{b} = - \begin{bmatrix} c_1 c_2 \\ c_3 c_4 \end{bmatrix}$$

If $\mathcal{A}$ is invertible, then both focals f_x and f_y may be recovered explicitly:

$$\overline{u} = \begin{bmatrix} 1/f_x^2 \\ 1/f_y^2 \end{bmatrix} = \mathcal{A}^{-1}\,\overline{b} \;\Rightarrow\; \begin{cases} f_x = \sqrt{1/u_1} \\ f_y = \sqrt{1/u_2} \end{cases}$$

or:

$$f_x = \sqrt{\frac{a_1 a_2 b_3 b_4 - a_3 a_4 b_1 b_2}{b_1 b_2 c_3 c_4 - b_3 b_4 c_1 c_2}}$$
$$f_y = \sqrt{\frac{a_1 a_2 b_3 b_4 - a_3 a_4 b_1 b_2}{a_3 a_4 c_1 c_2 - a_1 a_2 c_3 c_4}}$$

under the condition $u_1 > 0$ and $u_2 > 0$.

If $\mathcal{A}$ is not invertible (or $a_1 a_2 b_3 b_4 - a_3 a_4 b_1 b_2 = 0$), then both focals (f_x, f_y) cannot be recovered. However, if $\mathcal{A}$ is of rank one (i.e. it is not the zero matrix), then a single focal length model $f_c = f_x = f_y$ may be used. The following claim gives a necessary and sufficient condition for $\mathcal{A}$ to be rank one:

Claim 2: The matrix $\mathcal{A}$ is rank one if and only if the projection $\overline{\lambda}_H$ of the horizon line is parallel to either the x or y axis on the image plane (its first or second coordinate is zero, not both), or crosses the origin on the image plane (its last coordinate is zero). Since the matrix matrix K is diagonal, this condition also applies to the horizon line in pixel coordinates $\overline{\lambda}_H^p$.

Corollary: Since $\overline{\lambda}_H$ is proportional to the surface normal vector $\overline{n}_h$ (from proposition 2 in section 2.2.2), this degeneracy condition only depends upon the 3D orientation of the plane Π_h with respect to the camera, and not the way the calibration grid is positioned onto it (this is intrinsic to the geometry of the setup).

In such a rank-one degenerate case, the reduced focal model is acceptable. Then both constraint equations 3.11 and 3.12 may be written as a function of a unique focal f_c as follows:

$$\begin{bmatrix} c_1 c_2 \\ c_3 c_4 \end{bmatrix} f_c^2 = - \begin{bmatrix} a_1 a_2 + b_1 b_2 \\ a_3 a_4 + b_3 b_4 \end{bmatrix}$$

which may be solved in a least squares fashion, yielding the following solution:

$$f_c = f_x = f_y = \sqrt{-\frac{c_1 c_2 (a_1 a_2 + b_1 b_2) + c_3 c_4 (a_3 a_4 + b_3 b_4)}{c_1^2 c_2^2 + c_3^2 c_4^2}} \tag{3.13}$$

Alternative estimates may be derived by directly solving for either one of the constraint equations (3.11 or 3.12) taking $f_x = f_y = f_c$. This may be more appropriate in the case where one of the four vanishing points V_k is at infinity (corresponding to $c_k = 0$). It is then better to drop the associate constraint and only consider the remaining one (remark: having a vanishing point at infinity does not necessarily mean that the matrix $\mathcal{A}$ is singular). Since the vector $\overline{\lambda}_H$ is parallel to the normal vector $\overline{n}_h$ of the ground plane Π_h, this rank-one degeneracy case corresponds to having one of the camera axis X_c, Y_c or Z_c parallel to the calibration plane Π_h.

Note that if two vanishing points are at infinity, then the projection of the entire horizon line, $\overline{\lambda}_H$ is also at infinity on the image plane (its two first coordinates are zero). This occurs only when the calibration plane Π_h is strictly parallel to the image plane (or $\overline{n}_h = [0 \;\; 0 \;\; 1]^T$), which is known to be a degenerate case where there exists no solution for calibration.

In the case where the planar pattern is a rectangle, but not necessarily a square (or equivalently, the aspect ratio of the rectangle is not known), then the diagonal constraint is not available (equation 3.12). In that case, only equation 3.11 is available to estimate focal length. It is therefore necessary to use a reduced single focal model $f_c = f_x = f_x$:

$$f_c = f_x = f_y = \sqrt{-\frac{a_1 a_2 + b_1 b_2}{c_1 c_2}} \tag{3.14}$$

This expression will be used in a calibration experiment illustrated in figure 3.5.

Once the camera matrix K is estimated, the normalized horizon vector $\overline{\lambda}_H \simeq K^* \overline{\lambda}_H^p$ may be recovered. From proposition 2, this vector is known to be proportional to the coordinate vector $\overline{\omega}_h$ of Π_h (or its normal vector $\overline{n}_h$). Therefore, this directly provides the orientation in 3D space of the ground plane. The only quantity left to

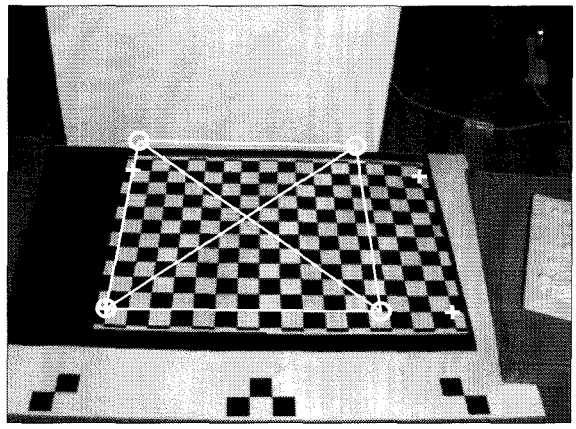


Figure 3.4: Camera calibration image: The extreme corners of the rectangular pattern are marked with crosses. This rectangle (of size 67.7 cm $\times$ 42.21 cm) is warped into a perspective view of a square. This square, marked with white circles, is used for computing the two pairs of orthogonal vanishing points (V_1^p, V_2^p) and (V_3^p, V_4^p) .

estimate is then its absolute distance d_h to the camera center, or equivalently the norm $\|\overline{\omega}_h\| = 1/d_h$. This step may be achieved by making use of the known area of the square ABCD and applying an inverse perspective projection on it (possible since the orientation of Π_h is known).

Implementation details: In principle, only the four extreme corners of the rectangular pattern are necessary to localize the four vanishing points V_1^p, V_2^p, V_3^p and V_4^p. However, in order to be less sensitive to pixel noise, it is better in practice to make use of all the detected corners on the grid (points extracted using the Harris corner finder [21]). This aspect is especially important given that vanishing point extraction is known to be very sensitive to noise in the corner point coordinates (depending on amount of depth perspective in the image). One possible approach is to fit a set of horizontal and vertical lines to the pattern points, and then recover the two vanishing points V_1^p, V_2^p by intersecting them in a least squares fashion. Once these two points are extracted, the position of the extreme corners of the rectangular pattern may be corrected by enforcing the four extreme edges of the grid to go through those vanishing points. The next step consists of warping the perspective view of the rectangle into a perspective view of a square (making use of the known aspect ratio of the original rectangle). The two remaining vanishing points V_3^p and V_4^p may then

be localized by intersecting the two diagonals of this square with the horizon line $\overline{\lambda}_H^p$ connecting V_1^p and V_2^p. Once those four points are extracted, the focal length may be estimated, together with the plane coordinate vector $\overline{\omega}_h$ following the method described earlier (using a one or two focal model).

Experimental results: Let us apply that calibration method to the image shown in figure 3.4. Notice that the four extreme corners of the rectangular pattern are marked with white crosses (the pattern is 67.7 cm $\times$ 42.21 cm large). After warping of the rectangle image into a square (whose corners are marked with white circles on the figure), the following set of vanishing points are retrieved (in pixels): $V_1^p \simeq [14060 \quad 251.76 \quad 1]^T$, $V_2^p \simeq [-36.36 \quad -809.09 \quad 1]^T$, $V_3^p \simeq [1055.2 \quad -726.95 \quad 1]^T$ and $V_4^p \simeq [-132.79 \quad -906.29 \quad 1]^T$ (since the image is 640×480, the optical center is fixed at $(c_x, c_y) = (319.5, 239.5)$).

A two-focal model (f_x, f_y) returns the solution $(f_x, f_y) = (849.40, 836.22)$ pixels, leading to an horizon line $\overline{\lambda}_H \simeq [0.0549 \quad -0.7188 \quad -0.6931]^T = \overline{n}_h$ and a distance of the plane to the camera center $d_h = 115.09$ cm. Observing that the horizon line is almost parallel to the x axis of the image plane (which is not surprising looking at the image), one may choose to use a single focal model instead. With that reduced model, equation 3.13 returns the estimate $f_c = f_x = f_y = 861.12$ pixels, leading to the very similar horizon line vector $\overline{\lambda}_H \simeq [0.0548 \quad -0.7288 \quad -0.6825]^T = \overline{n}_h$ and a distance $d_h = 114.91$ cm.

These estimates are then fed to the non-linear optimizer (see [18, 19] and equation 3.7) as initial guess for the calibration parameters. When using a two focal model, the final recovered set of parameters are: $(f_x, f_y) = (855.25, 857.04)$ pixels, $\overline{\lambda}_H \simeq [-0.0527 \quad 0.7335 \quad 0.6776]^T$, $d_h = 112.10$ cm and $k_c = -0.233$ (radial distortion factor). In the case of a single focal model, the recovered solution is: $f_c = f_x = f_y = 853.67$ pixels, $\overline{\lambda}_H \simeq [-0.0529 \quad 0.7322 \quad 0.6790]^T = \overline{n}_h$, $d_h = 112.13$ cm and $k_c = -0.233$. Notice how close the final estimates are to the ones computed using the direct closed form method. The difference comes mostly from the radial distortion lens model that can only be included when using all the grid points on the image.

In a second experiment, we apply the same algorithm on a "natural" image of

Figure 3.5: Calibration on a natural image: The two sets of parallel lines on the ground floor are used to infer the two vanishing points V_1^p and V_2^p (in pixel coordinates). The line connecting them is the horizon line λ_H^p attached to the plane. The image is 341 × 510 pixels.

an airport corridor (see figure 3.5). On that image, the floor tiles are known to be rectangular, but not necessarily square (or alternatively, the aspect ratio of the rectangles is not known). Therefore, one can only apply a single focal length model whose main solution is given by equation 3.14. The two vanishing points V_1^p and V_2^p are first estimated by intersecting pairs of parallel lines (see figure 3.5): $V_1^p \simeq [784.53 \;\; 146.22 \;\; 1]^T$, $V_2^p \simeq [-81.3234 \;\; 148.1453 \;\; 1]^T$. Then, equation 3.14 returns an estimate for the focal $f = f_x = f_y = 378.06$ pixels. Using that focal value, we can estimate the aspect ratio of the rectangular pattern (long segment length over short segment length) to 1.2 which is significantly different from 1 (estimation based on the angle between the two diagonal vanishing points V_3 and V_4). This confirms the fact that the pattern is not square. The horizon line λ_H, or equivalently the surface normal vector, was then estimated to $\lambda_H \simeq [0.0021 \;\; 0.9623 \;\; 0.2721]^T$. This means that the camera was tilted down towards the floor by $\theta = \arctan(\lambda_H[1]/\lambda_H[3]) = 15.8$ degrees (with the camera X-axis parallel to the ground plane). In that example, the limited amount of information in the image does not allow us to apply a final minimization including lens distortions.

3.2.2 When using a 3D calibration rig

Let us generalize the results to the case where a 3D rig is used for calibration. Figure 3.6 shows a perspective view of a cube in 3D. From that image, one may extract

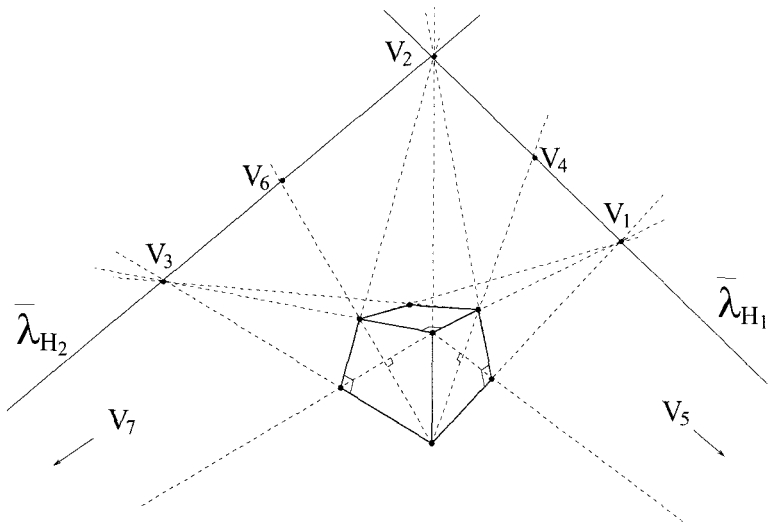


Figure 3.6: Camera calibration using a cubic rig: Perspective view of a cube in space.

seven vanishing points V_1, V_2, ... V_7. Similarly to the case of a planar square, this set of points must satisfy five orthogonality properties: $V_1 \perp V_2$, $V_1 \perp V_3$, $V_2 \perp V_3$, $V_4 \perp V_5$ and $V_6 \perp V_7$. Then, similarly to equation 3.10, we can write a set of five scalar constraints on the pixel coordinates of the vanishing points:

$$\begin{cases} V_1 \perp V_2 & \Longleftrightarrow & (V_1^p)^T \left(K^T K\right) (V_2^p) = 0 \\ V_1 \perp V_3 & \Longleftrightarrow & (V_1^p)^T \left(K^T K\right) (V_3^p) = 0 \\ V_2 \perp V_3 & \Longleftrightarrow & (V_2^p)^T \left(K^T K\right) (V_3^p) = 0 \\ V_4 \perp V_5 & \Longleftrightarrow & (V_4^p)^T \left(K^T K\right) (V_5^p) = 0 \\ V_6 \perp V_7 & \Longleftrightarrow & (V_6^p)^T \left(K^T K\right) (V_7^p) = 0 \end{cases} \tag{3.15}$$

where K is the intrinsic camera matrix and $V_i^p \simeq [a_i \quad b_i \quad c_i]^T$ $(i = 1, \ldots, 7)$ are the pixel coordinate vectors of the vanishing points (see figure 3.6). Note that the first three constraints in (3.15) enforce mutual orthogonality of the faces of the cube, whereas the last two force the left and right faces (and therefore all the others) to be squares.

Given those five independent constraints, one should be able to estimate a full 5 degrees of freedom (DOF) camera model for metric calibration including two focal lengths (f_x and f_y in pixels), the optical center coordinates (c_x and c_y in pixels) and

the skew factor α (see equation 3.2). In that case, the intrinsic camera matrix K takes its most general form [31]:

$$K = \begin{bmatrix} 1/f_x & \alpha/f_x & -c_x/f_x \\ 0 & 1/f_y & -c_y/f_y \\ 0 & 0 & 1 \end{bmatrix}$$

This model matches precisely the notation introduced in equation (3.2). Then, the semi-positive definite matrix $K^T K$ may be written as follows:

$$K^T K = \frac{1}{f_x^2} \begin{bmatrix} 1 & \alpha & -c_x \\ \alpha & \alpha^2 + (f_x/f_y)^2 & -\alpha c_x - c_y (f_x/f_y)^2 \\ -c_x & -\alpha c_x - c_y (f_x/f_y)^2 & f_x^2 + c_x^2 + c_y^2 (f_x/f_y)^2 \end{bmatrix} \tag{3.16}$$

$$= \frac{1}{f_x^2} \begin{bmatrix} 1 & u_5 & -u_3 \\ u_5 & u_2 & -u_4 \\ -u_3 & -u_4 & u_1 \end{bmatrix} \tag{3.17}$$

Notice that the vanishing point constraints (3.15) are homogeneous. Therefore, one can substitute $K^T K$ by its proportional matrix $f_x^2 K^T K$. Doing so, the five constraints listed in equation 3.15 are linear in the vector $\overline{u} = [u_1 \ \cdots \ u_5]^T$. Indeed, for $(i,j) \in \{(1,2),(1,3),(2,3),(4,5),(6,7)\}$, we have:

$$\begin{bmatrix} -c_i c_j & -b_i b_j & (a_i c_j + a_j c_i) & (b_i c_j + b_j c_i) & -(a_i b_j + a_j b_i) \end{bmatrix} \overline{u} = a_i a_j, \tag{3.18}$$

Therefore, this set of 5 equations may be written in a form of a linear system of 5 equations in the variable $\overline{u}$:

$$\mathcal{A}\,\overline{u} = \overline{b} \tag{3.19}$$

where $\mathcal{A}$ is a 5×5 matrix, and $\overline{b}$ a 5-vector:

$$\mathcal{A} = \begin{bmatrix} -c_1c_2 & -b_1b_2 & (a_1c_2 + a_2c_1) & (b_1c_2 + b_2c_1) & -(a_1b_2 + a_2b_1) \\ -c_1c_3 & -b_1b_3 & (a_1c_3 + a_3c_1) & (b_1c_3 + b_3c_1) & -(a_1b_3 + a_3b_1) \\ -c_2c_3 & -b_2b_3 & (a_2c_3 + a_3c_2) & (b_2c_3 + b_3c_2) & -(a_2b_3 + a_3b_2) \\ -c_4c_5 & -b_4b_5 & (a_4c_5 + a_5c_4) & (b_4c_5 + b_5c_4) & -(a_4b_5 + a_5b_4) \\ -c_6c_7 & -b_6b_7 & (a_6c_7 + a_7c_6) & (b_6c_7 + b_7c_6) & -(a_6b_7 + a_7b_6) \end{bmatrix}, \quad \overline{b} = \begin{bmatrix} a_1a_2 \\ a_1a_3 \\ a_2a_3 \\ a_4a_5 \\ a_6a_7 \end{bmatrix}.$$

If the matrix $\mathcal{A}$ is invertible, this systems admits a solution $\overline{u} = \mathcal{A}^{-1}\,\overline{b}$. Finally, the intrinsic camera parameters are retrieved from $\overline{u}$ as follows:

$$\begin{cases} f_x &= \sqrt{u_1 - u_3^2 - \frac{(u_4 - u_3\,u_5)^2}{u_2 - u_5^2}} \\ f_y &= f_x / \sqrt{u_2 - u_5^2} \\ c_x &= u_3 \\ c_y &= \frac{u_4 - u_3\,u_5}{u_2 - u_5^2} \\ \alpha &= u_5 \end{cases} \tag{3.20}$$

This final step consisting of de-embedding the intrinsic parameters from the vector $\overline{u}$ is equivalent to a Choleski decomposition of the matrix $K^T K$ in order to retrieve K.

If $\mathcal{A}$ is not invertible, then the camera model needs to be reduced. A similar situation occurs when only one face of the rectangular parallelepiped is known to be square. In that case, one of the last two constraints of (3.15) has to be dropped, leaving only 4 equations. A first model reduction consists of setting the skew factor $\alpha = 0$, and keeping as unknowns the two focals (f_x, f_y) and the camera center (c_x, c_y). That approximation is very reasonable for most cameras currently available. The resulting camera matrix K has the form:

$$K = \begin{bmatrix} 1/f_x & 0 & -c_x/f_x \\ 0 & 1/f_y & -c_y/f_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.21}$$

leading to the following $K^T K$ matrix:

$$K^T K = \frac{1}{f_x^2}\begin{bmatrix} 1 & 0 & -c_x \\ 0 & (f_x/f_y)^2 & -c_y\,(f_x/f_y)^2 \\ -c_x & -c_y\,(f_x/f_y)^2 & f_x^2 + c_x^2 + c_y^2\,(f_x/f_y)^2 \end{bmatrix} = \frac{1}{f_x^2}\begin{bmatrix} 1 & 0 & -u_3 \\ 0 & u_2 & -u_4 \\ -u_3 & -u_4 & u_1 \end{bmatrix}$$

Then, each constraint $(V_i^p)^T \left(K^T K\right) \left(V_j^p\right) = 0$ may be written in the form of a linear equation in $\overline{u} = [u_1 \ \cdots \ u_4]^T$:

$$\begin{bmatrix} -c_i c_j & -b_i b_j & (a_i c_j + a_j c_i) & (b_i c_j + b_j c_i) \end{bmatrix} \overline{u} = a_i a_j, \tag{3.22}$$

resulting in a 4×4 linear system $\mathcal{A}\overline{u} = \overline{b}$, admitting the solution $\overline{u}$ if $\mathcal{A}$ is rank 4. The intrinsic camera parameters (f_x, f_y, c_x, c_y) may then be computed from the vector $\overline{u}$ following the set of equations (3.20) setting $\alpha = u_5 = 0$. When $\mathcal{A}$ has rank less than 4, the camera model needs to be further reduced (that is the case when only 3 orthogonality constraints are available). A second reduction consists of using a single focal $f_c = f_x = f_y$, leading to a 3 DOF model. In that case, the K matrix takes on the following reduced expression:

$$K = \begin{bmatrix} 1/f_c & 0 & -c_x/f_c \\ 0 & 1/f_c & -c_y/f_c \\ 0 & 0 & 1 \end{bmatrix}$$

$$\Longrightarrow \quad K^T K = \frac{1}{f_c^2}\begin{bmatrix} 1 & 0 & -c_x \\ 0 & 1 & -c_y \\ -c_x & -c_y & (f_c^2 + c_x^2 + c_y^2) \end{bmatrix} = \frac{1}{f_c^2}\begin{bmatrix} 1 & 0 & -u_2 \\ 0 & 1 & -u_3 \\ -u_2 & -u_3 & u_1 \end{bmatrix}$$

Then, each constraint listed in (3.15) can be written in the form of a linear equation of the variable $\overline{u} = [u_1 \ u_2 \ u_3]^T$:

$$(V_i^p)^T \left(K^T K\right) \left(V_j^p\right) = 0 \iff \begin{bmatrix} -c_i c_j & (a_i c_j + a_j c_i) & (b_i c_j + b_j c_i) \end{bmatrix} \overline{u} = (a_i a_j + b_i b_j)$$

Once again, this leads to the linear problem $\mathcal{A}\,\overline{u} = \overline{b}$ where $\mathcal{A}$ is a 5×3 matrix, and $\overline{b}$ a 5-vector (if all five constraints are valid). A least squares solution is in general possible: $\overline{u} = \left(\mathcal{A}^T\mathcal{A}\right)^{-1}\mathcal{A}^T\,\overline{b}$. Notice that if the faces of the rig are known mutually orthogonal but not necessarily square, then only the three first constraints of (3.15) are enforceable. In that case, the linear system is 3×3, and its solution is $\overline{u} = \mathcal{A}^{-1}\,\overline{b}$. Once the vector $\overline{u}$ is recovered, the intrinsic camera parameters have the following expression:

$$\begin{cases} f_c = f_x = f_y &= \sqrt{u_1 - u_2^2 - u_3^2} \\ c_x &= u_2 \\ c_y &= u_3 \end{cases} \tag{3.23}$$

We applied the 3 DOF and 4 DOF camera calibration models on the image shown in figure 3.7 (image size: 276×185). On this image, the edges of the container are used to infer the first three mutually orthogonal vanishing points: $V_1^p = [279.1375 \quad 135.2584 \quad 1]^T$, $V_2^p = [156.2150 \quad -498.7323 \quad 1]^T$ and $V_2^p = [-21.5611 \quad 118.7831 \quad 1]^T$. Following the 3 DOF camera model given by equation 3.23, we retrieve the intrinsic parameters: $f = f_x = f_y = 146$ pixels, $(c_x, c_y) = (123.9, 90.6)$ pixels. Consequently, the right, left and bottom faces have the respective surface normals $\overline{n}_{\text{right}} = [0.6993 \quad -0.1356 \quad -0.7018]^T$, $\overline{n}_{\text{left}} = [-0.7128 \quad -0.2052 \quad -0.6706]^T$ and $\overline{n}_{\text{bottom}} = [0.0531 \quad -0.9693 \quad 0.2402]^T$. Using the four diagonal vanishing points V_4, V_5, V_6 and V_7 shown on figure 3.6, the aspect ratios between the segment lengths of the container may be estimated: $L_2/L_1 = 4.5$, $L_3/L_1 = 0.98$. Observing that the left face is "almost" square (since $L_3/L_1 \approx 1$), we add the corresponding constraint (the last one in (3.15)) and solve for the same camera model with 4 constraint equations instead of three. The least squares problem leads then to a very similar solution: $f = f_x = f_y = 145.71$ pixels and $(c_x, c_y) = (118.7, 91.4)$ pixels, making $L_2 = 4.57\,L_1$ and $L_3 = L_1$ (enforced by the constraint). Finally, we apply the 4 DOF camera model onto the 4 constraints and retrieve very similar camera parameters: $(f_x, f_y) = (146.11, 143.79)$ pixels and $(c_x, c_y) = (122.81, 91.67)$ pixels.

Figure 3.7: Camera calibration using a 3 and 4 DOF camera model on a natural image

When $\mathcal{A}$ has rank less than 3, the model needs to be further reduced to 2 or 1 DOF by taking the optical center out of the model (fixing it to the center of the image) and then going back to the original model adopted in the planar rig case (one or two focal models).

3.3 Conclusions

In this chapter, we applied the dual-space formalism to the problem of camera calibration. This approach enabled us to decouple intrinsic from extrinsic parameters and derive a set of closed-form solutions for intrinsic camera calibration in the case of five model orders: 1, 2, 3, 4 and 5 degrees of freedom. In addition, we stated conditions of observability of those models under different experimental situations corresponding to planar and three-dimensional rigs. The following table summarizes the results by giving, for each model order, the list of parameters we have retrieved explicit expressions for, as well as the minimum structural rig necessary to estimate the model:

Model order	Parameters	Calibration rig (minimum required structure)
1 DOF	$f = f_x = f_y$	2D rectangle
2 DOF	f_x, f_y	2D square
3 DOF	$f = f_x = f_y$, c_x, c_y	3D rectangular parallelepiped
4 DOF	f_x, f_y, c_x, c_y	3D rectangular parallelepiped with one square face
5 DOF	f_x, f_y, c_x, c_y, α	3D cube

One could use this explicit set of solutions for calibrating image sequences where intrinsic camera parameters are time varying. A typical sequence could be a flyby movie over a city with buildings.

In a broader sense, this work provides a general framework for approaching problems involving reconstruction of three-dimensional scenes with known structural constraints (for example orthogonality of building walls in a picture of a city). Indeed, constraints, such as parallelism or orthogonality, find very compact and exploitable representations in dual-space. This approach may avoid traditional iterative minimization techniques (computationally expensive) in order to exploit constraints in a scene.

Chapter 4 Passive methods for 3D reconstruction

In this chapter, we describe passive visual techniques for 3D reconstruction. This class of methods is called "passive" because no other device besides camera(s) is required (images are the only input data). This limited equipment cost constitutes one competitive advantage of passive techniques compared to active techniques that require external device such as laser or LCD projectors for projecting artificial texture in the scene (see chapter 5). Of course, one intrinsic limitation of passive approaches is that they may only be applied on scenes that are sufficiently textured.

The standard structure triangulation problem is first presented in Sec. 4.1, followed in Sec. 4.2 by a description of the combined 3D motion and structure estimation problem from two views obervation. Then, section 4.3 generalizes the problem of motion estimation to the case of three views, and then N_v views ($N_v \geq 3$) for both point and line observation. Sec. 4.4 presents implementation details for processing long sequences of images, as well as some experimental results. Finally, Sec. 4.5 closes the chapter with some conclusions.

4.1 Structure estimation

4.1.1 Structure estimation from two views - Stereo problem

Let us model the 3D world by a set of N points P_i ($i = 1, \ldots, N$) in space. Assume that the cloud of points is observed from two cameras at two different positions in space. Denote by p_i and p'_i the two projections of P_i on the two image planes, and let $\overline{x}_i = [x_i \;\; y_i \;\; 1]^T$ and $\overline{x}'_i = [x'_i \;\; y'_i \;\; 1]^T$ be their respective normalized homogeneous coordinates (the two cameras are assumed to be calibrated). Let $\overline{X}_i = [X_i \;\; Y_i \;\; Z_i]^T$

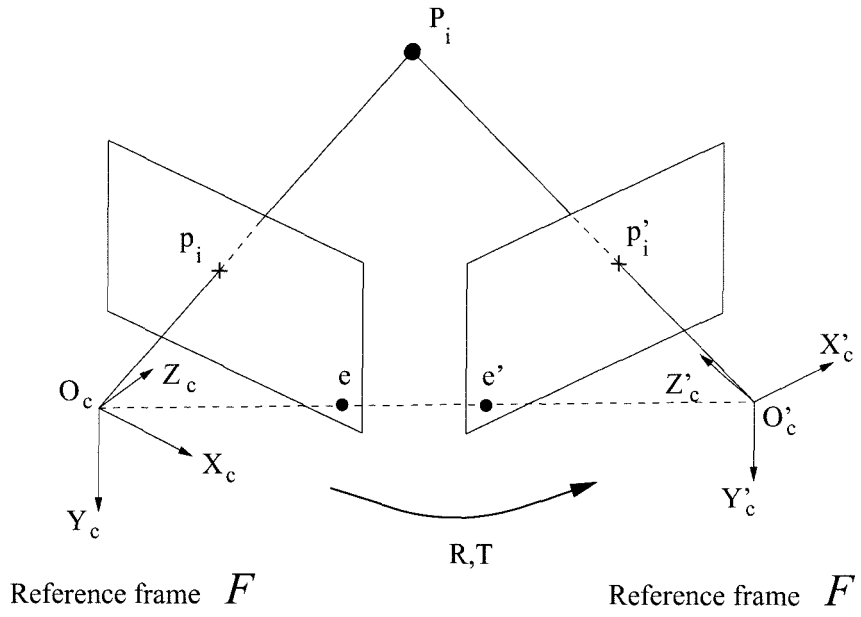


Figure 4.1: Stereo triangulation consists of retrieving the point P_i in space from its two observed projection p_i and p_i' onto the two image planes. Triangulation is impossible if P_i lies on (O_c, O_c'), or equivalently is $p_i = e$ or $p_i' = e'$ where e and e' are called the epipols.

and $\overline{X}_i' = [X_i' \;\; Y_i' \;\; Z_i']^T$ be the Euclidean coordinates of P_i in the reference frames attached to the two camera locations (denoted $\mathcal{F}$ and $\mathcal{F}'$ respectively).

Let R and T be the rigid motion parameters (rotation matrix and translation vector) between the two camera positions ($\mathcal{F}$ and $\mathcal{F}'$). Then, the two coordinate vectors $\overline{X}_i$ and $\overline{X}_i'$ are related to each other through the standard rigid body transformation equation (see equation 2.13):

$$\forall i = (1, \ldots, N), \qquad \overline{X}_i' = R\overline{X}_i + T \tag{4.1}$$

Then, the stereo triangulation problem consists of retrieving the 3D coordinate vectors $\overline{X}_i$ and $\overline{X}_i'$ from the image coordinates $\overline{x}_i$ and $\overline{x}_i'$ assuming that the relative position of the two cameras is known (R and T). This corresponds to intersecting the two rays (O_c, p_i) and (O_c', p_i') in space (see figure 4.1). Another fundamental assumption that is made here is that the correspondence between the two image projections has been previously established. In other words, we assume that the point p_i in the first image is known to correspond to the point p_i' in the other image, and

not to any other point p'_j $(j \neq i)$. This correspondence is often denoted $p_i \leftrightarrow p'_i$. In practice, establishing correspondence between two images is not a trivial task. In the case where the disparity between the two images is large then the only method that can be used is global image search (this is often the case when the two camera positions are far apart). This approach is known to be highly computationally expensive and is often not guaranteed to find the right solution due to local minima. When the two images are relatively similar, differential methods such as optical flow computation may be applied [32, 33, 34, 35]. These strategies are a lot less computationally expensive, and do not suffer from local minima as much as global search methods (since computations are done within small image neighborhoods). On the other hand, in order to apply optical flow methods, it is often required to have small camera displacements, therefore small geometrical baseline for triangulation. This will be shown to be affecting the accuracies in depth reconstruction. For now, let us go back to the fundamental estimation problem.

According to the perspective projection operator, we have $\overline{X}_i = Z_i \overline{x}_i$ and $\overline{X}'_i = Z'_i \overline{x}'_i$. Then equation 4.1 may be written:

$$Z'_i \overline{x}'_i = Z_i R \overline{x}_i + T \tag{4.2}$$

leading to the following linear system:

$$\begin{bmatrix} -R\overline{x}_i & \overline{x}'_i \end{bmatrix} \begin{bmatrix} Z_i \\ Z'_i \end{bmatrix} = T \tag{4.3}$$

Let $\mathcal{A}_i \doteq \begin{bmatrix} -R\overline{x}_i & \overline{x}'_i \end{bmatrix}$ (a 3×2 matrix). The least squares solution for 4.3 is then:

$$\begin{bmatrix} Z_i \\ Z'_i \end{bmatrix} = \left(\mathcal{A}_i^T \mathcal{A}_i\right)^{-1} \mathcal{A}_i^T T \tag{4.4}$$

Let $\overline{\alpha}_i \doteq -R\,\overline{x}_i$. From equation 4.4, an close form expression for Z_i may be expanded:

$$Z_i = \frac{\|\overline{x}'_i\|^2 \,\langle \overline{\alpha}_i, T\rangle - \langle \overline{\alpha}_i, \overline{x}'_i\rangle \,\langle \overline{x}'_i, T\rangle}{\|\overline{\alpha}_i\|^2 \,\|\overline{x}'_i\|^2 - \langle \overline{\alpha}_i, \overline{x}'_i\rangle^2} \tag{4.5}$$

In absence of noise on the point coordinate vectors $\overline{x}_i$ and $\overline{x}'_i$, equation 4.5 returns the exact value for Z_i, hence the exact position $\overline{X}_i$ of P_i in the first reference frame $\mathcal{F}$. In the case where the two vectors $\overline{x}_i$ and $\overline{x}'_i$ are noisy, then the two rays (O_c, p_i) and (O_c, p'_i) are not guaranteed to exactly intersect at a single point in space. Then, the solution provided by equation 4.5 is the point that is the closest to both rays in space.

Observe that if the two vectors $\overline{\alpha}_i$ and $\overline{x}'_i$ are proportional, then the denominator of equation 4.5 vanishes. In that case, triangulation is impossible. That corresponds to having two collinear rays (O_c, p_i) and (O'_c, p'_i) in space. Notice that this happens only for points in space lying on the line connecting the two optical centers (O_c, O'_c), or equivalently when the two projection points p_i and p'_i are at the *epipoles* e and e' (see figure 4.1). Another singular configuration occurs when $T = [0\ \ 0\ \ 0]^T$. In that case, stereo triangulation is impossible for all points in the scene. This is also known as the zero-parallax (or zero-baseline) degenerate case. In the limit, as the norm of translation goes to zero, triangulation becomes numerically more and more sensitive to noise.

4.1.2 Structure estimation from N_v views ($N_v > 2$)

Assume that the same point P in space is observed on $N_v > 2$ different views corresponding to N_v different camera locations. Then, in order to be more robust to measurement noise, it is beneficial to make use of all the information coming from all the projections for triangulation. One intuitive approach is looking for the point in space that is the closest to the set N_v optical rays generated by the N_v observations (the point in space that minimizes the sum of the squares of its distances to every ray).

For simplicity in the notation, let us drop the subscript i indexing the set of points

P_i in space, and let us focus on a unique point P in space. Let p^n $(n = 1, \ldots, N_v)$ be the projection of P on the n^{th} view, and let $\overline{x}^n = [x^n \;\; y^n \;\; 1]^T$ be its corresponding homogeneous coordinate vector. Let $\overline{X}^n = [X^n \;\; Y^n \;\; Z^n]^T$ be the coordinate vector of P in the n^{th} reference frame associated to the n^{th} camera position. Then, all vectors $\overline{X}^n$ for $n = 2, \ldots, N_v$ are related to $\overline{X}^1$ through a set of rigid transformation equations. These relations can be written:

$$\forall n = (1, \ldots, N_v), \qquad \overline{X}^n = R_n \overline{X}^1 + T_n \tag{4.6}$$

where R_n and T_n are the rotation matrix and the translation vector that define the location of the n^{th} camera with respect to the first one (notice that R_1 is the identity matrix, and T_1 the zero vector). Let $\overline{X} = \overline{X}^1$. Then, one may observe that the set of equations (4.6) may also be written:

$$\forall n = (1, \ldots, N_v), \qquad \overline{X} = \overline{O}_n + Z^n \overline{r}_n \tag{4.7}$$

where $\overline{O}_n = -R_n^T T_n$ is the coordinates of the center of projection of the n^{th} camera in the first camera reference frame, and $\overline{r}_n = R_n^T \overline{x}^n$ is the coordinate of the n^{th} optical ray direction vector in the first camera reference frame. Once the relative positions of the cameras are known (R_n and T_n), the (origin) vectors $\overline{O}_n$ are known (the camera trajectory in space). Then, the set of ray direction vectors $\overline{r}_n$ may also be computed from the observation vectors $\overline{x}^n$. Every optical ray will be denoted by the vector pair $\Delta^n = (\overline{O}_n, \overline{r}_n)$ (see figure 4.2).

The triangulation problem corresponds then to searching for the depth vector $\overline{Z} \doteq [Z^1 \;\; Z^2 \;\; \cdots \;\; Z^{N_v}]^T$ and the point coordinate $\overline{X}$ that minimize the sum of the squares of the orthogonal distances of the point P to the optical rays lines $\Delta^n = (\overline{O}_n, \overline{r}_n)$. See figure 4.2. This can be done by solving the following minimization problem:

$$\{\overline{X}, \overline{Z}\}\big|_{\text{opt}} = \underset{\overline{X},\overline{Z}}{\text{Argmin}} \; C(\overline{X}_c, \overline{Z}) \tag{4.8}$$

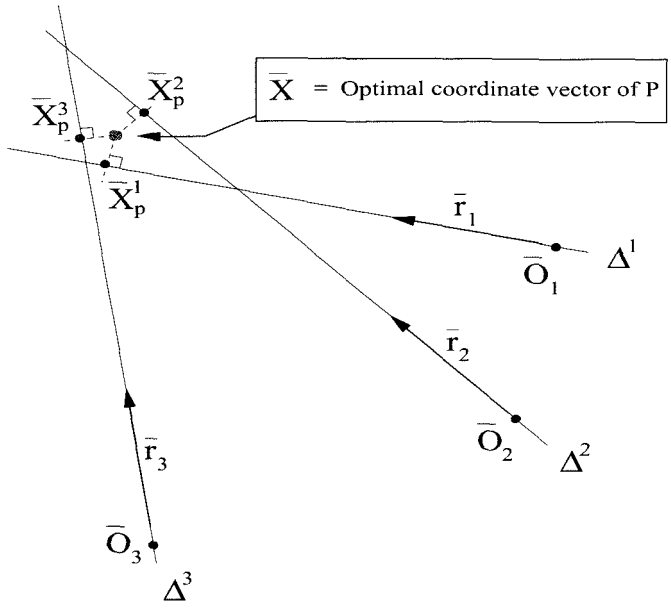


Figure 4.2: Optimal 3D triangulation from N_v views: The coordinate vector $\overline{X}$ corresponds to the optimal coordinate vector of P in space in a sense that it is point which is the closest to all the optical rays $\Delta^n = (\overline{O}_n, \overline{r}_n)$ in space $(n = 1 \ldots N_v)$. The set of points $\overline{X}_p^n$ are the orthogonal projections of $\overline{X}$ on the lines Δ^n. On this figure, we set the number of views (rays) to $N_v = 3$.

where the cost function C is defined as follows:

$$C(\overline{X}, \overline{Z}) \doteq \sum_{n=1}^{N_v} \| \left(\overline{O}_n + Z^n \, \overline{r}_n\right) \; - \overline{X} \|^2 \tag{4.9}$$

Observe that $\overline{X}_p^n \doteq \overline{O}_n + Z^n \, \overline{r}_n$ is the coordinate vector of a point P^n constrained to lie on the optical ray Δ^n. Therefore, at the optimal solution, the point P^n is expected to be the orthogonal projection of P on Δ^n. That is illustrated on figure 4.2.

Let us now derive an analytical solution for the optimal depth vector $\overline{Z}$ and the coordinate vector $\overline{X}$ of P in the first camera reference frame.

At the optimum, the Jacobian matrix of the cost function C is zero:

$$\left[\begin{array}{cc} \frac{\partial C}{\partial \overline{X}} & \frac{\partial C}{\partial \overline{Z}} \end{array} \right] = 0 \tag{4.10}$$

That leads to the following set of equations (for $i = 1, \ldots, N_v$):

$$\frac{\partial C}{\partial \overline{X}} = 0 \quad \Longrightarrow \quad \overline{X} = \frac{1}{N_v} \sum_{n=1}^{N_v} \left(\overline{O}_n + Z^n \, \overline{r}_n \right) \tag{4.11}$$

$$\frac{\partial C}{\partial Z^i} = 0 \quad \Longrightarrow \quad \left\langle \overline{r}_i, \overline{X} \right\rangle - \left\| \overline{r}_i \right\|^2 Z^i = \left\langle \overline{r}_i, \overline{O}_i \right\rangle \tag{4.12}$$

Notice that equation 4.11 gives an direct expression for the point location vector $\overline{X}$ as a function of the depth vector $\overline{Z}$. If we now insert that expression into the set of equations 4.12, we get for all $i = 1, \ldots, N_v$:

$$\left\| \overline{r}_i \right\|^2 Z^i - \frac{1}{N_v} \sum_{n=1}^{N_v} \left\langle \overline{r}_i, \overline{r}_n \right\rangle Z^n = \left\langle \overline{r}_i, \overline{\mu} - \overline{O}_i \right\rangle \tag{4.13}$$

where $\overline{\mu}$ is defined to be the mean value of all the vectors $\overline{O}_n$ (the vector coordinate of the centroid of all optical centers in the first camera reference frame):

$$\overline{\mu} = \frac{1}{N_v} \sum_{n=1}^{N_v} \overline{O}_n \tag{4.14}$$

Then, one can re-write the set of equations 4.13 in the following matrix form:

$$\mathbf{A} \, \overline{Z} = \mathbf{b} \tag{4.15}$$

where $\mathbf{A}$ is the following $N_v \times N_v$ matrix:

$$\mathbf{A} = \begin{bmatrix} \left\| \overline{r}_1 \right\|^2 & 0 & \cdots & 0 \\ 0 & \left\| \overline{r}_2 \right\|^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \left\| \overline{r}_{N_v} \right\|^2 \end{bmatrix} - \frac{1}{N_v} \mathbf{G} \tag{4.16}$$

where $\mathbf{G}$ is the Gram matrix:

$$\mathbf{G} = \begin{bmatrix} \langle \overline{r}_1, \overline{r}_1 \rangle & \langle \overline{r}_1, \overline{r}_2 \rangle & \cdots & \langle \overline{r}_1, \overline{r}_{N_v} \rangle \\ \langle \overline{r}_2, \overline{r}_1 \rangle & \langle \overline{r}_2, \overline{r}_2 \rangle & \cdots & \langle \overline{r}_2, \overline{r}_{N_v} \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \overline{r}_{N_v}, \overline{r}_1 \rangle & \langle \overline{r}_{N_v}, \overline{r}_2 \rangle & \cdots & \langle \overline{r}_{N_v}, \overline{r}_{N_v} \rangle \end{bmatrix} \tag{4.17}$$

and $\mathbf{b}$ the following N_v vector:

$$\mathbf{b} = \begin{bmatrix} \langle \overline{r}_1, \overline{\mu} - \overline{O}_1 \rangle \\ \langle \overline{r}_2, \overline{\mu} - \overline{O}_2 \rangle \\ \vdots \\ \langle \overline{r}_{N_v}, \overline{\mu} - \overline{O}_{N_v} \rangle \end{bmatrix}$$

Notice the similarity between equations (4.15) and (4.3). The optimal depth vector $\overline{Z}$ is then computed by inverting the matrix $\mathbf{A}$ in equation 4.15:

$$\overline{Z} = \mathbf{A}^{-1}\,\mathbf{b} \tag{4.18}$$

and, finally, equation 4.11 provides the optimal coordinate vector $\overline{X}$:

$$\overline{X} = \frac{1}{N_v} \sum_{n=1}^{N_v} \left(\overline{O}_n + Z^n\, \overline{r}_n \right) = \frac{1}{N_v} \sum_{n=1}^{N_v} \overline{X}_p^n \tag{4.19}$$

Notice that in equation 4.11, the vector $\overline{X}_p^n$ (for $n = 1, \ldots, N_v$) are the coordinates of the closest point P^n on the line Δ^n to the point P (or orthogonal projection). If there were no noise in the measurements, all the lines would intersect exactly at a unique point in space, and then the vectors $\overline{X}_p^n$ would all be identical to $\overline{X}$. In the presence of noise, however, we have shown that the optimal vector $\overline{X}$ is the mean value (or centroid) of the set of vectors $\left\{ \overline{X}_p^n \right\}_{n=1,\ldots,N_v}$ (from equation 4.11). Additionally, one may make use of the standard deviation vector $\delta\overline{X}$ attached to the set $\left\{ \overline{X}_p^n \right\}$ to

evaluate the accuracy on estimating each coordinate of $\overline{X}$:

$$\delta\overline{X} \doteq \sqrt{\frac{1}{N_v - 1}\sum_{n=1}^{N_v}\left(\overline{X} - \overline{X}_p^n\right)^2} \tag{4.20}$$

where the square root and square are assumed to operate on each coordinate individually. This estimate gives us an indication on how well the set of lines intersect in space.

One can finally show that, at the optimum, the standard deviation vector $\delta\overline{X}$ satisfies the following property:

$$\|\delta\overline{X}\|^2 = \frac{1}{N_v - 1}C(\overline{X}, \overline{Z}) = \frac{1}{N_v - 1}\sum_{n=1}^{N_v}\left\langle \overline{X}_p^n - \overline{X}, \overline{O}_n \right\rangle \tag{4.21}$$

It may be shown that in the case where $N_v = 2$, the depth solution given by equation 4.18 is equivalent that previously derived in the special case of two views (equation 4.4).

One may also notice that the derivations presented in that section answer the general problem of optimally intersecting a set of lines in space. These results will be used in several other applications (in sections 6.2.3 and 6.2.5).

4.2 Motion and structure from two views

So far, we have assumed that the relative positions of the cameras in space are known before structure triangulation. Essentially, the motion parameters are needed to identify the coordinates of the optical rays associated to a single point in a unique reference frame. One question remains: what can be done when the camera motion parameters are not available? In other words, is there a way of recovering the *camera ego-motion* from point observation?

In this section, we will describe a technique for recovering the motion parameters (R and T) between two camera locations, from point observation.

Let P_i $(i = 1, \ldots, N)$ be a set of points in space, and denote p_i and p_i' the

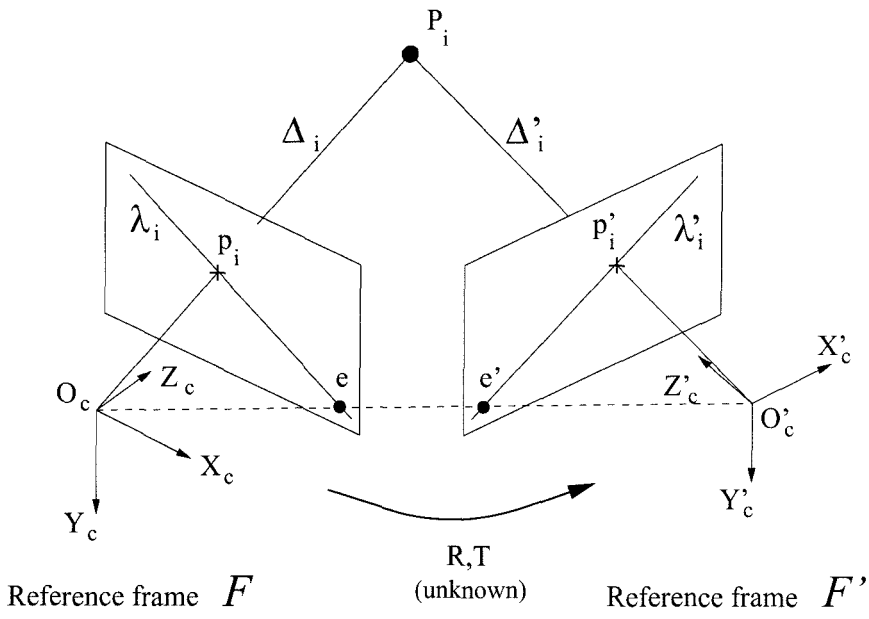


Figure 4.3: Epipolar geometry for two views: The lines λ_i and λ'_i are the two epipolar lines associated to the two point observation p'_i and p_i. The line λ_i (λ'_i) is the projections of the optical ray Δ'_i (Δ_i) on the first (second) camera image plane. The ego-motion parameters R and T must satisfy the constraints $p_i \in \lambda_i$ and $p'_i \in \lambda'_i$ for all $i = 1, \ldots, N$. They are called the two-view epipolar constraints.

perspective projections of P_i on the two camera image planes. Figure 4.3 illustrates the geometry of a single point observation. On that figure, the two optical rays Δ_i and Δ'_i associated to p_i and p'_i must be intersecting in space. That observation is the fundamental basis of all motion estimation schemes. The motion parameters R and T that brings the first camera reference frame to the second camera reference frame must satisfy the constraint that every pair of rays must intersect in space. That is also known as *geometrical constraint*. Let λ'_i be the projection of Δ_i on the second image plane (see figure 4.3). Then, one may observe that the two lines Δ_i and Δ'_i intersect in space if and only if the point p'_i lies on the line λ'_i (for $i = 1, \ldots, N$). Similarly, the two rays will intersect if and only if the point p_i lies on λ_i, the projection of Δ'_i onto the first camera image plane. The line λ_i (λ'_i) is called the *epipolar line* associated to the point P_i onto the first (second) image plane. Observe that for any point P_i in space, both lines contain the epipoles e and e' (intersections of (O_c, O'_c) with the two image planes).

Call $\overline{\lambda}_i$ and $\overline{\lambda}'_i$ the homogeneous coordinate vectors of λ_i and λ'_i respectively, and let $\overline{x}_i$ and $\overline{x}'_i$ be the coordinate vector of the two points p_i and p'_i. Then, it may be shown that:

$$\overline{\lambda}'_i \simeq Q\,\overline{x}_i \simeq \left((T\wedge)\,R\right)\,\overline{x}_i \tag{4.22}$$

$$\overline{\lambda}_i \simeq Q^T\,\overline{x}'_i \simeq \left(R^T\,(T\wedge)\right)\,\overline{x}'_i \tag{4.23}$$

where $Q = (T\wedge)\,R$ is called the *essential matrix* (the wedge operator $\wedge$ is defined in equation 2.15). Then, the two rays Δ_i and Δ'_i are coplanar (or intersect) if and only if the following condition is satisfied for all $i = 1, \ldots, N$:

$$p'_i \in \lambda'_i \quad \Longleftrightarrow \quad \left\langle \overline{x}'_i, \overline{\lambda}'_i \right\rangle = 0 \quad \Longleftrightarrow \quad \overline{x}'^T_i\,Q\,\overline{x}_i = 0 \tag{4.24}$$

This scalar equation is also known as the bilinear epipolar constraint. The very same expression is retrieved through the other equivalent constraint $p_i \in \lambda_i$. Let $\overline{\Omega} = [\Omega_x \;\; \Omega_y \;\; \Omega_z]^T$ be the rotation vector associated to the rotation matrix R (eq. 2.14). Then, the 3×3 matrix Q is function of $\overline{\Omega}$ and T (6 DOF), and may be written $Q = Q(\overline{\Omega}, T)$. Define the residual vector $\overline{e}(\overline{\Omega}, T) = [e_1 \;\; e_2 \;\; \ldots \;\; e_N]^T$ such that:

$$\forall i = 1, \ldots, N, \qquad e_i(\overline{\Omega}, T) = \overline{x}'^T_i\,Q(\overline{\Omega}, T)\,\overline{x}_i \tag{4.25}$$

Then, the motion parameters $\overline{\Omega}$ and T may be found by solving the set of equations $\overline{e}(\overline{\Omega}, T) = 0$. In presence of noise on the measurement data $\overline{x}_i$ and $\overline{x}'_i$, this equation will not be exactly satisfied. Therefore, numerically, it is necessary to define a scalar cost function to minimize. Experimentally, the 2-norm of the residual vector $\overline{e}$ a valid cost function:

$$\{\overline{\Omega}, T\}\big|_{\text{opt}} = \underset{\overline{\Omega},T}{\text{Argmin}} \sum_{i=1}^{N} e_i^2(\overline{\Omega}, T) = \underset{\overline{\Omega},T}{\text{Argmin}} \sum_{i=1}^{N} \left(\overline{x}'^T_i\,Q(\overline{\Omega}, T)\,\overline{x}_i\right)^2 \tag{4.26}$$

A standard gradient descent strategy is applied to identify the optimal solution, with

an initial guess provided by the linear method described by Longuet-Higgins [36]. Observe that the translation vector T can only be recovered up to a scale factor. Indeed, if T is solution of problem (4.26), then αT is also solution for all $\alpha \neq 0$. Physically, that corresponds to the very intuitive fact that it is not possible to identify the absolute size (or dimension) of the scene from two perspective views of it (there is no notion of a meter). Therefore, the optimization problem can only be solved constraining the translation to be unit norm ($\|T\| = 1$). This corresponds to solving for 5 DOF motion parameters (3 for rotation and 2 for translation) parameterizing the translation vector using spherical coordinates (azimuth and elevation). A large body of work may be found in the literature on the topic of two-view motion analysis [11, 37, 38, 10, 39, 40, 41, 42, 43]. All existing techniques for motion estimation are based on the same principle of solving for the set of non-linear epipolar constraints (4.24). Originally, Longuet Higgins in [36] proposed a closed-form linear method for estimating the essential matrix Q from point correspondence, and then de-embed the motion parameters R and T from Q through a Singular Value Decomposition. Our method is very much inspired from the algorithm due to Heeger and Jepson proposed in [41, 42, 43]: minimization by iterative gradient descent in the five-dimensional space of all unit translation motions. This technique has since been extended to uncalibrated cases (for example by Hartley in [44]) for recovering the *fundamental matrix* (the equivalent of Q for uncalibrated cameras). This closed-form technique is still used now to retrieve an initial guess for motion parameters. Several non-linear optimization techniques have since been proposed. Recently, some authors have proposed a dynamical framework for motion analysis [45, 46, 47, 48, 49, 50, 51] based on Kalman filtering for optimal estimation [52, 53]. Those techniques allow to include complex dynamical model for motion in order to improve estimation (for example, the dynamics of a car could be modeled if the geometry and the mass of the vehicle is known). However, if no specific dynamical motion model is known prior to estimation, those techniques often reduce after implementation to single steps motion state refinements that may very well be thought of as "generalized gradient descents." A complete description of the theory underlining this dynamical

framework may be found in [45]. Experimentally, we found no significant improvement using Kalman filtering versus standard gradient descent techniques (mainly because we never experimented using an elaborated dynamical motion model).

All those techniques are based on estimating motion from point correspondence on two views. In the next section, we propose to extend that formalism to cases where more than two views are used to infer 3D motion and structure. Then, not only points can be used as basic geometric primitives, but also lines.

4.3 Motion and structure from N_v views ($N_v > 2$)

In this section, we first generalize the problem of structure and motion estimation in the case of three view observation. This problem is tackled in such a way that many generalization are straightforward, for example when many more frames are observed at the same time, or when a set of points and lines are available. We will derive geometric constraints from point and line observation on multiple views. For example, the epipolar constraint (equation 4.24) will be a special case when having point observation on two frames. Augmenting the observation to three frames, we will see appearing a new type of object, the $3 \times 3 \times 3$ *trifocal tensor*, presented in several past publications. The main work that we wish to cite is the one due to Hartley [54], although other authors before him used similar mathematical objects. We will mention for example Weng [55] who originally treated this tensor as a set of three 3×3 matrices, and Shashua [56] who chose to represent it in the format of nine 3-vectors. To our knowledge, Vieville [57] is the first author who referred to it as a tensor. More recently, internal properties of the tensor have been quite actively studied in the uncalibrated case [58].

The main contribution of this work is in giving a very intuitive and complete geometrical description of the multi-view motion estimation problem, while providing the essential mathematical tools useful for any practical implementation.

Before starting the analysis, let us define the basic notation that will be used throughout this section. Consider a calibrated camera at three different locations in

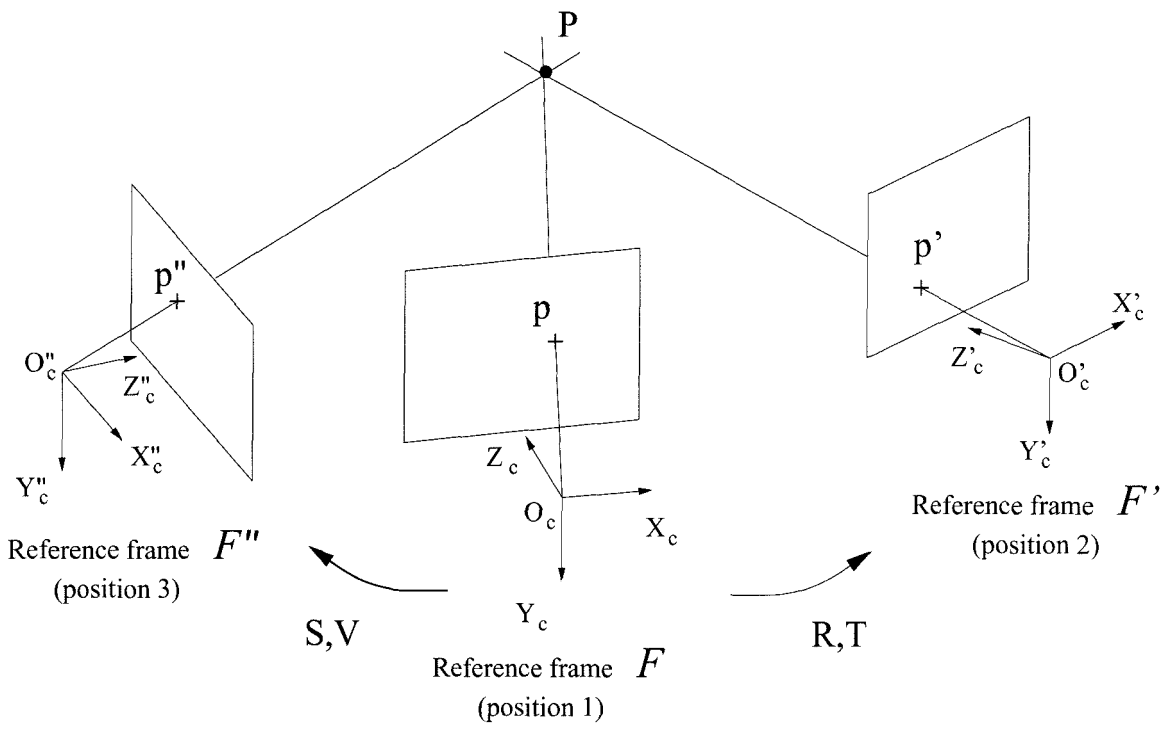


Figure 4.4: Geometry of three views: A point P in space is projected on the three camera image planes at at p, p' and p'' ($p \leftrightarrow p' \leftrightarrow p''$).

space, and define by $\mathcal{F}$, $\mathcal{F}'$ and $\mathcal{F}''$ the three associated camera reference frames. See figure 4.4. Consider a generic point P in space, and call p, p' and p'' its projection onto the three image planes. For convenience in the notation, we will denote $\overline{u} \simeq [u_1 \;\; u_2 \;\; u_3]^T$, $\overline{u}' \simeq [u'_1 \;\; u'_2 \;\; u'_3]^T$ and $\overline{u}'' \simeq [u''_1 \;\; u''_2 \;\; u''_3]^T$ the homogeneous coordinates of p, p' and p'' respectively (in practice, one may always normalize those vectors such that $u_3 = u'_3 = u''_3 = 1$). Let $\overline{\mathbf{X}} \simeq [X \;\; Y \;\; Z \;\; 1]^T$ be the homogeneous coordinate vector of the point P in the first reference frame $\mathcal{F}$ (chosen as main frame of reference), and let R and T the rigid body motion parameters between the first camera position and the second camera position (see equation 2.13), and S and V the motion parameters between position 1 and position 3. In this notation, both R and S are rotation matrices, and T and V are translation vectors:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \; T = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}, \; S = \begin{bmatrix} s_{11} & s_{12} & s_{13} \\ s_{21} & s_{22} & s_{23} \\ s_{31} & s_{32} & s_{33} \end{bmatrix}, \; V = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}. \quad (4.27)$$

Then, following equation 2.21, the three image coordinate vectors $\overline{u}$, $\overline{u}'$ and $\overline{u}''$ are linearly related to $\overline{\mathbf{X}}$ through the projection matrices $\mathbf{P}$, $\mathbf{P}'$ and $\mathbf{P}''$:

$$\begin{aligned} \overline{u} &\simeq \mathbf{P}\,\overline{\mathbf{X}} \\ \overline{u}' &\simeq \mathbf{P}'\,\overline{\mathbf{X}} \\ \overline{u}'' &\simeq \mathbf{P}''\,\overline{\mathbf{X}} \end{aligned} \tag{4.28}$$

where the three projection matrices are defined as follows (equation 2.21):

$$\begin{aligned} \mathbf{P} &= \left[\begin{array}{cc} I_{3\times 3} & 0_{3\times 1} \end{array}\right] \\ \mathbf{P}' &= \left[\begin{array}{cc} R & T \end{array}\right] \\ \mathbf{P}'' &= \left[\begin{array}{cc} S & V \end{array}\right] \end{aligned} \tag{4.29}$$

The next two sections will derive the general motion constraints for line and point observation on N_v views starting with $N_v = 3$.

4.3.1 From line correspondence

Let Λ be a line in 3D space, and denote by λ, λ' and λ'' its projections on the three image planes. See figure 4.5. The goal of this section is to derive geometrical constraints on the motion parameters R, T, S and V given the line correspondence $\lambda \leftrightarrow \lambda' \leftrightarrow \lambda''$. Let $\overline{\lambda} \simeq [\lambda_1 \;\; \lambda_2 \;\; \lambda_3]^T$, $\overline{\lambda}' \simeq [\lambda'_1 \;\; \lambda'_2 \;\; \lambda'_3]^T$ and $\overline{\lambda}'' \simeq [\lambda''_1 \;\; \lambda''_2 \;\; \lambda''_3]^T$ be the homogeneous coordinates of λ, λ' and λ'' respectively. Let Π, Π' and Π'' be the three planes spanned in space by the three images lines, and let $\overline{\pi}$, $\overline{\pi}' \simeq [\pi'_1 \;\; \pi'_2 \;\; \pi'_3 \;\; \pi'_4]^T$ and $\overline{\pi}'' \simeq [\pi''_1 \;\; \pi''_2 \;\; \pi''_3 \;\; \pi''_4]^T$ their corresponding homogeneous coordinate vectors in the camera reference frame $\mathcal{F}$. Then, following equation 2.29, the three plane coordinate vectors are naturally related to the line coordinates vectors and the motion parameters

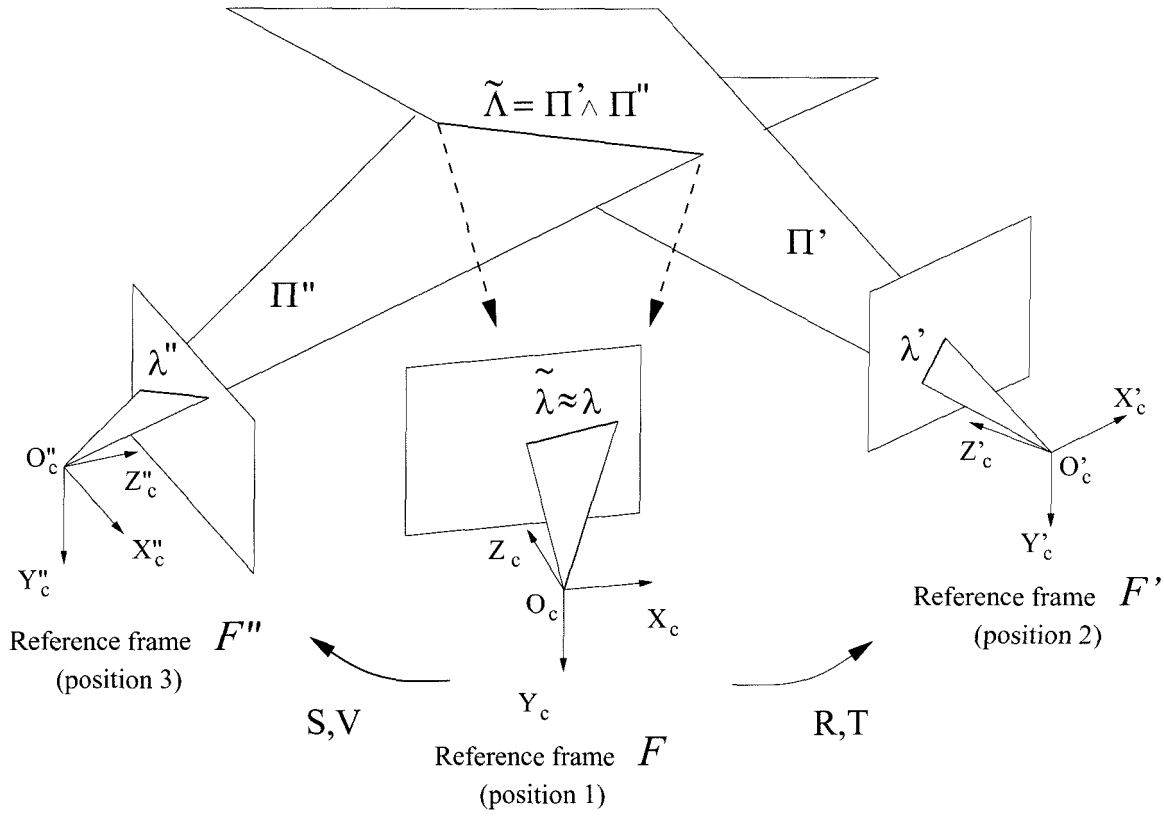


Figure 4.5: Line observation on three views: The line $\Lambda = \tilde{\Lambda}$ is projected on the three camera image planes at λ, λ' and λ''.

R, T, S and V:

$$\begin{aligned} \overline{\pi} &\simeq \mathbf{P}^T \overline{\lambda} &&= \begin{bmatrix} \overline{\lambda} \\ 0 \end{bmatrix} \\ \overline{\pi}' &\simeq \mathbf{P}'^T \overline{\lambda}' &&= \begin{bmatrix} R^T \overline{\lambda}' \\ \langle T, \overline{\lambda}' \rangle \end{bmatrix} \\ \overline{\pi}'' &\simeq \mathbf{P}''^T \overline{\lambda}'' &&= \begin{bmatrix} S^T \overline{\lambda}'' \\ \langle V, \overline{\lambda}'' \rangle \end{bmatrix} \end{aligned} \tag{4.30}$$

It is well known that at least 3 views are necessary estimating 3D rigid motion based on line observation [55]. This can be fairly easily understood by recalling that in space, in general, two planes always intersect along a line. Therefore, having only two different corresponding views of a single 3D line provides two planes of observation

which in general intersect along a line. Therefore, for any rigid motion between those two views, there will be an existing 3D line structure (namely the intersection between the planes) that will match to the observation. In other words, no information what so ever can be extracted about the rigid motion from a corresponding pair of lines on two views. However, three planes in space do not in general intersect along a line (generally, they intersect at a point). This means that not any rigid motions will be permissible for the given corresponding triplet of lines on three views. In that sense, we say that enforcing the three planes Π, Π' and Π'' to intersect along a line in space provides a constraint on the motion between the three views. This process is equivalent to constraining the 3 vectors $\overline{\pi}$, $\overline{\pi}'$ and $\overline{\pi}''$ to be linearly dependent.

An equivalent condition is to constrain the projection $\tilde{\lambda}$ of the line $\tilde{\Lambda} = \Pi' \cap \Pi''$ onto the first image plane (associated to the first camera position) to be identical to the observed line λ on that view. That principle is geometrically illustrated on figure 4.5. Notice that, at the solution (meaning for the right motion parameters R, T, S and T), the two lines in space $\tilde{\Lambda}$ and Λ are identical, and so are the two image lines $\tilde{\lambda}$ and λ. In general, the line $\tilde{\Lambda}$ and $\tilde{\lambda}$ are function of R, T, S and T. According to equation 2.12, the coordinate vector $\tilde{\overline{\lambda}} \simeq [\tilde{\lambda}_1 \quad \tilde{\lambda}_2 \quad \tilde{\lambda}_3]^T$ of $\tilde{\lambda}$ has the following expression:

$$\tilde{\overline{\lambda}} \simeq \begin{bmatrix} \pi''_4 \pi'_1 - \pi'_4 \pi''_1 \\ \pi''_4 \pi'_2 - \pi'_4 \pi''_2 \\ \pi''_4 \pi'_3 - \pi'_4 \pi''_3 \end{bmatrix} \tag{4.31}$$

or equivalently:

$$\tilde{\overline{\lambda}} \simeq \langle V, \overline{\lambda}'' \rangle R^T \overline{\lambda}' - \langle T, \overline{\lambda}' \rangle S^T \overline{\lambda}'' \tag{4.32}$$

This vector expression may also be written in the following way:

$$\tilde{\tilde{\lambda}} \simeq \begin{bmatrix} \lambda'_j \lambda''_k (r_{j1} v_k - t_j s_{k1}) \\ \lambda'_j \lambda''_k (r_{j2} v_k - t_j s_{k2}) \\ \lambda'_j \lambda''_k (r_{j3} v_k - t_j s_{k3}) \end{bmatrix} \tag{4.33}$$

where each term is summed over all values of the "dummy" indices $j = 1, 2, 3$ and $k = 1, 2, 3$ (this compact notation convention is also known as the Einstein's convention, or Einstein contraction).

Following Hartley's notations [54], we define the $3 \times 3 \times 3$ tensor $\mathcal{T}_{ijk}$ for $i, j, k = 1, \ldots, 3$ as follows:

$$\mathcal{T}_{ijk} = r_{ji} v_k - t_j s_{ki} \tag{4.34}$$

Then, the coordinates $\tilde{\lambda}_i$ of $\tilde{\lambda}$ have the expression:

$$\tilde{\lambda}_i = \lambda'_j \lambda''_k \mathcal{T}_{ijk} \tag{4.35}$$

The tensor T_{ijk} is also called *trifocal tensor.*

Notice that taking the 3 successive "slices" of the tensor $\mathcal{T}_{ijk}$ for $i = 1, 2, 3$, we get three 3×3 matrices that Weng in [55] denoted E, F and G.

The line $\tilde{\lambda}$ is actually the observation line transferred from the two last observation views back to the first one given some rigid motion parameters (R, T, S, V). In that sense, we can call that process a *line transfer.* This is based on constraining the transferred line $\tilde{\lambda}$ to be identical to the actual observed one λ. Strict equality for lines is well defined, this means that all the components of the two 3-vectors representing the lines have to be equal up to a unique non zero scale. However, to write a constraint, we need to choose a way of measuring how far, or how different two lines are on the image plane. This "difference measurement function" will give us the motion constraint equation.

This overall transferring process can be though as an equivalent to the epipolar

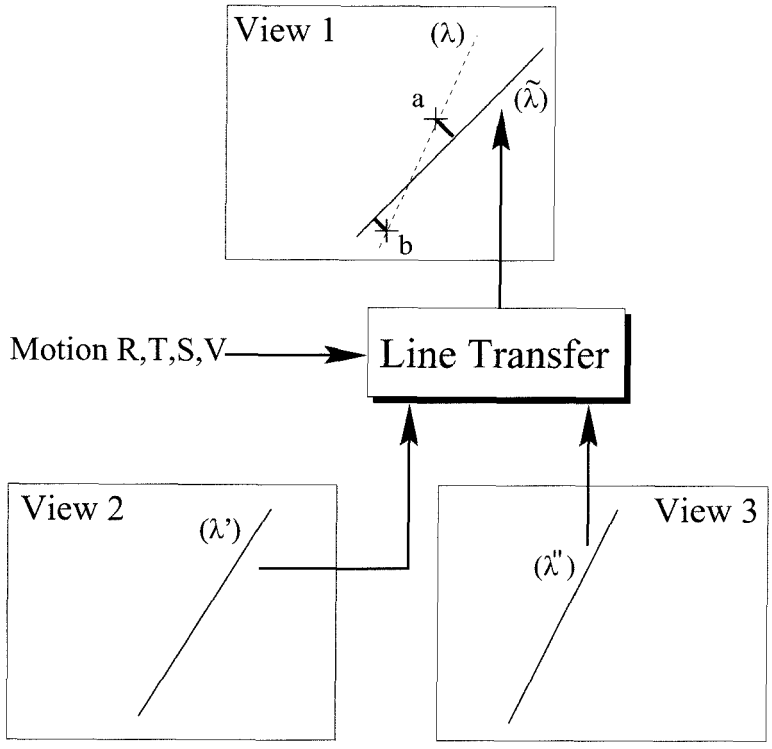


Figure 4.6: The two views 2 and 3 induce the line $\tilde{\lambda}$ on the first view. The line constraint enforces this line to be identical to the observed one λ.

constraint for point observation on two views. Indeed, in that case, the set of observed points on the first view get transferred to lines (epipolar lines) on the second view, given some rigid motion parameters between the two views (these are the intersecting lines between the image plane in the second view and the planes containing both optical centers and the observation rays in the first view). Given an point correspondence $p \leftrightarrow p'$ on the two first views, the epipolar line transferred by p to the second view is λ' whose coordinate vector is given by equation 4.22. Then, the motion constraint is based on enforcing the actual observed point on the second view p'_i to lie on its associated epipolar line λ_i (for $i = 1, \ldots, N$). The set of algebraic constraints is then given by equation 4.24.

Concerning lines observation on three views, the overall process goes similarly except that the types of the elements involved are different. The two "last" views together transfer a line $\tilde{\lambda}$ on the first one that needs to be identical to the observed line λ. One can say that 2 lines on a plane are identical if and only if they have same

"slope" and same distance to the origin. We can write the two lines λ and $\tilde{\lambda}$ in the following form:

$$\begin{cases} \overline{\lambda} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = \sqrt{\lambda_1^2 + \lambda_2^2} \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \\ d \end{bmatrix} \\ \tilde{\overline{\lambda}} = \begin{bmatrix} \tilde{\lambda}_1 \\ \tilde{\lambda}_2 \\ \tilde{\lambda}_3 \end{bmatrix} = \sqrt{\tilde{\lambda}_1^2 + \tilde{\lambda}_2^2} \begin{bmatrix} \cos(\tilde{\theta}) \\ \sin(\tilde{\theta}) \\ \tilde{d} \end{bmatrix} \end{cases} \tag{4.36}$$

where θ and $\tilde{\theta}$ (in the range $[-\pi/2, \pi/2]$) are the direction angles of the lines, and d and $\tilde{d}$ their orthogonal distances to the origin. Then, one can write the following equivalent expressions:

$$\overline{\lambda} \approx \tilde{\overline{\lambda}} \;\Leftrightarrow\; \begin{cases} \sin(\theta - \tilde{\theta}) = 0 \\ d - \tilde{d} = 0 \end{cases} \;\Leftrightarrow\; \begin{cases} \lambda_2\tilde{\lambda}_1 - \lambda_1\tilde{\lambda}_2 = 0 \\ \lambda_3\sqrt{\tilde{\lambda}_1^2 + \tilde{\lambda}_2^2} - \tilde{\lambda}_3\sqrt{\lambda_1^2 + \lambda_2^2} = 0 \end{cases} \tag{4.37}$$

Then one could keep the last pair of expressions in (4.37) as vector constraint on the motion. One disadvantage of choosing such an expression is that it involves two quantities which are not of the same nature. The first one is comparing the orientation angles (via their sine), the other the distances to the origin. This implies that one should appropriately weight them one with respect to the other, in order to make a consistent measure.

Another possible choice for motion constraints is considering the three components of the vector product $\overline{\lambda} \times \tilde{\overline{\lambda}}$, and enforce them to be zero. That corresponds to taking $\overline{\lambda} \times \tilde{\overline{\lambda}} = 0$ as motion constraint, knowing that only two out of the three resulting scalar equations are independent. That leads to:

$$\overline{\lambda} \approx \tilde{\overline{\lambda}} \;\Leftrightarrow\; \lambda \wedge \tilde{\lambda} = 0 \;\Leftrightarrow\; \begin{cases} \lambda_2\tilde{\lambda}_1 - \lambda_1\tilde{\lambda}_2 = 0 \\ \lambda_3\tilde{\lambda}_1 - \tilde{\lambda}_3\lambda_1 = 0 \\ \lambda_3\lambda_2 - \tilde{\lambda}_3\lambda_2 = 0 \end{cases} \tag{4.38}$$

This is the choice that Weng made in [55]. Notice the similarities between (4.37) and (4.38).

A third possible choice consists of using two points of the observed line λ and constrain them to lie on the transferred line $\tilde{\lambda}$. Consider that two end-points of the line λ are available: a and b of respective coordinates $\overline{u} \simeq [u_1 \quad u_2 \quad u_3]^T$ and $\overline{v} \simeq [v_1 \quad v_2 \quad v_3]^T$). Then constraining those two points to belong to the transferred line $\tilde{\lambda}$ is equivalent to enforcing the two lines λ and $\tilde{\lambda}$ to be identical.

This means that the two following relations must hold:

$$\begin{cases} \langle \overline{u}, \tilde{\overline{\lambda}} \rangle = 0 \\ \langle \overline{v}, \tilde{\overline{\lambda}} \rangle = 0 \end{cases} \tag{4.39}$$

These are precisely the two scalar constraints that Hartley chose in [54]. This corresponds to looking at the orthogonal distances of the two points a and b to the line $\tilde{\lambda}$. This process is similar to the one used when deriving the algebraic epipolar constraint in the case of point observation on two views (eq. 4.23).

Taking either one of the three constraints (4.37), (4.38) or (4.39) is algebraically possible. They all extract the maximum geometrical information.

Observe that in both equations in (4.39), the first view always contributes with its "full" point coordinates ($\overline{u}$ or $\overline{v}$) whereas the two other views are used for "line transfer" (to induce the "generalized" epipolar lines $\tilde{\lambda}$ to the first view). For this reason, this configuration is sometimes called *point-line-line* configuration [59].

One important issue to address when designing estimators is numerical stabilities, or robustness to measurement noise. Indeed, in realistic situations, all line measurements will be corrupted by noise. The way the noise affect the data is very much dependent on the line selection algorithm. Effectively, small edges on the image plane will be noisier that long ones, considering the noise in estimating the line components λ's. It might therefore not be appropriate to use directly the constraints (4.37) or (4.38) involving only λ's without any weighting equally for each line. Concretely, if for example the two lines λ' and λ'' appear to be very small on the image plane, that

means that their components λ'_i and λ''_j are very unreliable because very sensitive to image noise. That way, the transferred line $\tilde{\lambda}$ is very noisy as well. Therefore, directly comparing the components of the two lines λ and $\tilde{\lambda}$ through (4.37) or (4.38) might not be the best thing to do. The third constraint however would actually not be so much affected if the center line λ is itself inferred from two end-points u and v very close together (the line is small). Indeed, in the limit when the two points u and v are identical, the set of equations (4.39) reduces to constraining one point to lie on the transferred line (weak condition). On the other hand, the two other sets (4.37) and (4.38) still try to rely on the very noisy line components λ_i. The last proposed set of constraints has however its own limitations. The computation of $\tilde{\lambda}$ still relies on the line coordinates λ'_i and λ''_j through equation (4.35). Therefore, if either one of the two lines λ' or λ'' is inaccurate (due to image noise or small line length), the corresponding transferred line $\tilde{\lambda}$ will itself be very inaccurate. Numerically, a way to go around that conditioning problem would be to use weights on each individual line. These weights would depend on the reliability of the constraint measurement with respect to the quality of the features used. Weng in [55] was computing these weights based on the lengths of the lines. Long lines would have a larger weighting coefficient than small ones. An extension to choosing simple scalar weights is using the complete covariance matrices in the final numerical constraint. This would be the covariance matrix of the measured quantities computed from the derivative of the constraint (before modification) with respect to the feature components (the sensitivity matrix).

This numerical conditioning issue is essential for the implementation stage of the motion estimator. It is our believe that more effort should be put on studying it, as well as degeneracies [59]. In this work, we chose to put the emphasis on the geometrical properties of the system, and their associated algebraic expressions.

Regarding line observation on three views, the main observation is that one extracts exactly two independent constraints from a triplet of line correspondences. From this point on, we will choose the last constraint (4.39) acting on the detected end-points of λ. These are precisely the two scalars that Hartley chose in [54]. We

define the implicit measurement vector (or residual vector) $h_{\lambda,\lambda',\lambda''}$ as follows:

$$h_{\lambda,\lambda',\lambda''}(R,T,S,V) = \begin{bmatrix} \overline{u}^T \tilde{\overline{\lambda}} \\ \overline{v}^T \tilde{\overline{\lambda}} \end{bmatrix} \tag{4.40}$$

If N triplets of lines are available, then $2N$ scalar constraints may be retrieved in the form of a $2N$-vector (residual vector). The final step consists of finding the motion parameters R, T, S and V that minimize a (weighted) sum of the squares of the coordinates of that vector. This corresponds to a (weighted) least squares minimization. This technique may be implemented by means of gradient descent. This method gives a unique motion solution if "enough" line measurements are provided (this issue will be addressed later on).

Fig. 4.6 illustrates this principle of line transfer from 3 views. Note that the vector constraint $h_{\lambda,\lambda',\lambda''}$ as it is written in (4.40) differs from the two geometric distances of a and b to $\tilde{\overline{\lambda}}$ by the factor $\sqrt{\tilde{\lambda}_1^2 + \tilde{\lambda}_2^2}$ which is not unity since it is a function of the motion parameters (as seen in (4.32)). This is true even if the observed line vectors $\overline{\lambda}'$ and $\overline{\lambda}''$ are pre-normalized.

It is well known that from 3 perspective views of a rigid scenery, we can at most reconstruct (under calibration assumption) the 2 rigid motions (R,T) and (S,V) up to an overall scale factor for the translations. This means that the motion problem is an 11 degrees of freedom problem. If we wish to estimate these 11 parameters, we need at least 11 scalar constraints. Since each triplet of corresponding lines provides 2 scalar constraints, at least 6 line correspondences are required. This is only true while solving directly for motion parameters. An alternative way is first solving for the tensor coefficients $\mathcal{T}_{ijk}$ (appearing linearly in the constraint expression (4.40)) and then de-embed the motion from this tensor. In that case, one needs to solve for $27-1=26$ unknowns (the norm of $\mathcal{T}_{ijk}$ cannot be recovered), which requires at least $26/2 = 13$ line correspondences. The numerical methods for retrieving the camera matrices $\mathbf{P}'$ and $\mathbf{P}''$ from the tensor $\mathcal{T}_{ijk}$ is largely discussed by Hartley in [54]. This is done in two stages. In the first stage, the two unitary translation vectors $T/\|T\|$ and

$V/\|V\|$ are reconstructed by looking for the null-spaces of the three 3×3 matrices $\mathcal{T}_{1ij}$, $\mathcal{T}_{2ij}$ and $\mathcal{T}_{3ij}$. The second stage consists of retrieving the remaining coefficients (the two rotations if we are in the calibrated case). For this second stage, different methods are possible. Under calibration assumption, one may make use of that fact that the two matrices R and S are unitary, and then retrieve both of them and the ratio of norms $s = \frac{\|V\|}{\|T\|}$ (also called relative scale). Weng in [55] proposes a method for doing that. Hartley derives in [54] a numerical algorithm that deals with the uncalibrated case.

Line observation on more than three views

As previously recalled, three is the minimum number of views required for extracting motion information from line observation. In the case of three views, one line correspondence provides 2 scalar constraints corresponding to the three planes Π, Π' and Π'' having to intersect along a line in space. What happens if more than 3 views are available? In the case of $N_v = 4$ views, four such planes may be inferred (each spanned by a line on the image plane). Denote them Π, Π', Π'' and Π'''. Similarly to the three view case, we wish then to enforce all planes to intersect along a single line in space. Define the 4×4 matrix $\mathbf{A}$ as follows:

$$\mathbf{A} = \begin{bmatrix} \overline{\pi} & \overline{\pi}' & \overline{\pi}'' & \overline{\pi}''' \end{bmatrix} \tag{4.41}$$

where $\overline{\pi}$, $\overline{\pi}'$, $\overline{\pi}''$ and $\overline{\pi}'''$ are the homogeneous coordinate vectors of the four planes in the first camera reference frame.

Then, the above plane intersection condition is equivalent to enforcing any 3 vectors picked in $\{\overline{\pi}, \overline{\pi}', \overline{\pi}'', \overline{\pi}'''\}$ have to be linearly dependent. Essentially, this means that all 4 vectors can be expressed as linear combinations of any two (i.e., the matrix $\mathbf{A}$ is of rank 2). Without loss of generality, pick $\overline{\pi}'$ and $\overline{\pi}''$ as basis vectors, and enforce $\overline{\pi}$ and $\overline{\pi}'''$ to be linear combinations of them. This leads to two independent scalar constraints for each of the two planes, or a total of $2(4-2) = 4$ independent scalar constraints. In the case of N_v views, in the non-degenerate case (no 2 planes are

identical), it is sufficient to choose any $N_v - 2$ triplets of plane vectors, and enforce them to be linearly dependent to the remaining two. Since each linear dependence brings 2 scalar constraints, this leads to a total of $2(N_v-2) = 2N_v-4$ scalar constraints per line observation across N_v views.

Notice that in the non-degenerate case, if we treat N_v views as successive overlapping triplets of views, we can extract the maximum number of independent constraints $2N_v - 4$. We observe that the elementary observation cell for line correspondences is 3 views. Considering all frames at once would help to handle degenerate cases where the observed line generates two planes that are identical across 3 successive views ($\overline{\pi} \simeq \overline{\pi}'$, $\overline{\pi} \simeq \overline{\pi}''$ or $\overline{\pi}' \simeq \overline{\pi}''$). That way, one can pick constraints acting on planes belonging to far apart views.

4.3.2 From point correspondence

In this section, we derive the set of geometrical constraints provided by feature point correspondence across N_v views. Since the problem of point observation on more than two views has not yet been completely understood, we propose to address this problem in details taking increasing number of views one, two, three and generalize to N_v.

From one view

Let us first examine what can be said about a single point observation on one view. Consider a point P in space, and let $\overline{\mathbf{X}} \simeq [X \;\; Y \;\; Z \;\; 1]^T$ be its homogeneous coordinate in the camera reference frame $\mathcal{F}$. This point is projected onto the image plane at p of coordinates $\overline{u} \simeq [u_1 \;\; u_2 \;\; u_3]^T$.

Considering the first view, the matrix $\mathbf{P} = [I_{3\times3} \;\; 0_{3\times1}]$ is the projection matrix. Therefore:

$$\overline{u} \simeq \mathbf{P}\,\overline{\mathbf{X}} = \begin{bmatrix} \overline{e}_1^T \\ \overline{e}_2^T \\ \overline{e}_3^T \end{bmatrix} \overline{\mathbf{X}} = \begin{bmatrix} \overline{e}_1^T\,\overline{\mathbf{X}} \\ \overline{e}_2^T\,\overline{\mathbf{X}} \\ \overline{e}_3^T\,\overline{\mathbf{X}} \end{bmatrix} \tag{4.42}$$

where $\overline{e}_1$, $\overline{e}_2$, $\overline{e}_3$ are the following 4-vectors:

$$\overline{e}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad \overline{e}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \quad \overline{e}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \tag{4.43}$$

Equation 4.42 implies:

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \simeq \begin{bmatrix} \overline{e}_1^T \, \overline{\mathbf{X}} \\ \overline{e}_2^T \, \overline{\mathbf{X}} \\ \overline{e}_3^T \, \overline{\mathbf{X}} \end{bmatrix} \tag{4.44}$$

This "up-to-scale" vector equality gives us two linearly independent scalar constraints. Those two constraints can be chosen among the three following ones:

$$\begin{cases} (u_3 \, \overline{e}_1 - u_1 \, \overline{e}_3)^T \, \overline{\mathbf{X}} = 0 \\ (u_3 \, \overline{e}_2 - u_2 \overline{e}_3)^T \, \overline{\mathbf{X}} = 0 \\ (u_2 \, \overline{e}_1 - u_1 \overline{e}_2)^T \, \overline{\mathbf{X}} = 0 \end{cases} \tag{4.45}$$

that can be written:

$$\begin{cases} \langle \overline{\pi}_1, \overline{\mathbf{X}} \rangle = 0 \\ \langle \overline{\pi}_2, \overline{\mathbf{X}} \rangle = 0 \\ \langle \overline{\pi}_3, \overline{\mathbf{X}} \rangle = 0 \end{cases} \tag{4.46}$$

where the three 4-vectors $\overline{\pi}_1$, $\overline{\pi}_2$ and $\overline{\pi}_3$ are defined as follows:

$$\begin{cases} \overline{\pi}_1 = u_3 \, \overline{e}_1 - u_1 \, \overline{e}_3 \\ \overline{\pi}_2 = u_3 \, \overline{e}_2 - u_2 \, \overline{e}_3 \\ \overline{\pi}_3 = u_2 \, \overline{e}_1 - u_1 \, \overline{e}_2 \end{cases} \tag{4.47}$$

The system (4.46) means geometrically that the point P lies on three planes Π_1, Π_2 and Π_3 of homogeneous coordinate vectors $\overline{\pi}_1$, $\overline{\pi}_2$ and $\overline{\pi}_3$ in the camera reference

frame $\mathcal{F}$. The three plane coordinate vectors may be expanded as follows:

$$\overline{\pi}_1 \simeq \begin{bmatrix} u_3 \\ 0 \\ -u_1 \\ 0 \end{bmatrix} \quad \overline{\pi}_2 \simeq \begin{bmatrix} 0 \\ u_3 \\ -u_2 \\ 0 \end{bmatrix} \quad \overline{\pi}_3 \simeq \begin{bmatrix} u_2 \\ -u_1 \\ 0 \\ 0 \end{bmatrix} \tag{4.48}$$

Let us give a geometric interpretation of these planes: Assume that the image point coordinate vector $\overline{u}$ is normalized to $\overline{u} \simeq [u_1 \;\; u_2 \;\; 1]^T$ ($u_3 = 1$). Then, we have the following set of equivalence:

$$\begin{cases} P \in \Pi_1 & \Leftrightarrow & X - u_1 Z = 0 \\ P \in \Pi_2 & \Leftrightarrow & Y - u_2 Z = 0 \\ P \in \Pi_3 & \Leftrightarrow & u_2 X - u_1 Y = 0 \end{cases} \tag{4.49}$$

Therefore:

- Π_1 is the plane containing the center of projection O_c and the line on the image plane going through p and parallel to the (O_c, Y_c) axis (see Fig. 4.7a).
- Π_2 is the plane containing the center of projection O_c and the line on the image plane going through p and parallel to the (O_c, X_c) axis (see Fig. 4.7b).
- Π_3 is the plane containing the center of projection O_c and the line on the image plane going through p and the optical center c (see Fig. 4.7c).

Notice that the three planes contain the optical ray (O_c, P), line going through the optical center and the point P in space. Therefore they intersect along this line. That basically says that 2 planes are enough to extract all the information about the geometry of a single view. Equivalently, only two equations out the three present in the system (4.46) are independent.

We choose for simplicity a 2D symbolic representation for the planes Π_1, Π_2 and Π_3. Each plane is associated to the line of intersection between itself and the image plane. This is shown on figure 4.8. On this figure, each of the three lines are

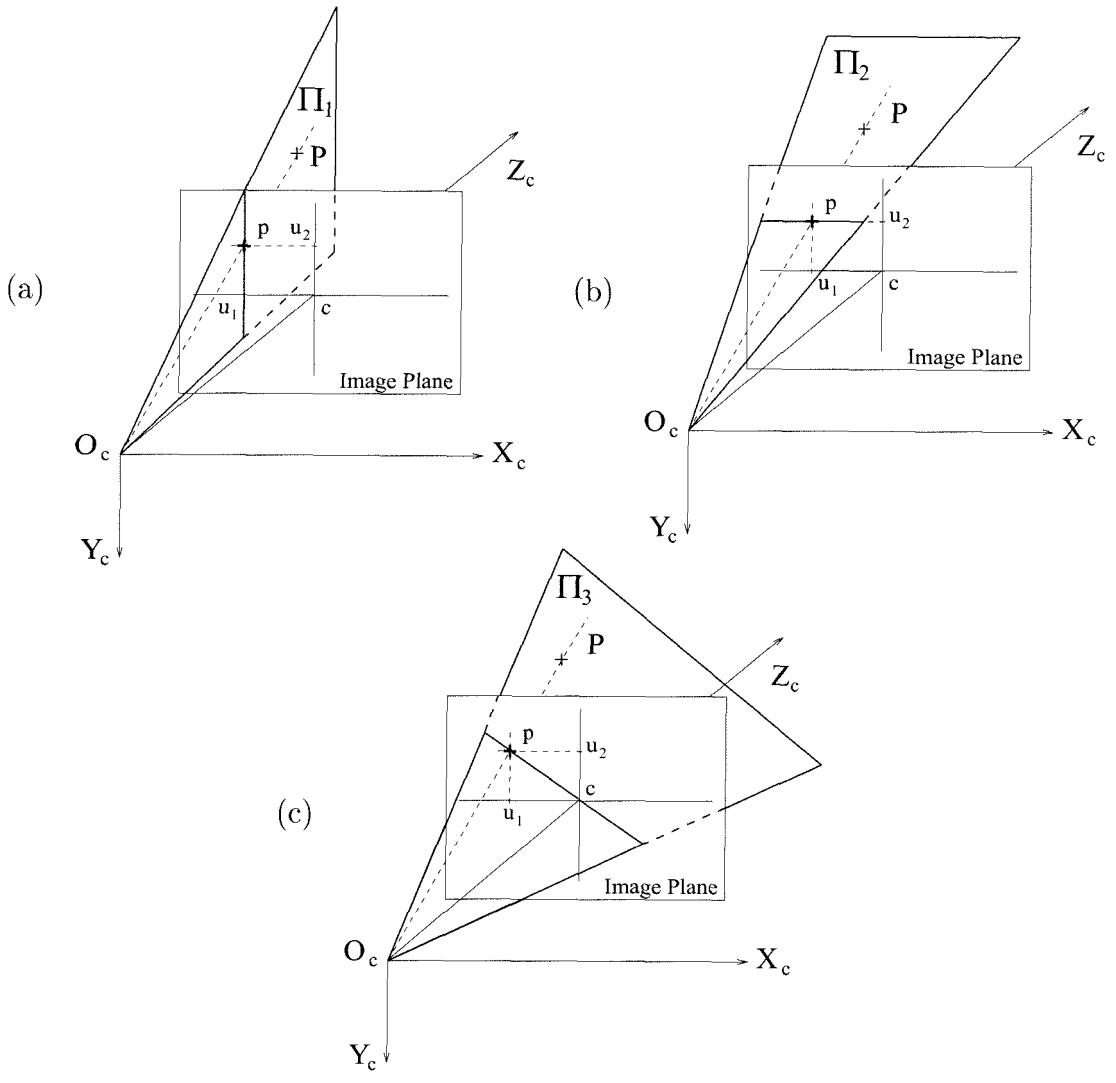


Figure 4.7: The three planes Π_1, Π_2 and Π_3 observed from the projection p of a point P in plane. Notice that all planes intersect along the optical ray (O, P).

conventionally denoted Π_1, Π_2 and Π_3.

Note that Π_1 and Π_2 will be defined for any arbitrary image point coordinate $\overline{u} \simeq [u_1 \quad u_2 \quad 1]^T$ on the image plane, unlike Π_3 which is not defined at the camera center c (at that point $\overline{u} \simeq [0 \quad 0 \quad 1]^T$). Although in principle one may choose any pair of planes among the set $\{\Pi_1, \Pi_2, \Pi_3\}$ (actually linear combinations are also permissible), in the following derivations, we will choose to keep as two independent geometric constraints the planes Π_1 and Π_2. This choice is motivated by two reasons:

- Both planes Π_1 and Π_2 are defined over the complete image plane unlike Π_3.

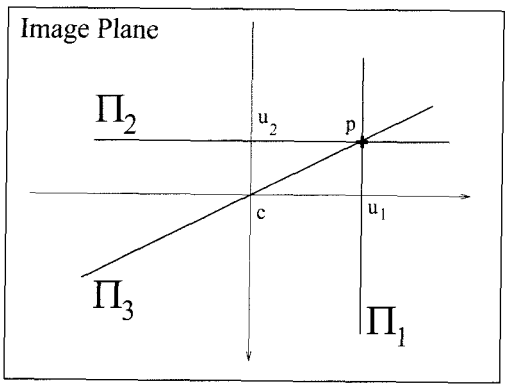


Figure 4.8: Representation on the image plane of the three planes Π_1, Π_2 and Π_3 associated to an observed point p of coordinates $\overline{u} \simeq [u_1 \quad u_2 \quad u_3]^T$. Notice that we take $u_3 = 1$.

- They naturally decouple the two image components of the observed points u_1 and u_2 (u_3=1), having Π_1 depending only on u_1 and Π_2 depending on u_2.

Actually, the fact that Π_1 and Π_2 act on the two image axes independently provides a representation that we believe could be applied to the reduced 2 dimensional case where the observation is restricted on a horizontal (resp. vertical) line. In this problem only one of the two planes (Π_1 or Π_2) would be available.

From two views

As we saw previously, observing a point p on the image plane can be thought of as observing 2 planes Π_1 and Π_2 in space that intersect along the optical ray going through the center of projection and the observed 3-D point P (see fig. 4.7ab).

Assume now that the same point P is observed on the second view, and denote p' its new projection. Let $\overline{u}' \simeq [u'_1 \quad u'_2 \quad u'_3]^T$ be its homogeneous coordinate vector. See figure 4.4. The two observation points provide two rays in space, one from each camera position. By imposing those rays to intersect in space, we naturally reach to the well known epipolar constraint (or coplanarity constraint) between the two views (eq. 4.24).

Now, instead of thinking in terms of rays, let us think in terms of planes. Those two observations actually provide two planes each. All those planes can be denoted Π_1 and Π_2 for the first view, and Π'_1 and Π'_2 for the second view (see fig. 4.9).

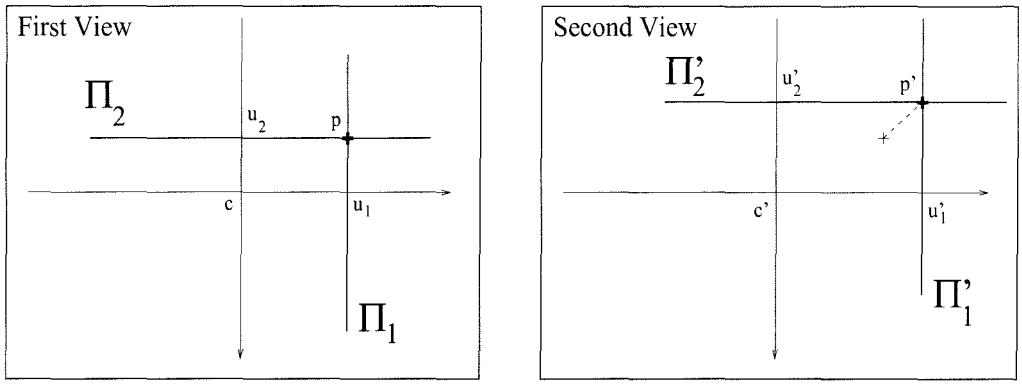


Figure 4.9: The two pairs of planes (Π_1, Π_2) and (Π'_1, Π'_2) obtained from the two projections p and p' of a single point P in space. The coordinate vectors of p and p' are $\overline{u} \simeq [u_1 \;\; u_2 \;\; u_3]^T$ and $\overline{u}' \simeq [u'_1 \;\; u'_2 \;\; u'_3]^T$ respectively. On the figure, $u_3 = u'_3 = 1$.

Let $\overline{\pi}_1$, $\overline{\pi}_2$, $\overline{\pi}'_1$ and $\overline{\pi}'_2$ be the homogeneous coordinate vectors of the four planes Π_1, Π_2, Π'_1 and Π'_2 in the first camera reference frame $\mathcal{F}$. Following the same derivations as the ones done for the one view case, we obtain the following expressions for the four plane vectors:

$$\begin{array}{c|c} \text{First view:} & \text{Second view:} \\ \overline{\pi}_1 = u_3\,\overline{e}_1 - u_1\,\overline{e}_3 & \overline{\pi}'_1 = u'_3\,\overline{a}_1 - u'_1\,\overline{a}_3 \\ \overline{\pi}_2 = u_3\,\overline{e}_2 - u_2\,\overline{e}_3 & \overline{\pi}'_2 = u'_3\,\overline{a}_2 - u'_2\,\overline{a}_3 \end{array} \tag{4.50}$$

where the three vectors $\overline{a}_1$, $\overline{a}_2$ and $\overline{a}_3$ are the homologous of $\overline{e}_1$, $\overline{e}_2$, $\overline{e}_3$ for the second view:

$$\overline{a}_1 = \begin{pmatrix} r_{11} \\ r_{12} \\ r_{13} \\ t_1 \end{pmatrix} \quad \overline{a}_2 = \begin{pmatrix} r_{21} \\ r_{22} \\ r_{23} \\ t_2 \end{pmatrix} \quad \overline{a}_3 = \begin{pmatrix} r_{31} \\ r_{32} \\ r_{33} \\ t_3 \end{pmatrix} \tag{4.51}$$

These expressions are derived by substituting $\mathbf{P}$ with $\mathbf{P}'$ and $\overline{u}$ with $\overline{u}'$ in equation 4.42. The three vectors $\overline{a}_1^T$, $\overline{a}_2^T$ and $\overline{a}_3^T$ are then the three row vectors of $\mathbf{P}'$.

The four planes Π_1, Π_2, Π'_1 and Π'_2 must contain the point P in space. In other words, they have to intersect at at least one point. This is equivalent to saying that

their coordinate vectors $\overline{\pi}_1$, $\overline{\pi}_2$, $\overline{\pi}'_1$ and $\overline{\pi}'_2$ are linearly dependent, or equivalently that the matrix $\mathbf{A} = \begin{bmatrix} \overline{\pi}_1 & \overline{\pi}_2 & \overline{\pi}'_1 & \overline{\pi}'_2 \end{bmatrix}$ has rank less than 4, or $\det(\mathbf{A}) = 0$. After expansion of the determinant of $\mathbf{A}$, we obtain:

$$\det(\mathbf{A}) = u_3 u'_3 \begin{bmatrix} u'_1 & u'_2 & u'_3 \end{bmatrix} Q_{12} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = u_3 u'_3 \left(\overline{u}'^T Q_{12} \, \overline{u} \right) \tag{4.52}$$

where the matrix Q_{12} is:

$$\begin{aligned} Q_{12} &= \begin{bmatrix} \det(\overline{e}_2, \overline{e}_3, \overline{a}_2, \overline{a}_3) & \det(\overline{e}_3, \overline{e}_1, \overline{a}_2, \overline{a}_3) & \det(\overline{e}_1, \overline{e}_2, \overline{a}_2, \overline{a}_3) \\ \det(\overline{e}_2, \overline{e}_3, \overline{a}_3, \overline{a}_1) & \det(\overline{e}_3, \overline{e}_1, \overline{a}_3, \overline{a}_1) & \det(\overline{e}_1, \overline{e}_2, \overline{a}_3, \overline{a}_1) \\ \det(\overline{e}_2, \overline{e}_3, \overline{a}_1, \overline{a}_2) & \det(\overline{e}_3, \overline{e}_1, \overline{a}_1, \overline{a}_2) & \det(\overline{e}_1, \overline{e}_2, \overline{a}_1, \overline{a}_2) \end{bmatrix} \\ &= \begin{bmatrix} r_{21}t_3 - r_{31}t_2 & r_{22}t_3 - r_{32}t_2 & r_{23}t_3 - r_{33}t_2 \\ r_{31}t_1 - r_{11}t_3 & r_{32}t_1 - r_{12}t_3 & r_{33}t_1 - r_{13}t_3 \\ r_{11}t_2 - r_{21}t_1 & r_{12}t_2 - r_{22}t_1 & r_{13}t_2 - r_{23}t_1 \end{bmatrix} \\ &= \begin{bmatrix} 0 & t_3 & -t_2 \\ -t_3 & 0 & t_1 \\ t_2 & -t_1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = (T\wedge) \, R \end{aligned} \tag{4.53}$$

We recognize the essential matrix Q between the two views 1 and 2 (eq. 4.23). Then, constraining the four planes to have a common point in space is equivalent to setting $\det(\mathbf{A}) = 0$ or $\overline{u}'^T Q_{12} \, \overline{u} = 0$. This is the well known epipolar constraint for a pair of views (eq. 4.24). Observe that choosing the plane Π_3 or Π'_3 (the homologous of Π_3 on the second view) among the pairs of planes leads to the same final constraint. Only the scalar coefficient in the determinant expression is different (referring to the coefficient $u_3 u'_3$ in (4.52)).

Therefore, the two pairs of planar constraints (four planes) provided by two projective views of a point lead to the same well known epipolar constraint. Let us see now how one can naturally extend that analysis to the case of three views. Generalization to N_v views will then be straightforward.

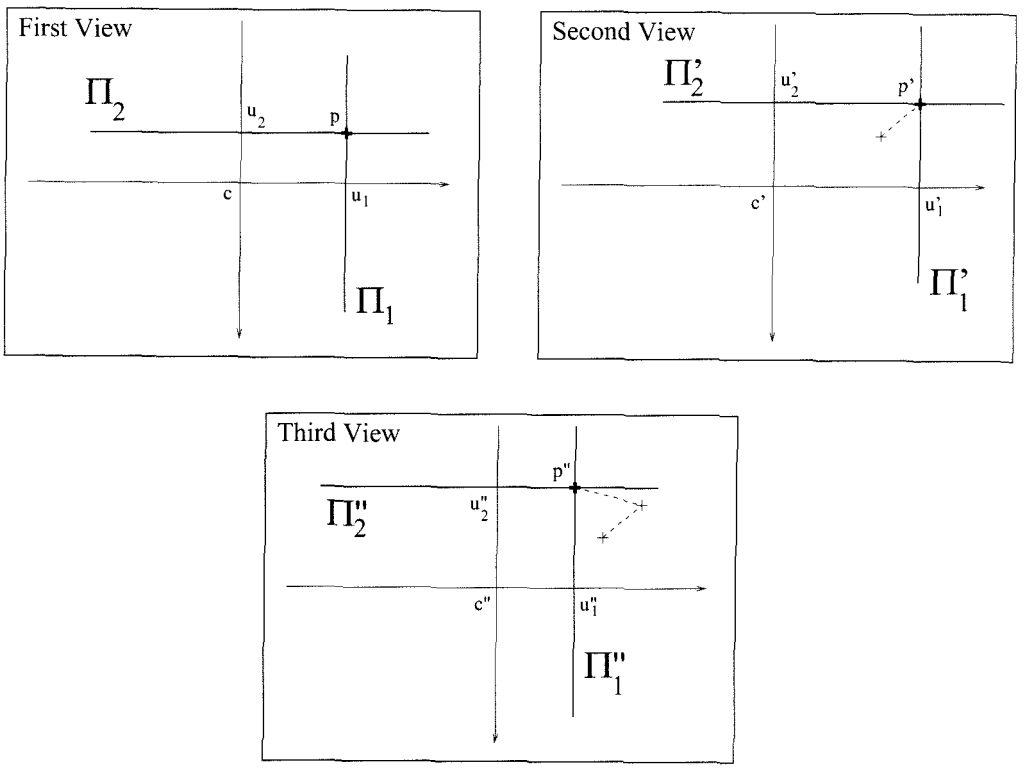


Figure 4.10: The three pairs of planes (Π_1, Π_2), (Π'_1, Π'_2) and (Π''_1, Π''_2) provided by the three projections p, p' and p'' of a single point P in space.The coordinate vectors of p, p' and p'' are $\overline{u} \simeq [u_1 \quad u_2 \quad u_3]^T$, $\overline{u}' \simeq [u'_1 \quad u'_2 \quad u'_3]^T$ and $\overline{u}'' \simeq [u''_1 \quad u''_2 \quad u''_3]^T$ respectively. On the figure, $u_3 = u'_3 = u''_3 = 1$.

From three views – Generalization to N_v views

Assume we have available three projective views p, p' and p'' of the point P, and let $\overline{u}'' \simeq [u''_1 \quad u''_2 \quad u''_3]^T$ be the homogeneous coordinate vector of p''. See figure 4.4 for general notation. Similarly to the case of two views, these observations provide now three pairs of planes (Π_1, Π_2), (Π'_1, Π'_2), (Π''_1, Π''_2) (see fig. 4.10). The respective coordinate vector of those planes in the first camera reference frame $\mathcal{F}$ are:

$$\begin{array}{c|c|c} \text{First view:} & \text{Second view:} & \text{Third view:} \\ \overline{\pi}_1 = u_3\,\overline{e}_1 - u_1\,\overline{e}_3 & \overline{\pi}'_1 = u'_3\,\overline{a}_1 - u'_1\,\overline{a}_3 & \overline{\pi}''_1 = u''_3\,\overline{b}_1 - u''_1\,\overline{b}_3 \\ \overline{\pi}_2 = u_3\,\overline{e}_2 - u_2\,\overline{e}_3 & \overline{\pi}'_2 = u'_3\,\overline{a}_2 - u'_2\,\overline{a}_3 & \overline{\pi}''_2 = u''_3\,\overline{b}_2 - u''_2\,\overline{b}_3 \end{array} \qquad (4.54)$$

where the three vectors $\overline{b}_1$, $\overline{b}_2$ and $\overline{b}_3$ are the homologous of $\overline{a}_1$, $\overline{a}_2$, $\overline{a}_3$ for the third

view:

$$\bar{b}_1 = \begin{pmatrix} s_{11} \\ s_{12} \\ s_{13} \\ v_1 \end{pmatrix} \quad \bar{b}_2 = \begin{pmatrix} s_{21} \\ s_{22} \\ s_{23} \\ v_2 \end{pmatrix} \quad \bar{b}_3 = \begin{pmatrix} s_{31} \\ s_{32} \\ s_{33} \\ v_3 \end{pmatrix} \tag{4.55}$$

Therefore, from the triplet observation $p \leftrightarrow p' \leftrightarrow p''$, we extract the six planes $\Pi_1, \Pi_2, \Pi'_1, \Pi'_2, \Pi''_1, \Pi''_2$ that have P as common point in space.
Define the 4×6 matrix $\mathbf{A}$ as follows:

$$\mathbf{A} = \begin{bmatrix} \overline{\pi}_1 & \overline{\pi}_2 & \overline{\pi}'_1 & \overline{\pi}'_2 & \overline{\pi}''_1 & \overline{\pi}''_2 \end{bmatrix} \tag{4.56}$$

Then, any set of four planes among $\{\Pi_1, \Pi_2, \Pi'_1, \Pi'_2, \Pi''_1, \Pi''_2\}$ must have a common point in space. Consider the non-degenerate case where the three camera positions are not aligned in space and the observed point P does not lie on either line (O_c, O'_c), (O_c, O''_c) or (O'_c, O''_c). Denoting $\overline{X} = [X \;\; Y \;\; Z]^T$ the Euclidean coordinate vector of P in the first camera frame, we have successively for all scalar α:

$$\begin{cases} (O_c, O'_c): & \overline{X} \neq \alpha R^{-1} T \\ (O_c, O''_c): & \overline{X} \neq \alpha S^{-1} V \\ (O'_c, O''_c): & \overline{X} \neq \alpha (S^{-1} V - R^{-1} T) - R^{-1} T \end{cases} \tag{4.57}$$

Under these conditions, any set of three vectors picked in $\{\overline{\pi}_1, \overline{\pi}_2, \overline{\pi}'_1, \overline{\pi}'_2, \overline{\pi}''_1, \overline{\pi}''_2\}$ is linearly independent. In other words, any three columns of A are linearly independent.

In that case, we wish to investigate how many independent conditions we can write to make every set of four vectors picked in $\{\overline{\pi}_1, \overline{\pi}_2, \overline{\pi}'_1, \overline{\pi}'_2, \overline{\pi}''_1, \overline{\pi}''_2\}$ linearly dependent. This is satisfied if the 6 column vectors of $\mathbf{A}$ can be expressed as linear combinations of any 3 basis vectors. Under the condition that we are not in a degenerate case, this condition is fullfiled as soon as any $6 - 3 = 3$ independent sets of linear dependence equations of 4 vectors are established. In other words, we only need to enforce any three 4×4 minors of $\mathbf{A}$ to have zero determinant. This is equivalent to having

that the last three column vectors of $\mathbf{A}$ expressible as linear combinations of the three first columns vectors. Under this condition, any set of 4 vectors picked among $\{\overline{\pi}_1, \overline{\pi}_2, \overline{\pi}'_1, \overline{\pi}'_2, \overline{\pi}''_1, \overline{\pi}''_2\}$ will be linearly dependent. This number of constraints is really $6-3$, where 6 is the number of planes available from the 3 views, and $3 = 4-1$ is the rank to which we want to bring any minor of A. In the case of N_v views, we would have $2N_v - 3$ independent constraints, since $2N_v$ planes would be available (note that in the case of N_v views, we have $6(N_v - 1) - 1 = 6N_v - 7$ motion unknowns).

Now denote by $\mathbf{A}_{ijkl}$ the 4×4 minor constructed by picking the i^{th}, j^{th}, k^{th} and l^{th} column of $\mathbf{A}$ $(1 \leq i < j < k < l \leq 6)$. This constitutes fifteen possible matrices. In the non-degenerate case, we can choose any three 4×4 minors, and force their determinant to zero. Doing so, we force all the others minors to have determinant zero. For example, one can take the 3 minors A_{1234}, A_{1256}, and A_{3456}. This is equivalent to considering the three pairwise of views and write the epipolar constraints (see the previous section dealing with the two views case). Other combinations are possible, such as A_{1235}, A_{1236} and A_{1245}. For that particular choice, the set of constraints is:

$$\begin{cases} \det(\mathbf{A}_{1235}) = \det(\overline{\pi}_1, \overline{\pi}_2, \overline{\pi}'_1, \overline{\pi}''_1) = 0 \\ \det(\mathbf{A}_{1236}) = \det(\overline{\pi}_1, \overline{\pi}_2, \overline{\pi}'_1, \overline{\pi}''_2) = 0 \\ \det(\mathbf{A}_{1245}) = \det(\overline{\pi}_1, \overline{\pi}_2, \overline{\pi}'_2, \overline{\pi}''_1) = 0 \end{cases} \tag{4.58}$$

Observe that all these constraints combine informations from the 3 views (unlike the epipolar constraints which act only on pairs of views), taking the first view as reference. For example, $\det(\mathbf{A}_{1235}) = 0$ means that the two planes Π'_1 and Π''_1 (coming from the two last views) intersect along a line that needs to be coplanar with the optical ray (O_c, p) provided by the first view. This is equivalent to enforcing the point p (on the first view) to lie on the projection of the line of intersection between the planes Π'_1 and Π''_1.

Although we know that three constraints are sufficient (and necessary) for the non-degenerate 3 views case, we may add to these three last constraints a fourth one, $\det(\mathbf{A}_{1246})$ to make them symmetric. That way, the problem is over-constrained and

is expected to handle better some degenerate cases (see [60]). We have for this last equation:

$$\det(\mathbf{A}_{1246}) = \det(\overline{\pi}_1, \overline{\pi}_2, \overline{\pi}'_2, \overline{\pi}''_2) = 0 \tag{4.59}$$

Consider now as total set of constraints the equations given in equations (4.58) and (4.59). Let us now give a geometrical interpretation of those algebraic equations: The two last views generate four lines in space coming from intersections of pairs of planes: $\Lambda_1 = \Pi'_1 \cap \Pi''_1$, $\Lambda_2 = \Pi'_1 \cap \Pi''_2$, $\Lambda_3 = \Pi'_2 \cap \Pi''_1$ and $\Lambda_4 = \Pi'_2 \cap \Pi''_2$, and let λ_1, λ_2, λ_3 and λ_4 the projections of those four space lines on the first camera image plane. The four trilinear constraints are then equivalent to enforcing the observed point p on the initial view to lie on all four images lines λ_1, λ_2, λ_3 and λ_4. Observe that those lines may be thought of as generalized epipolar lines attached to P.

After expansion of the first determinants $\det(A_{1235})$, we get:

$$\begin{aligned}
\det(\mathbf{A}_{1235}) = u_3\{ & \\
& u_3u'_3u''_3 \det(\overline{e}_1, \overline{e}_2, \overline{a}_1, \overline{b}_1) + u_3u'_3u''_1 \det(\overline{e}_1, \overline{e}_2, \overline{b}_3, \overline{a}_1) + \\
& u_3u'_1u''_3 \det(\overline{e}_1, \overline{e}_2, \overline{b}_1, \overline{a}_3) + u_3u'_1u''_1 \det(\overline{e}_1, \overline{e}_2, \overline{a}_3, \overline{b}_3) + \\
& u_2u'_3u''_3 \det(\overline{e}_3, \overline{e}_1, \overline{a}_1, \overline{b}_1) + u_2u'_3u''_1 \det(\overline{e}_3, \overline{e}_1, \overline{b}_3, \overline{a}_1) + \\
& u_2u'_1u''_3 \det(\overline{e}_3, \overline{e}_1, \overline{b}_1, \overline{a}_3) + u_2u'_1u''_1 \det(\overline{e}_3, \overline{e}_1, \overline{a}_3, \overline{b}_3) + \\
& u_1u'_3u''_3 \det(\overline{e}_2, \overline{e}_3, \overline{a}_1, \overline{b}_1) + u_1u'_3u''_1 \det(\overline{e}_2, \overline{e}_3, \overline{b}_3, \overline{a}_1) + \\
& u_1u'_1u''_3 \det(\overline{e}_2, \overline{e}_3, \overline{b}_1, \overline{a}_3) + u_1u'_1u''_1 \det(\overline{e}_2, \overline{e}_3, \overline{a}_3, \overline{b}_3)\}
\end{aligned}$$

or:

$$\det(\mathbf{A}_{1235}) = u_3 \sum_{i=1}^{3} u_i \{u'_1u''_1\mathcal{T}_{i33} - u'_3u''_1\mathcal{T}_{i13} - u'_1u''_3\mathcal{T}_{i31} + u'_3u''_3\mathcal{T}_{i11}\} \tag{4.60}$$

where we recognize the trifocal tensor $\mathcal{T}_{ijk} = r_{ij}u_k - t_j s_{ki}$ for $i, j, k = 1, \ldots, 3$ previously introduced for line correspondence (eq. 4.34). Proceeding similarly with the

other determinants, we obtain the following set of trilinear expressions:

$$\begin{cases} \det(\mathbf{A}_{1235}) = u_3 \sum_{i=1}^{3} u_i \{u'_1 u''_1 \mathcal{T}_{i33} - u'_3 u''_1 \mathcal{T}_{i13} - u'_1 u''_3 \mathcal{T}_{i31} + u'_3 u''_3 \mathcal{T}_{i11}\} \\ \det(\mathbf{A}_{1236}) = u_3 \sum_{i=1}^{3} u_i \{u'_1 u''_2 \mathcal{T}_{i33} - u'_3 u''_2 \mathcal{T}_{i13} - u'_1 u''_3 \mathcal{T}_{i32} + u'_3 u''_3 \mathcal{T}_{i12}\} \\ \det(\mathbf{A}_{1245}) = u_3 \sum_{i=1}^{3} u_i \{u'_2 u''_1 \mathcal{T}_{i33} - u'_3 u''_1 \mathcal{T}_{i23} - u'_2 u''_3 \mathcal{T}_{i31} + u'_3 u''_3 \mathcal{T}_{i21}\} \\ \det(\mathbf{A}_{1246}) = u_3 \sum_{i=1}^{3} u_i \{u'_2 u''_2 \mathcal{T}_{i33} - u'_3 u''_2 \mathcal{T}_{i23} - u'_2 u''_3 \mathcal{T}_{i32} + u'_3 u''_3 \mathcal{T}_{i22}\} \end{cases} \tag{4.61}$$

Therefore, setting the determinants to zero lead to the following set of algebraic constraints:

$$\begin{cases} \sum_{i=1}^{3} u_i \{u'_1 u''_1 \mathcal{T}_{i33} - u'_3 u''_1 \mathcal{T}_{i13} - u'_1 u''_3 \mathcal{T}_{i31} + u'_3 u''_3 \mathcal{T}_{i11}\} = 0 \\ \sum_{i=1}^{3} u_i \{u'_1 u''_2 \mathcal{T}_{i33} - u'_3 u''_2 \mathcal{T}_{i13} - u'_1 u''_3 \mathcal{T}_{i32} + u'_3 u''_3 \mathcal{T}_{i12}\} = 0 \\ \sum_{i=1}^{3} u_i \{u'_2 u''_1 \mathcal{T}_{i33} - u'_3 u''_1 \mathcal{T}_{i23} - u'_2 u''_3 \mathcal{T}_{i31} + u'_3 u''_3 \mathcal{T}_{i21}\} = 0 \\ \sum_{i=1}^{3} u_i \{u'_2 u''_2 \mathcal{T}_{i33} - u'_3 u''_2 \mathcal{T}_{i23} - u'_2 u''_3 \mathcal{T}_{i32} + u'_3 u''_3 \mathcal{T}_{i22}\} = 0 \end{cases} \tag{4.62}$$

These are identical to the ones obtained by Hartley in [54] and Shashua [56] and called trilinear constraints. In their work however treat them as purely algebraic constraints (constraints on the point locations on the image plane) in an uncalibrated scenario. In that sense they show that they are *algebraically* independent (in fact, all other combinations of algebraic constraint are also independent). In our case, we keep a geometrical interpretation of those equations (constraints on the motion parameters, assuming point coordinates given) and show that, in that sense, only three of them are independent. Indeed, we recall that there are exactly three independent constraints among the ones listed in (4.62).

Observe that this list of four trilinear constraints may also be written:

$$\begin{cases} \langle \overline{u}, \overline{\lambda}^1 \rangle = 0 \\ \langle \overline{u}, \overline{\lambda}^2 \rangle = 0 \\ \langle \overline{u}, \overline{\lambda}^3 \rangle = 0 \\ \langle \overline{u}, \overline{\lambda}^4 \rangle = 0 \end{cases} \tag{4.63}$$

where $\overline{\lambda}^k \simeq [\lambda_1^k \;\; \lambda_2^k \;\; \lambda_3^k]^T$ $(k = 1, \ldots, 4)$ is the homogeneous coordinate vector of the

transferred line λ_k on the first image plane:

$$\begin{cases} \lambda_i^1 = u'_1 u''_1 \mathcal{T}_{i33} - u'_3 u''_1 \mathcal{T}_{i13} - u'_1 u''_3 \mathcal{T}_{i31} + u'_3 u''_3 \mathcal{T}_{i11} \\ \lambda_i^2 = u'_1 u''_2 \mathcal{T}_{i33} - u'_3 u''_2 \mathcal{T}_{i13} - u'_1 u''_3 \mathcal{T}_{i32} + u'_3 u''_3 \mathcal{T}_{i12} \\ \lambda_i^3 = u'_2 u''_1 \mathcal{T}_{i33} - u'_3 u''_1 \mathcal{T}_{i23} - u'_2 u''_3 \mathcal{T}_{i31} + u'_3 u''_3 \mathcal{T}_{i21} \\ \lambda_i^4 = u'_2 u''_2 \mathcal{T}_{i33} - u'_3 u''_2 \mathcal{T}_{i23} - u'_2 u''_3 \mathcal{T}_{i32} + u'_3 u''_3 \mathcal{T}_{i22} \end{cases} \tag{4.64}$$

Observe that the four equations (4.64) provides us with closed-form expression for the transferred line coordinate vectors as a function of the motion parameters (embedded in the tensor $\mathcal{T}_{ijk}$), similarly to the epipolar line equations 4.22 and 4.23 in the two view case. As formulated in equations 4.62 or 4.63, the trilinear constraints have also been called *trilinearities* [61, 62].

In this new derivation, we not only give a geometric understanding of these quantities, but we also precisely know how many independent entities can be extracted (three). This provides also ways to extend the analysis to N_v views. Indeed, as we mentioned earlier in this section, likewise the 2 and 3 views cases, one single point in space observed on N_v views provides $2N_v$ planes (or 4-vectors) that have to intersect at this point. That means that every combination of 4 vectors have to be linearly dependent. Assuming that the observation and the camera geometry is not degenerate, making any $2N_v - 3$ combinations of four vectors linearly dependent are enough (for $N_v = 2$, there is only 1, the epipolar constraint, for $N_v = 3$, there are 3). Taking more combinations helps to handle degenerate cases and effect of noise.

In the case of $N_v = 4$ views, one can write for a point observation 5 independent constraints that can be again picked among pairwise epipolar constraints (for example, between views 1-2, 2-3, 1-3, 2-4 and 3-4), or the trilinear constraints (taking two planes on one view, and the two others on 2 different views). A new type of constraint can be also considered by taking one vector on each view (which makes four 4-vectors) and forcing the induced 4×4 matrix to have zero determinant. This leads to the quadrilinear constraints first mentioned by Triggs [63] and more recently by Hartley [64] and Heyden [65].

If more than three views are used in the basic observation cell, the dimension of the motion space is also larger. Indeed, there are $N_v - 1$ rigid motions in a set of N_v. This leads to estimating $6(N_v-1)-1$ motion unknowns (the overall scene scale cannot be recovered). Under these conditions at least $N_p = (6N_v - 7)/(2N_v - 3)$ points are necessary to reconstruct in the calibrated case the complete camera geometry. For $N_v = 2$, $N_p = 5$ points are enough (well known), and for any number of views larger than 2, at least $N_p = 4$ points are necessary. Notice that in the limit, when the number of views tends to infinity, three points is the minimum required.

It is important to notice that the set of trilinearity equations (4.62) constitutes only one particular choice of motion constraints. The previous derivation was particularly useful for giving new interpretation of them. At this point, one can still pick the 3 epipolar constraints as set of motion constraints for three views. That way, in the non-degenerate case, considering any number of views as successive overlapping triplets of views allows us to extract the maximum information in terms of motion constraints. Indeed, if the sequence consists of a stream a N_v images, applying for example the 3 epipolar constraints to each successive overlapping triplets of views exactly leads to $2N_v - 3$ constraints. This is however not the case while considering only successive pairs of views (that way, only $N_v - 1$ scalar constraints would be retrievable).

In conclusion, three views is the fundamental observation cell for extracting all the geometrical constraints on the motion parameter for both point or line observation. The only advantage of processing the complete set of frames at once (in a batch mode) would be that one can take larger processing time baselines for the observation (by choosing constraints acting on far apart frames) and therefore make the motion estimation more accurate (in the presence of noise). This would also help to handle possible degenerate cases.

4.4 Long sequence processing - Experimental results

In order to capture an entire scene, two or three images are often not enough. In most cases, a long sequence of images will be necessary for an complete scene coverage. We choose to describe the implementation details of motion and structure estimation in the context of two long sequence experiments. The first one is a reconstruction of a rock from an orbital image sequence, and the second one is a reconstruction of a corridor in a long navigation sequence.

4.4.1 Rock experiment

Experimental setup:

Figure 4.11 shows a few images of a complete sequence acquired by a calibrated camera while rotating around a rock. The complete sequence is 226 frames long, each image being of size 640×480. For acquiring the sequence, the rock was positioned on a turn table that was rotated by 2 degrees between consecutive frames. Consequently, throughout the whole sequence, the camera made a turn and a quarter around the rock (considering the relative motion of the camera with respect to the rock). The first processing step consists of tracking point features on the images using multi-resolution implementation of the standard optical flow algorithm from Tomasi and Kanade [32, 33, 34, 35]. This method is applicable since the image disparity between consecutive frames is relatively small (maximal pixel displacement of about 10 pixels). Figure 4.12 shows the results of tracking on the six images shown on figure 4.11. These feature coordinates are the input data for our motion and structure estimator. The reader may visit our web page at `http://www.vision.caltech.edu/bouguetj/Motion/comet.html` to better visualize the input data (available in the form of movies).

Motion and structure estimation:

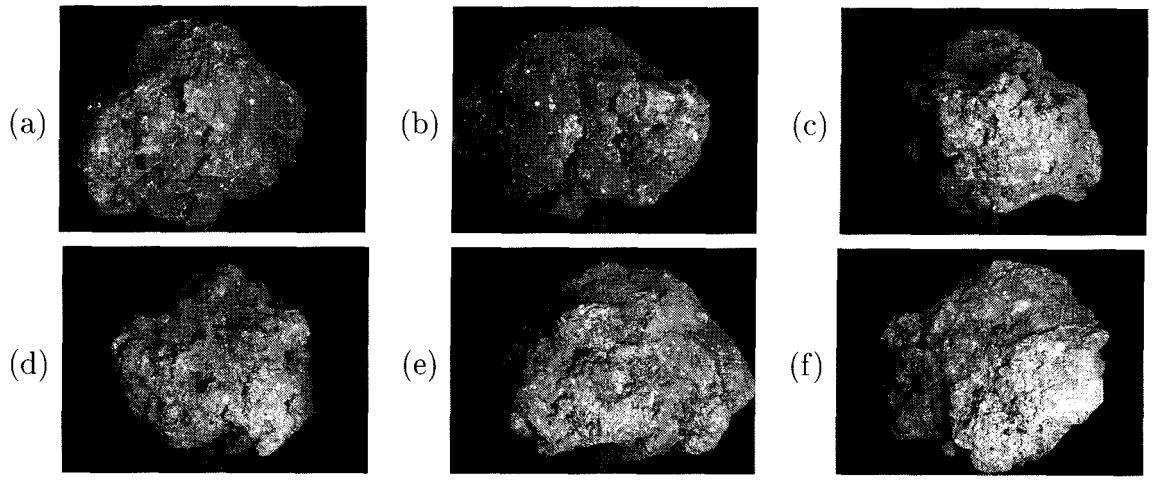


Figure 4.11: Rock experiment: Six images among the 226 images constituting the sequence. Between two images shown on this figure, the turn table rotation angle was 60 degrees (images taken at position angles 0, 60, 120, 180, 240 and 300 degrees).

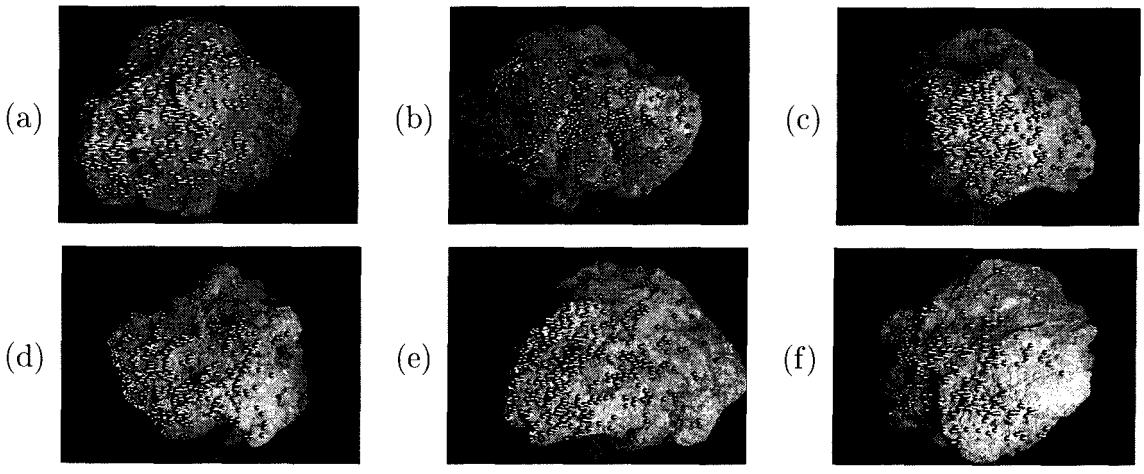


Figure 4.12: Rock experiment: Figure (a) shows the total set of feature points tracked between the first image and the second one (a total of 431 features). The other figures show similar tracking results on the 5 other images shown on figure 4.11. Notice that the total number of tracked features varies: on the five remaining images, there are respectively 390, 405, 415, 423 and 295 tracked points. The tracking is processed with an accuracy of approximately 0.2 pixel.

We separate the overall reconstruction process into two steps: first motion (and camera positions) estimation, and then 3D structure reconstruction.

In the previous sections 4.2 and 4.3, we developed the fundamental theoretical ground for estimating motion between a set of two or three perspective views of a rigid scene. Using that same formalism, we also extended the analysis to any number of frames. We choose to process the sequence by overlapping sets of three frames for enabling relative scale estimation (recall that the relative scale is part of the motion unknowns that can be estimated).

Observe that no feature point is tracked throughout the entire sequence. Point feature appear and disappear from the visual field due to natural occlusions existing in the scene, as well as changes in illumination. Experimentally, the average life-time of a feature is between 25 to 30 frames (meaning a differential rotation angle of 50 to 60 degrees between appearance and disappearance). This constitutes an important experimental observation since the motion estimation algorithm cannot be applied on any set of three views picked within the sequence. The views need to be sufficiently "close" to each other in order to share a sufficient number of feature points to allow for motion estimation (for example, one cannot use frames taken at angles 0, 90 and 180 degrees). Therefore, one possible strategy is processing the sequence by overlapping triplets of consecutive views. However, that strategy may not be the optimal one given the small parallax between consecutive views. Indeed, as the motion parallax decreases, motion parameters are more and more unreliable to estimate. In the limit, if the translation vector is zero between consecutive views, then the full motion model cannot be estimated (notice that a reduced motion model may be observable, but that observation goes beyond the scope of this work). Consequently, it is beneficial to process motion using a non unitary baseline. In other words, a number of frames in the original sequence may be skipped between the three frames used as elementary observation (for example processing motion between frames 1, 5 and 9 would mean skipping frames 2, 3 , 4, 6, 7 and 8). In this experiment, we choose a baseline of baseline $= 4$ frames for motion estimation. This means that within every triplet of views, $2 * 3 = 6$ frames are skipped. Observe that the final goal is reconstructing the

three-dimensional structure of the rock. For that purpose, camera positions are the real quantities that one needs to estimate (necessary for 3D triangulation - see section 4.1.2). Instantaneous motion is only a way to access to position (or trajectory). The step of computing trajectory from motion is done by integration (or motion concatenation). Therefore, all errors on the instantaneous motion are transferred to position information via integration *with no hope for recovery* (since the whole process is done in "open-loop"). Therefore, it is absolutely crucial to limit the errors on the motion parameters (especially the bias on the motion estimator), and limit the number of integration steps. Consequently, it is also beneficial to choose the largest possible baseline between observation frames. Here, we experimentally picked a fixed baseline of 4 frames (i.e., an elementary rotation of 8 degrees of the turn table between two consecutive frames). The average number of features used for motion estimation is 255 (i.e., average number of point features present over at least $2 * 4 + 1 = 9$ consecutive images of the initial sequence). One other important component of the motion estimation stage is the rejection of false tracks. Indeed, it is quite well known that motion estimation is very sensitive to feature outliers [45, 51]. Therefore, at every step, every point that does not closely satisfy the algebraic motion constraints equations (4.62) are rejected (using the motion parameters of the previous step). All the remaining points are then used for motion update (through gradient descent). Experimentally, this segmentation step is extremely crucial [45, 51] for avoiding divergence of the motion estimator. Notice however that a certain amount of smoothness in motion is required for validating such a approach . Indeed, at every estimation step, the instantaneous motion is implicitly assumed to be similar to that of the previous step.

For addressing motion estimation accuracies, other that looking at all the motion parameters, let us focus on the one that is most representative: the instantaneous rotation angle. The ground truth for instantaneous rotation angle is $4 * 2 = 8$ degrees (4 for the frame baseline, and 2 for the elementary rotational motion of the turn table between consecutive frames of the original sequence). After running the motion estimator, this angle was estimated to 8.034 degrees (on average) with an error of 0.057

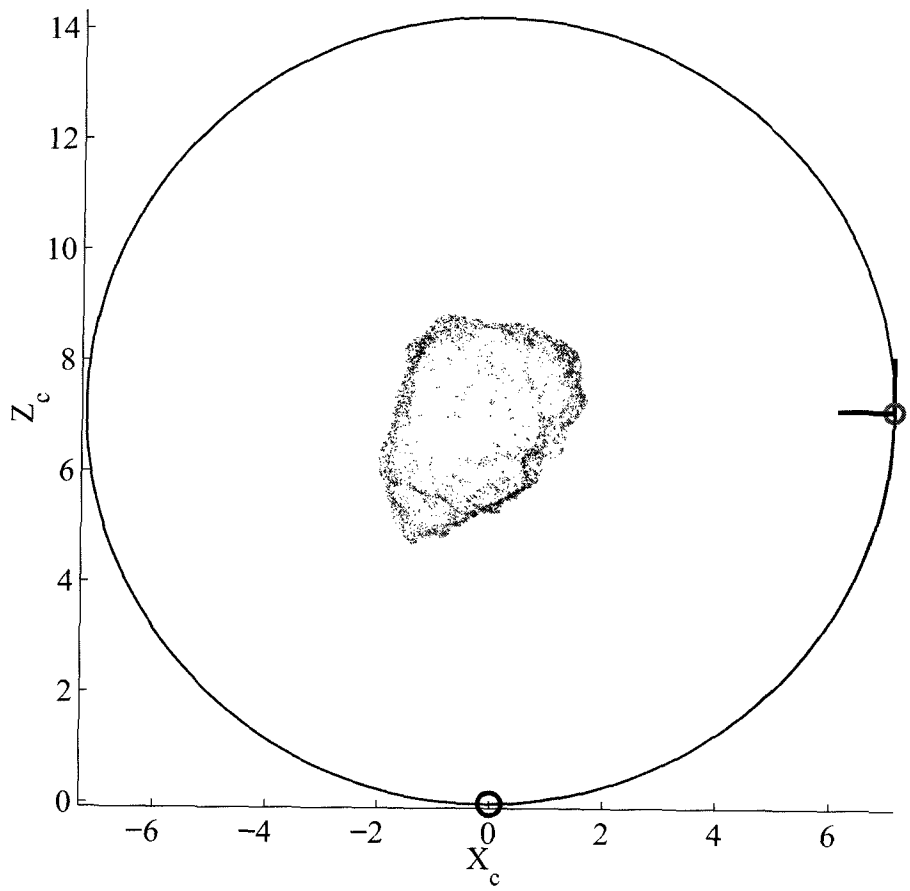


Figure 4.13: Rock experiment: Top view of the trajectory and 3D structure reconstructions. The 3D structure contains 5818 points.

degrees in standard deviation. The peak values for this angle throughout the sequence were 7.934 and 8.146 degrees. The difference between the mean instantaneous rotation angle (8.034 degrees) and the ground truth (8 degrees) is called the "bias rotation error" (equal to 0.034 degrees). We will see later that it is essential to have this bias as small as possible for accurate trajectory reconstruction.

Figures 4.13 and 4.14 show a top view and a side view of the overall reconstructed camera trajectory together with the estimated 3D structure of the rock. The 3D structure was estimated using the tools developed in section 4.1.2: every feature point is triangulated using all the views on which it is observed (all the optical rays are used). That is possible once all the camera positions are computed by integrating all the elementary instantaneous motions. Observe on that figure how well the

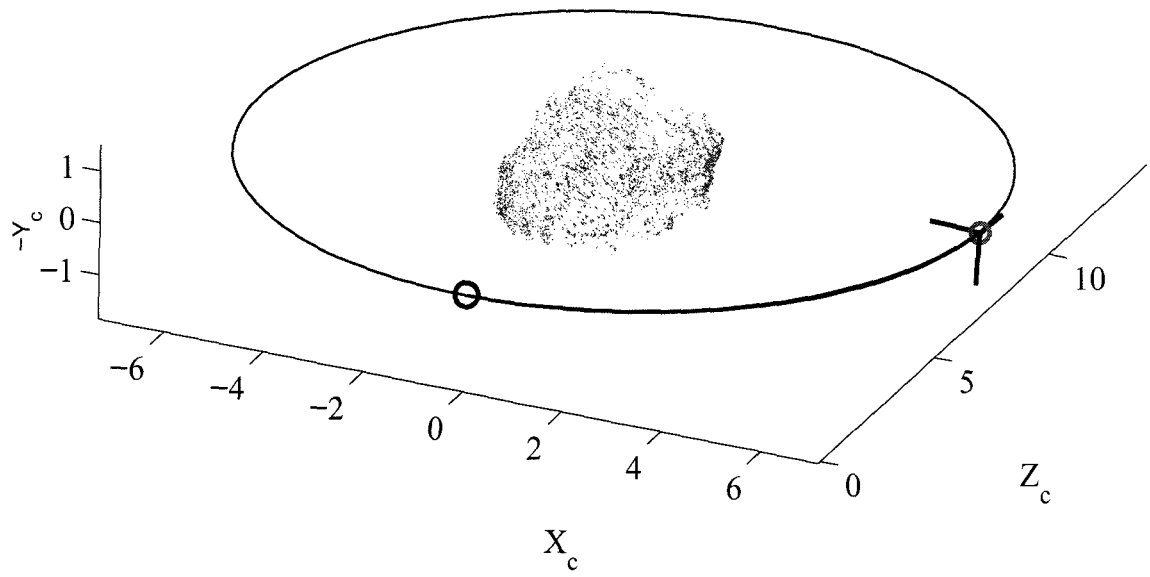


Figure 4.14: Rock experiment: Side view of the trajectory and 3D structure reconstructions. The 3D structure contains 5818 points.

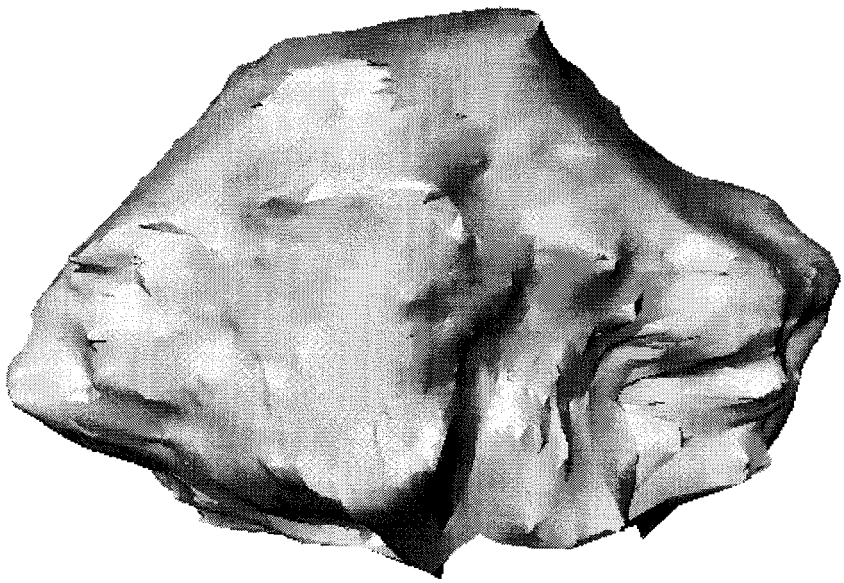


Figure 4.15: Rock experiment: One view of the triangulated mesh (5732 vertices and 11436 triangles).

trajectory closes onto itself (the two trajectory curves on the overlapping segment are indistinguishable one from the other). The final absolute orientation error (about the vertical Y_c axis) is 1.84 degrees. Notice that this final orientation error directly corresponds to the "bias rotation error" (0.034 degrees) integrated over the entire motion length (55 motion steps): $1.84 \approx 0.034 * 55$. Therefore, it is absolutely crucial to design an unbiased motion estimator. There might exist some motion algorithms that generates less bias than the one we are using (from the trilinearities (4.62)). This is an fundemental subject for future research.

After one complete turn, the estimated camera position differs from the true one by 0.17 units of translations (units of the plots of figures 4.13 and 4.14)

Notice that the scale was appropriately propagated throughout the sequence: no significantly shrinking or expansion is noticeable after a complete turn around the rock. Of course, if we were to process a lot more than one turn in an "open loop" fashion (a longer sequence), then scale divergence would start appearing. That is an intrinsic behavior of all sequential motion estimators operating in open loop. One way to avoid scale divergence would be to automatically detect trajectory closing, and enforce it at every turn. This process could be achieved either based on motion (by recognizing that the computed camera position overlaps with the original camera position), or based on structure observation (by recognizing some areas in the scene that have been already visited and match feature points). Both strategies could also be used in conjunction for better results. That is an essential part of future work that ought to be carried out for designing a full visual based navigation system. In that present experiment, we illustrate the fact that at least a full orbital turn around an compact object (that could be a comet) can be sufficiently well estimated in order to achieve acceptable 3D reconstruction qualities.

Observe that the structure can only be reconstructed up to an overall scale. In this experiment, the final reconstructed scene was scaled to fit in a box of size $4 \times 4 \times 4$ units of translation (at the frame baseline 4). In this unit measure, the surface errors are estimated to .2, which corresponds to a relative reconstruction error of 5%.

Figure 4.15 shows one view of the triangulated mesh of the reconstructed rock.

This mesh was obtained by connecting the point features present in the structure using standard Delaunay triangulation [66].

All data and results are available online in the form of movies and VRML (or Open Inventor) meshes (`http://www.vision.caltech.edu/bouguetj/Motion/comet.html`).

4.4.2 Corridor experiment

The same reconstruction scheme was applied on a long indoor navigation sequence. Figure 4.16 shows 6 images among the 3985 that constitute the entire sequence. The sequence was acquired by a camera attached on a rolling cart, manually pushed by two operators in a corridor (with almost uniform speed). Observe from the images that the corridor walls were previously covered with black and white sheets for generating texture. Without these additional sheets, some portions of the sequence would have been totally featureless (in such a case, no motion and structure processing is possible).

Similarly to the previous experiment, we first track point features throughout the image sequence. The average number of points detected and tracked on each image is 353. Tracking results are illustrated on figure 4.17.

The motion (and overall camera trajectory) is computed using a temporal baseline of 16 images. That basically means that the first three images used for motion estimation are the first, the 17^{th} and the 33^{rd} (recall that motion estimation is done on the basis of overlapping triplets of frames). The average number of point feature used per triplet of views is 128.

Figure 4.18 shows a top view of the reconstructed camera trajectory, together with the reconstructed corridor. In that present experiment, we noticed a large scale divergence while trying to propagate the scale factor from triplet of views to triplet of views (from beginning till end of the sequence, the computed norm translation varies from one to approximately two). We believe that the reason for that noticeable divergence is the large total number of motion steps that need to be concatenated (249 instead of 55 for the rock experiment). Although the ground truth for camera

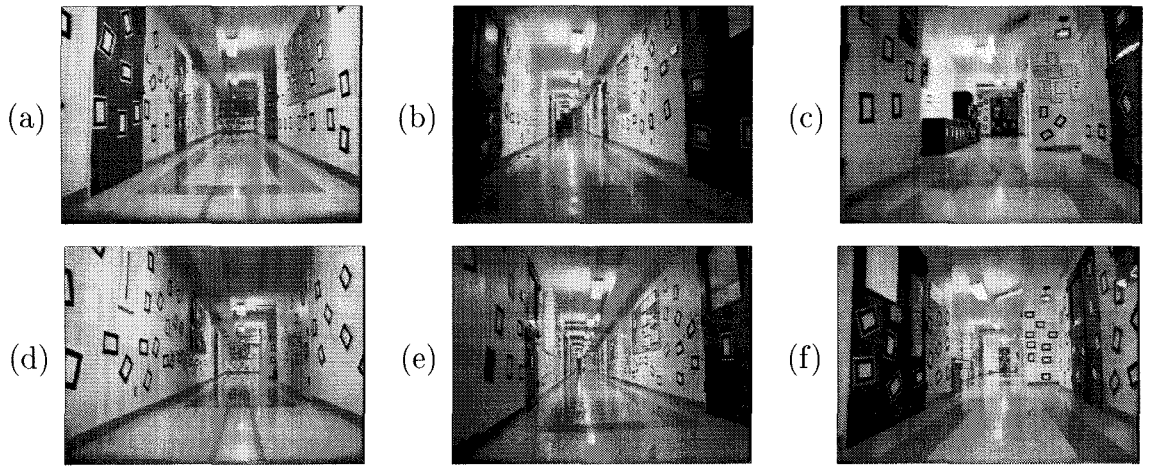


Figure 4.16: Corridor experiment: Six images among the 3985 images constituting the entire sequence. Each image is 640 × 480 interlaced. After de-interlacing, they are 640 × 240.

motion and trajectory was not known in that experiment, it is clear that the real displacement speed did not increase by a factor of two between the beginning and the end of the sequence. For that reason, we decided to compute camera trajectory while enforcing unit length translation throughout the entire sequence (notice that this is not strictly true either, since the real motion was not exactly uniform in speed). Figure 4.18 shows the motion and structure results assuming uniform translational speed. A side view of the same constructed scene is shown on figure 4.19. Notice that this figure exhibits global trajectory estimation errors. Indeed, on that figure, the camera appears to have climbed up a hill (by approximately 15 units of translation), while the real trajectory was planar. These errors are due to the fact that a full 6 DOF motion model was used here allowing for any type of motion is space. In practical planar navigation scenarios, it is possible to avoid that problem by choosing a purely planar motion model, and thereby enforcing the resulting reconstructed trajectory to lie on a plane. This non-planarity on this current reconstruction is however quite minimal compared to the overall trajectory length (approximately 249). In that sense, the relative non planarity error is approximately $15/249 \approx 6\%$.

All data and results are available online at:

`http://www.vision.caltech.edu/bouguetj/Motion/navigation.html`.

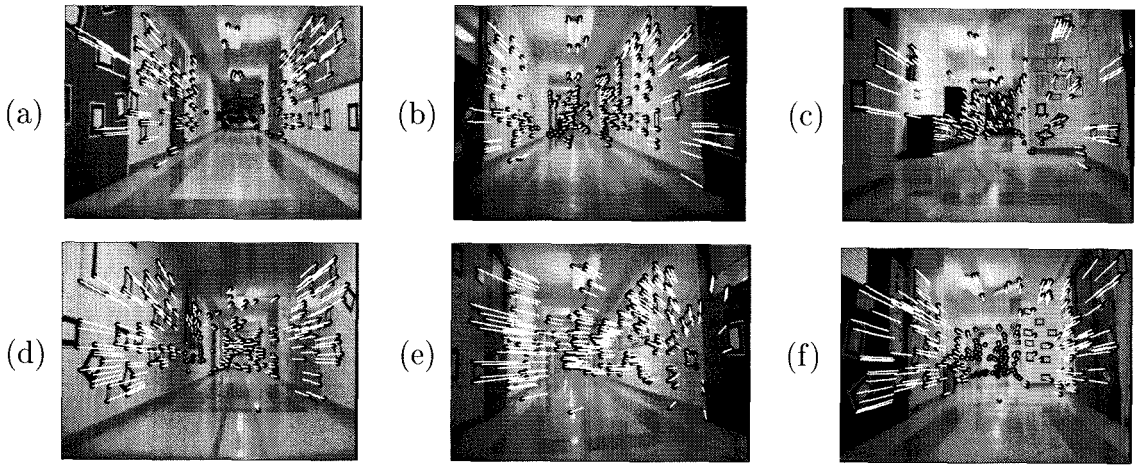


Figure 4.17: Corridor experiment: Tracking results on the six images of figure 4.16. The baseline to display tracking results is 16 images (i.e., (a) is the computed optical flow between the first and the 17^{th} image).

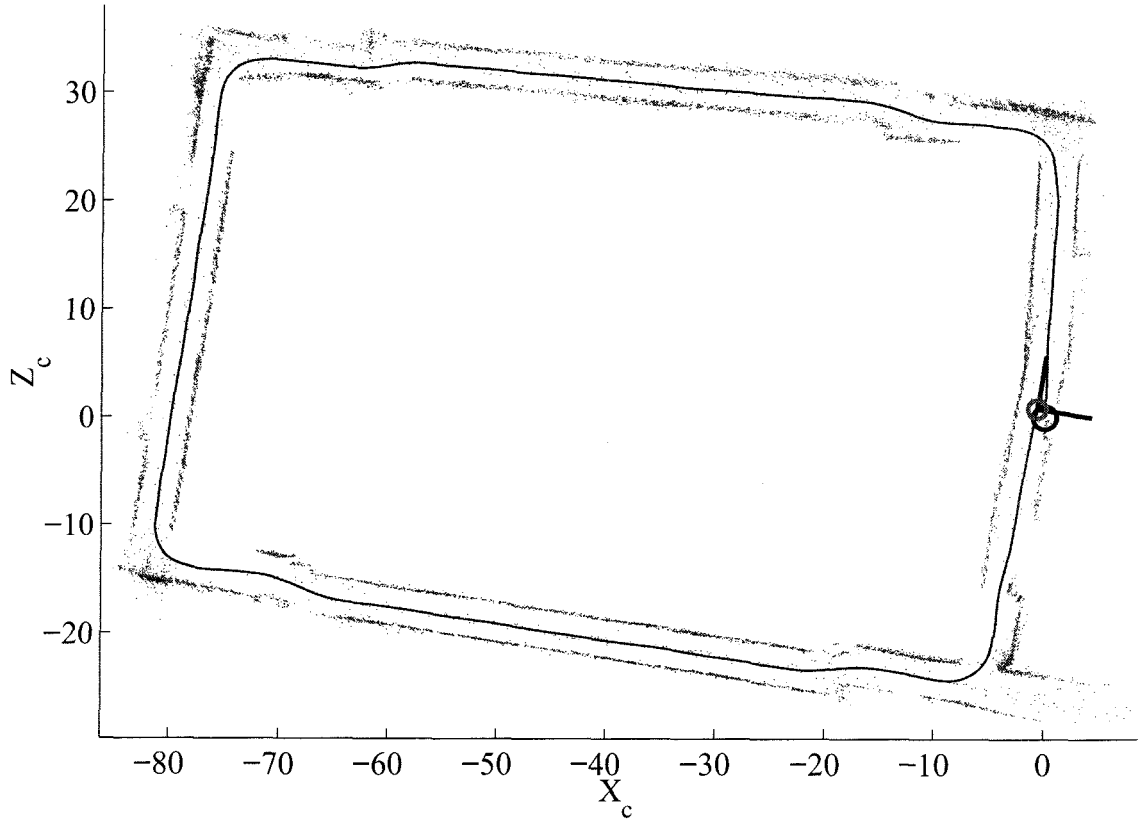


Figure 4.18: Corridor experiment: Top view of the camera trajectory and 3D structure reconstructions. There are 14850 points in the structure.

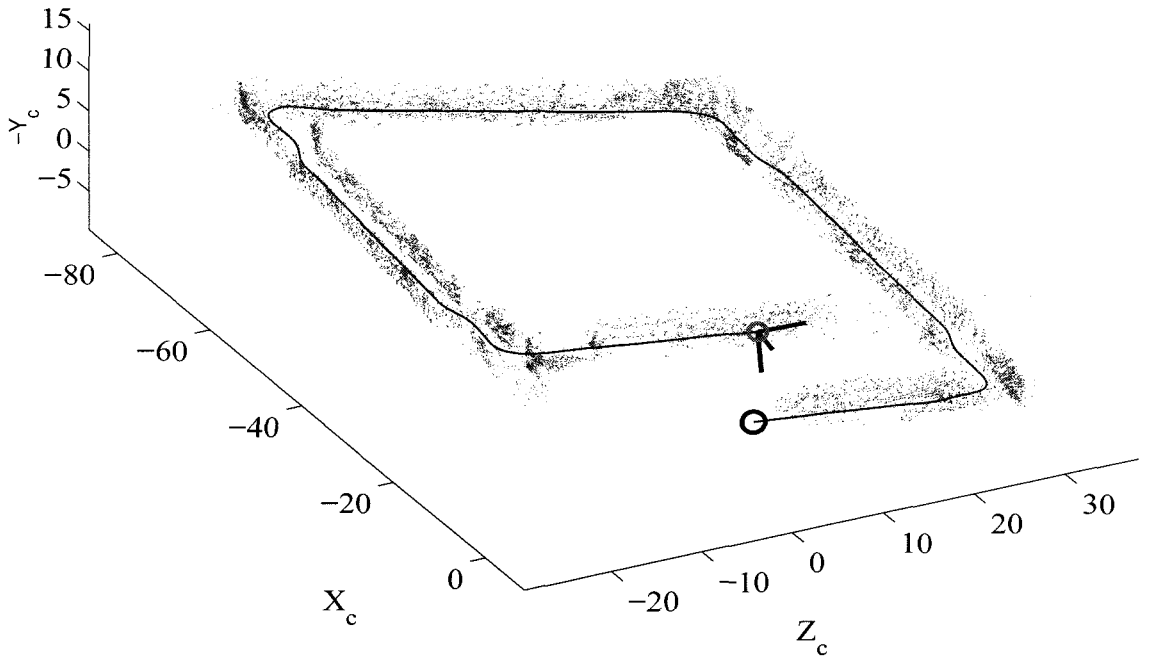


Figure 4.19: Corridor experiment: Side view of the camera trajectory and 3D structure reconstructions. There are 14850 points in the structure.

4.5 Conclusions

We have presented in this chapter the fundamental tools for 3D modeling based on passive visual input.

The main advantage of passive techniques for scene reconstruction is that no other device besides a single camera is needed. In addition, by moving the camera all around the object to acquire, it is possible to achieve a globally consistent 3D reconstruction. This constitutes another significant advantage compared to standard structured lighting technologies that very often require registration (or alignment) of several individual 3D scans in order to achieve complete models.

Finally, since camera trajectory is computed together with the three-dimensional object structure, the camera does not have to be attached to any computer-controlled motion stages (unlike most active scanning techniques that require highly accurate motorized transports).

However, in order to apply passive techniques for 3D modeling, the object to model is required to be sufficiently textured in order to allow for image feature detection and tracking. This is the main limitation of passive technologies: if the object of interest is textured-less or contains sparse texture, then no (or very few) point or line features may be extracted from the acquired images. In that case, it is impossible to achieve a dense object surface reconstruction.

The following chapters 5 and 6 propose alternative methods for modeling possibly textured-less objects using active (or structured) lighting approaches.

Chapter 5 Grayscale structured lighting

5.1 Introduction and motivation

In the previous chapter, we presented the basic tools for reconstructing the three-dimensional shape of objects when using the natural texture present in the scene for extracting elementary geometrical descriptors such as points and lines. This is also known as passive techniques for 3D reconstruction. We also showed that 3D reconstruction is achievable even when the overall camera motion is not previously known.

Unfortunately, passive techniques can only be used when the scene contains a sufficient amount of texture. For modeling textured-less objects, a different class of techniques can be applied: *active (or structured) lighting techniques.*

Structured lighting is based on projecting light patterns in the scene, and infer its three-dimensional shape from the images acquired by a camera placed at a different location in space. The major advantage of this category of approaches is that the complete 3D acquisition system can be made fully automatic, and very robust with respect to changes in texture in the scene.

There exists many different versions of structured lighting methods, mainly depending on the choice of projecting device (laser projector, LCD projector, sets of mirrors) and projected patterns (points, lines, stripes, circles, ...). There exists a very large history of past work on active lighting techniques for 3D scanning (dating back to the early 80's). We propose here to cite a limited list of references (papers and books) that contain most of the relevant work on the subject [6, 7, 67, 68, 69, 70, 71, 4, 72, 9]. Among those references, the book by Klette, Schluns and Koschan [9] provides a very complete overview of most successful active lighting techniques when using laser-based or LCD-based projecting devices, with one or more cameras. In particular, when a LCD projector is utilized, standard active techniques are mostly based on projecting

binary stripe patterns consisting of a succession of vertical black and white stripes across the image at different resolutions [7, 67]. The new technique that we describe in that chapter differs form that binary stripe technique in the nature of the patterns, as well as in the type of processing [68]. It consists of projecting a sequence of periodic *gray-scale patterns* (sinusoidal wave along the horizontal direction, uniform along the vertical direction). A few sample patterns may be found on figure 5.3. One may notice that all the patterns shown on this figure are several phase shifted versions of a unique sinusoidal waveform. As a consequence, every physical point in the scene is illuminated by a light intensity that is periodic in time with a phase linearly depending upon its horizontal coordinate in the projector reference frame. Therefore, extracting the phase of that intensity function (in time) directly leads to an estimate of the horizontal coordinate in the projector reference frame. Once this is done, correspondence between projector image and camera image is established (in a dense way), and 3D shape is reconstructed using geometrical triangulation.

We first describe in section 5.2 the final 3D reconstruction stage leading to the final 3D shape of the scene (geometrical stage). In the following section 5.3, we explore the details of extraction of the image features necessary to perform the geometrical reconstruction (dense correspondence problem). This is the main novelty of our method. Experimental results are presented in section 5.4 followed by a complete noise sensitivity analysis in section 5.5, and some conclusions in section 5.6.

5.2 Depth measurement

Figure 5.1 gives a description of the general setup used for 3D scanning. On this figure, the scene is represented by an object that is faced by the two main devices used for scanning: a camera and a light projector.

The projector projects a series of 2D patterns with varying brightness profile along the horizontal direction (x_p) and uniform brightness profile along the vertical direction (y_p). Figure 5.3 shows a few sample patterns. Consequently, only the horizontal coordinate (x_p) in the projector reference frame is directly observable (in

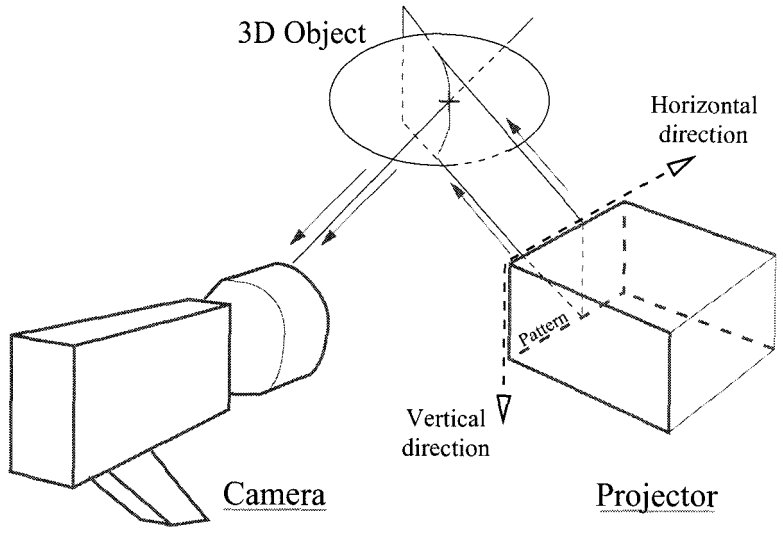


Figure 5.1: General setup: The camera and the projector are facing the scene consisting of one or more objects. The projected patterns are uniform along the vertical direction and vary sinusoidally along the horizontal direction. Then, in the projector image, two points at the same vertical locations are indistinguishable. Equivalently, we can think of the system as follows: the projector projects vertical planes defined by their horizontal position (similar to standard vertical stripe methods).

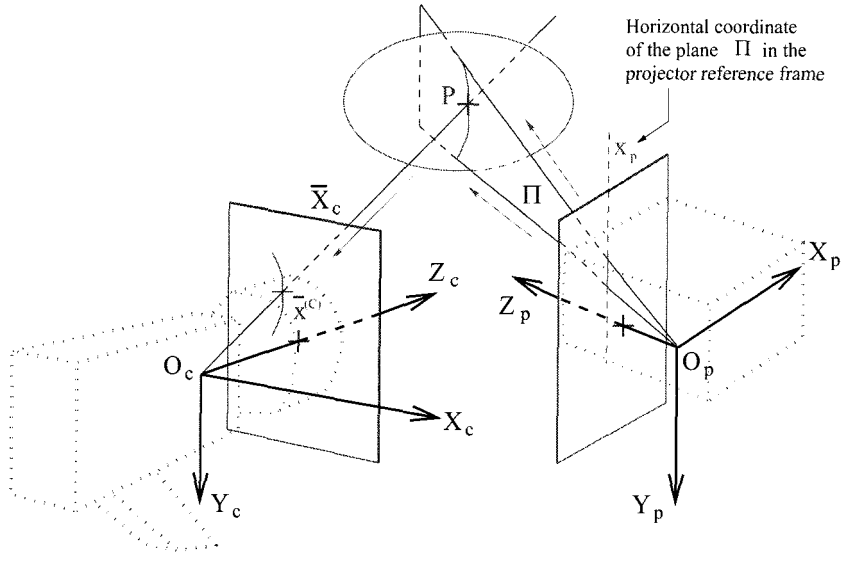


Figure 5.2: Triangulation stage: The 3D coordinates $\overline{X}_c$ of a point P in the scene may be computed from its pixel coordinates $\overline{x}_c$ on the camera image, and its horizontal projector coordinate x_p. The triangulation operation consists then of intersecting the plane Π with the optical ray $(O_c, \overline{x}_c)$. This may be done only if the relative spatial position of the projector with respect to the camera, is known (from calibration).

the projector image, there is no disparity between points along vertical band). In other words, one can only directly identify which vertical plane Π illuminates a given point $\overline{x}_c$ observed on the camera image, not its vertical position along that plane. From the camera however, the full image of the scene is observed, which means that each pixel $\overline{x}_c = [x_c \;\; y_c]^T$ in the image is the projection of a point P in the scene that lies on the optical ray $(O_c, \overline{x}_c)$ (see figure 5.2). Therefore, once its associated projector coordinate x_p is identified, the point P may be localized by intersecting the projector plane Π with the corresponding optical ray $(O_c, \overline{x}_c)$. This final 3D recovery stage is called *triangulation*. On figure 5.2, the coordinate vector of P in the camera frame is denoted $\overline{X}_c$. A method for establishing correspondence between image coordinates and projector coordinates $(\overline{x}_c \leftrightarrow x_p)$ is presented in the next section 5.3.

For now, let us go through the derivation of the triangulation operator (denoted Δ) that returns the coordinate vector of P in the camera reference frame $\overline{X}_c$ from its image and projector coordinates $\overline{x}_c$ and x_p:

$$\overline{X}_c = \Delta(\overline{x}_c, x_p) \tag{5.1}$$

Let $\overline{X}_c = [\; X_c \;\; Y_c \;\; Z_c \;]^T$ and $\overline{X}_p = [\; X_p \;\; Y_p \;\; Z_p \;]^T$ be the 3D position coordinate vectors of P in the camera and the projector reference frame respectively. Let us make a minor change in notation by denoting $\overline{x}_c = [x_c \;\; y_c \;\; 1]^T$ and $\overline{x}_p = [x_p \;\; y_p \;\; 1]^T$ the respective homogeneous coordinates of the projections of P onto the camera and projector image planes. The following expressions relate image coordinates to 3D coordinates:

$$\overline{x}_c = \begin{bmatrix} x_c \\ y_c \\ 1 \end{bmatrix} = \frac{1}{Z_c}\overline{X}_c \tag{5.2}$$

$$\overline{x}_p = \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \frac{1}{Z_p}\overline{X}_p \tag{5.3}$$

From pre-calibration, we also know the relative positions of the camera and the projector. In other words, we know the rigid body transformation equation that leads the 3D coordinates $\overline{X}_c$ of any point P in the camera reference frame to its coordinates in the projector reference frame $\overline{X}_p$:

$$\overline{X}_p = R\,\overline{X}_c + T \tag{5.4}$$

where R and T are the rotation matrix and the translation vector that define that rigid motion between projector and camera. We will assume those two quantities known from pre-calibration:

$$R = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \qquad T = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} \tag{5.5}$$

By substituting equations 5.2 and 5.3 into equation 5.4, we obtain:

$$Z_p \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = Z_c\,R\,\overline{x}_c + T \tag{5.6}$$

Since we do not have direct access to the second projector coordinate y_p, the second equation in (5.6) may be dropped, leading to the following system of two equations and two unknowns:

$$Z_p \begin{bmatrix} x_p \\ 1 \end{bmatrix} = Z_c\,R_{[1,3]}\,\overline{x}_c + T_{[1,3]} \tag{5.7}$$

where $R_{[1,3]}$ is the 2×3 matrix containing the first and third row of the rotation matrix R, and $T_{[1,3]} = [T_1 \;\; T_3]^T$ is a 2-vector containing the first and third coordinates of the translation vector T.

From 5.7, we can solve explicitly for the depths unknowns Z_c and Z_p:

$$\begin{bmatrix} Z_c \\ Z_p \end{bmatrix} = \mathbf{M}^{-1} T_{[1,3]} \tag{5.8}$$

where $\mathbf{M}$ is the following 2×2 matrix:

$$\mathbf{M} = \begin{bmatrix} -R_{[1,3]} \overline{x}_c & \begin{bmatrix} x_p \\ 1 \end{bmatrix} \end{bmatrix} \tag{5.9}$$

A closed form expression for the depth in the camera reference frame Z_c may then be derived:

$$Z_c = \frac{T_1 - x_p T_3}{\langle -R_{[1]} + x_p R_{[3]}, \overline{x}_c \rangle} \tag{5.10}$$

where $R_{[1]} = [R_{11} \quad R_{12} \quad R_{13}]^T$ and $R_{[3]} = [R_{31} \quad R_{32} \quad R_{33}]^T$ are the first and third rows vectors of R. Notice that the denominator of equation 5.10 is a scalar product. Observe that equation 5.10 may also be written in the form of equation 2.33:

$$Z_c = \frac{1}{\langle \overline{\omega}, \overline{x}_c \rangle} \tag{5.11}$$

where $\overline{\omega}$ is the coordinate vector of the plane Π in the camera reference frame (using dual-space formalism):

$$\overline{\omega} = \frac{-R_{[1]} + x_p R_{[3]}}{T_1 - x_p T_3} \tag{5.12}$$

Then, triangulation is possible if and only if the vector $\overline{\omega}$ is well defined, or equivalently if $T_1 - x_p T_3 \neq 0$. This is equivalent to enforcing the optical ray $(O_c, \overline{x}_c)$ and the plane Π to intersect each other (see figure 5.2).

Under that condition, the final expression for $\overline{X}_c$ is:

$$\overline{X}_c = Z_c \overline{x}_c = \frac{T_1 - x_p T_3}{\langle -R_{[1]} + x_p R_{[3]}, \overline{x}_c \rangle} \overline{x}_c = \Delta(\overline{x}_c, x_p) \tag{5.13}$$

which concludes the derivation of the triangulation operator Δ introduced in equation 5.1.

Observe that the plane Π is the plane spanned by the vertical line λ' (in the projector image) of homogeneous coordinate vector $\overline{\lambda}' \simeq [1 \;\; 0 \;\; -x_p]^T$. Therefore, the plane coordinate vector $\overline{\omega}$ could also have been easily computed from equation 2.41:

$$\overline{\omega} = -\frac{R^T \overline{\lambda}'}{\langle T, \overline{\lambda}' \rangle} \tag{5.14}$$

Equation 5.13 is then equivalent to equation 2.43:

$$\overline{X}_c = -\frac{\langle T, \overline{\lambda}' \rangle \, \overline{x_c}}{\langle R^T \overline{\lambda}', \overline{x}_c \rangle} \tag{5.15}$$

Notice that in order to compute the scene depth at a pixel $\overline{x}_c$, it is necessary to know its corresponding horizontal projector coordinate x_p. A technique for computing that correspondence is described in the next section 5.3.

5.3 Temporal processing for correspondence

The main novelty of our technique is in the process of computing the correspondence between image coordinates $\overline{x}_c$ and projector coordinates x_p. For that purpose, we project a succession of horizontally shifted grayscale sinusoidal patterns (see section 5.3.1). Then, from the temporal brightness information collected from the images, we recover, at every pixel $\overline{x}_c$ the corresponding projector coordinate x_p.

5.3.1 Sequence of patterns

The projector projects a succession of N grayscale patterns which are translated from one to another along the horizontal direction (in the x_p direction). Figure 5.3 shows four samples patterns. In this case, a sinusoidal waveform is chosen, and $N = 32$ patterns are projected (fewer patterns might be sufficient).

Figure 5.4 shows the horizontal profile of two patterns, the first one and the 9^{th}

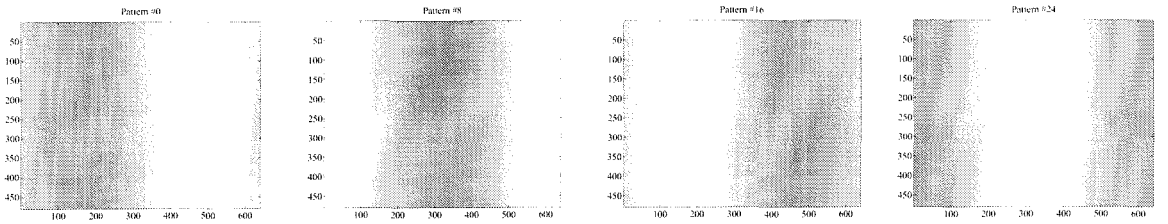


Figure 5.3: The set of projected pattern: We project a succession of $N = 32$ sinusoidal wave patterns of one period over the screen width of 640 pixels. The height of the pattern images is 480 pixels. The brightness extrema of the brightness wave are 255 and 176. Two consecutive patterns are shifted to the right by 20 pixels one with respect to the other (640/32). We show here a sample of 4 patterns among the 32: the first (pattern #0), 9^{th} (pattern #8), 17^{th} (pattern #16) and 25^{th} (pattern #24). Notice how the sinusoidal patterns translates to the right. Notice as well that all patterns are uniform along the vertical direction.

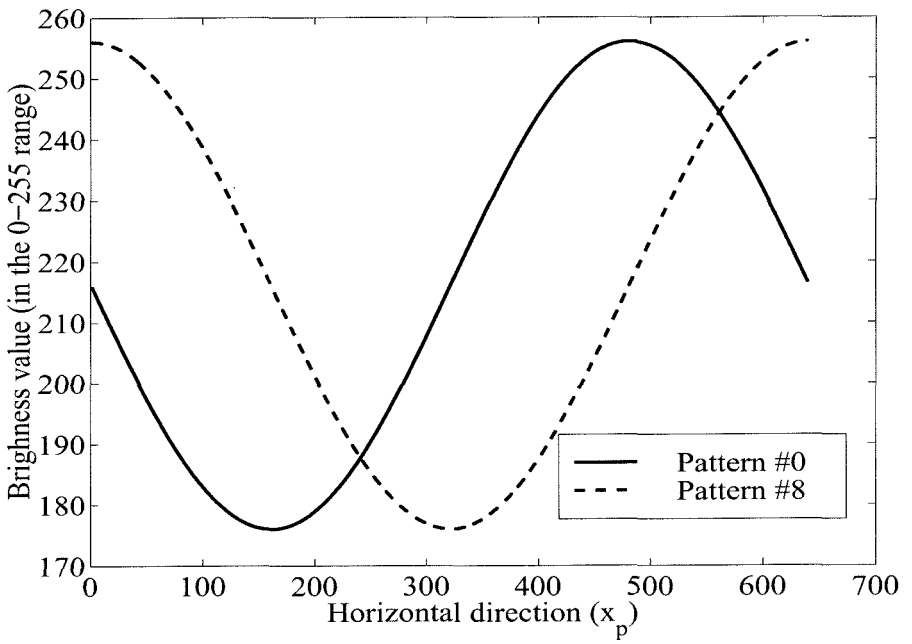


Figure 5.4: Brightness profile of two of the pattern: All the patterns have a similar sinusoidal shape with extrema 255 and 176. This figure shows the horizontal brightness profile of the first (pattern #0) and 9^{th} (pattern #8) projected patterns. All of them are horizontally shifted to the right (by 20 pixels between two consecutive ones). Notice that the waveforms show only one period of the sinusoid.

one (therefore as a function of x_p). Notice that pattern #8 is shifted to the right by a quarter of period ($\pi/4$ phase) with respect to pattern #0.

5.3.2 Temporal processing of the brightness function at every pixel

As each pattern is projected onto the scene, one camera image is acquired (see figure 5.5). Throughout the sequence, one given pixel $\overline{x}_c$ corresponds to a unique point P in the scene. This point P is also illuminated by a unique vertical stripe defined by one horizontal coordinate x_p in the projector image. We know from section 5.2 that the 3D location of P may be computed from the image-projector correspondence $\overline{x}_c \leftrightarrow x_p$. We will show here that given the temporal brightness signal at a pixel $\overline{x}_c$, we can infer its corresponding projector coordinate x_p.

If we observe the temporal patterns of the incident light at two points P_1 and P_2 in the 3D scene, they only differ from the phase. Indeed, they are both sinusoidal signals, but one is shifted with respect to the other by an amount that is directly related to the difference of their projector coordinates x_p. For example, the temporal signal attached to a point illuminated by the medium projector stripe $x_p = 320$ will be π phase shifted with respect to that of a point illuminated by the first stripe $x_p = 1$. There is therefore a linear one-to-one map between the temporal phase shift, and the projector coordinate. Extracting the phase shift of the incident light signal corresponds to estimating the coordinate in the projector image of the vertical stripe that lit the observed point in the scene.

However, we don't have direct access to the incident light going in the scene, but only the reflected light leaving the object and going to the camera sensor. If we assume that the material reflection function in the scene and the imaging sensor (the camera) have significantly linear behaviors, we can still make the same phase statement on the temporal brightness waveform collected in the images for every pixel. Therefore, the problem of extracting projector coordinate at a given pixel in the image directly translates into estimating the phase of the temporal brightness signal at that pixel.

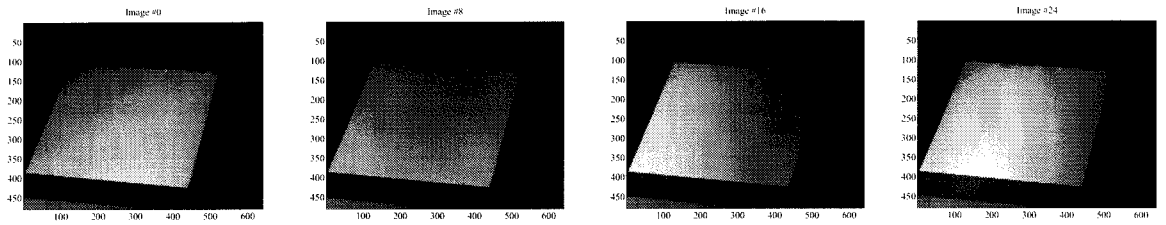


Figure 5.5: The set of acquired images: One image is acquired per pattern. We show here four of them (out of 32): the first one (#0), the 9^{th} (#8), the 17^{th} (#16) and the 25^{th} (#24). Across time, every pixel $\overline{x}_c$ in the image sees a sinusoidal wave. See figure 5.6. From the phase shift of that sinusoidal wave, we can infer the coordinate x_p of the vertical plane in the projector that lit the point P in space. Once x_p is estimated, 3D triangulation (sec. 5.2) can be performed.

Figure 5.5 shows an example set of collected images.

Figure 5.6 shows the temporal brightness at 5 different pixels located on the same row in the image, as a function of time (or patterns index). Notice that the waveforms are all sinusoidal, and differ one from the other in amplitude (A), offset (B) and phase (Φ). They all have the same frequency: $\omega_0 = 2\pi/N$, where $N = 32$ is the number of patterns.

Let us define $I(n)$ to be the observed temporal brightness function at a given pixel $\overline{x}_c = (x_c, y_c)$ in the image, as a function of n, the pattern number (n goes from 0 to $N-1$ and is sometimes associated to time). For clarity reasons, we will not index $I(n)$ with the pixel location $\overline{x}_c$. However the reader needs to keep in mind that this function is different from pixel to pixel. We can model $I(n)$ as follows:

$$I(n) = A\,\sin(\omega_0 n - \Phi) + B \tag{5.16}$$

Given the type of pattern we use in that particular case (a single period sinusoidal waveform), the phase shift Φ can be shown to be linearly related to the projector coordinate x_p through the following one-to-one equation:

$$x_p = \frac{N_p \Phi}{2\pi} \tag{5.17}$$

if Φ is assumed to be expressed in the $0 - 2\pi$ range, and N_p is the width (in pixel) of

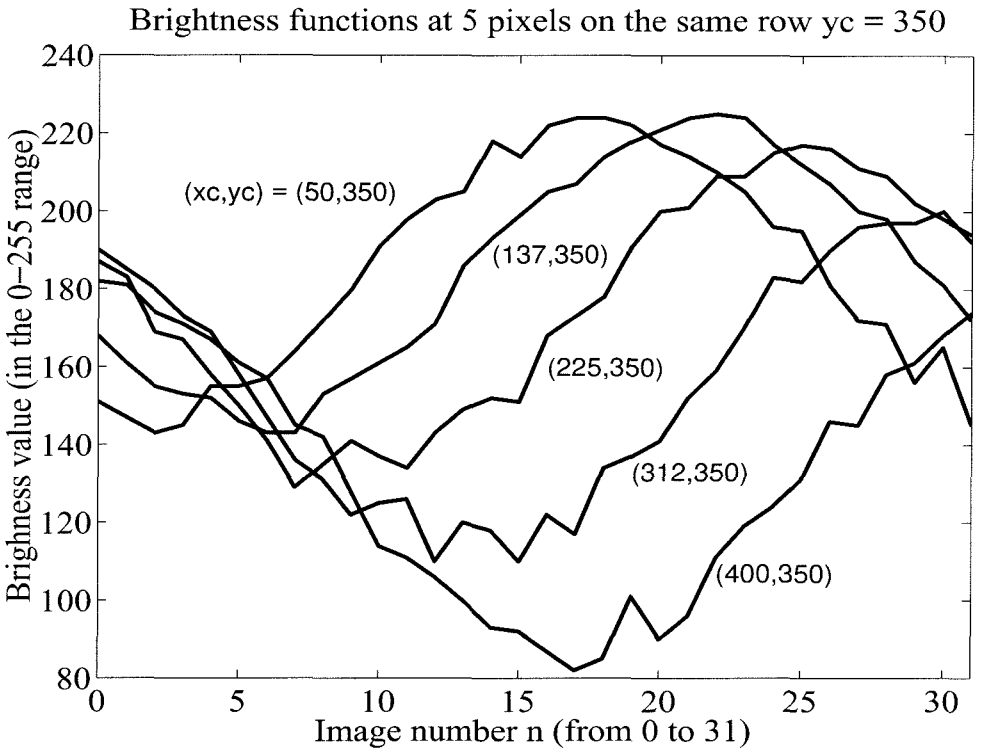


Figure 5.6: Temporal brightness value at every pixel $\overline{x}_c$ in the image: Every pixel $\overline{x}_c$ sees a sinusoidal brightness wave pattern going across it as a function of time. The figure shows that brightness function for 5 different pixels picked on the same row ($y_c = 350$), at positions $x_c = 50$, $x_c = 137$, $x_c = 225$, $x_c = 312$ and $x_c = 400$ (this corresponds to a lower row in the images 5.5). Notice that all the wave forms are sinusoidal with different phases, amplitudes and offsets. The phase information will give us direct estimate of the vertical projector coordinate x_p, This is that quantity that we wish to extract from the acquired waveforms.

the projector image ($N_p = 640$ pixels here). Therefore, estimating x_p at every pixel is equivalent to estimating the phase shift Φ of the associate temporal brightness function.

Define now two quantities a and b as follows:

$$a \doteq \langle I(n), \sin(\omega_0 n) \rangle \doteq \frac{2}{N} \sum_{n=0}^{N-1} I(n) \sin(\omega_0 n) \tag{5.18}$$

$$b \doteq -\langle I(n), \cos(\omega_0 n) \rangle \doteq -\frac{2}{N} \sum_{n=0}^{N-1} I(n) \cos(\omega_0 n) \tag{5.19}$$

One can see that given the model equation 5.16 for the temporal brightness function $I(n)$, we have the following properties for a and b:

$$\begin{cases} a &= A\cos(\Phi) \\ b &= A\sin(\Phi) \end{cases} \tag{5.20}$$

The proof of those relations is relatively straightforward (it only involves simple trigonometry). The most interesting feature of those relations is that neither a nor b contain the offset B. That allows to naturally isolate the amplitude A and the phase Φ. Actually, the quantities Φ and A are respectively the argument and modulus of the complex number $a + ib$ (i is the pure imaginary number such as $i^2 = -1$), and can therefore be easily extracted independently:

$$\begin{cases} \Phi &= \arg(a + ib) = \arctan(b/a) \\ A &= \|a + ib\| = \sqrt{a^2 + b^2} \end{cases} \tag{5.21}$$

Notice that the arctan function here is assumed to return the argument in the $0 - 2\pi$ range without any π ambiguity. In other words, we have access here to both values a and b not only the ratio b/a between them. There is therefore no sign ambiguity in the two terms, which means that the phase is extracted with no π ambiguity.

Finally, from relations 5.17 and 5.21 we obtain expressions for both the projector

coordinate x_p and the sine wave amplitude A:

$$x_p = \frac{N_p \Phi}{2\pi} = \frac{N_p}{2\pi} \arctan(b/a) = \frac{N_p}{2\pi} \arctan\left(-\frac{\langle I(n), \cos(\omega_0 n) \rangle}{\langle I(n), \sin(\omega_0 n) \rangle} \right) \tag{5.22}$$

$$A = \sqrt{a^2 + b^2} = \sqrt{\langle I(n), \sin(\omega_0 n) \rangle^2 + \langle I(n), \cos(\omega_0 n) \rangle^2} \tag{5.23}$$

Experimentally, pixels with large corresponding amplitudes A will be more reliable for phase estimation than pixels with small amplitudes. This is used to help rejecting noisy regions in the image. If a pixel falls in a shadow region of the scene (outside of the field of view of the projector), it will not be lit by the projected sine wave pattern. And therefore, its associate temporal brightness signal will almost not change across time (except within the pixel noise). A similar situation occurs for dark regions of the scene (with small surface albedo). In those cases, any phase extraction is hopeless from the start. Fortunately, those situations nicely translate into significantly small amplitude estimates A. Therefore, one can simply reject regions of the image that have a corresponding amplitude A less than the pre-chosen threshold A_T. It turns out that this thresholding method for "good" area selection is very robust: from "good" to "bad" regions, A typically abruptly drops by one order of magnitude, going from 50 to 5 gray levels. We consistently picked in our experiments a threshold of $A_T = 20$ gray levels.

From that stage, we obtain for every pixel $\overline{x}_c$ whose amplitude estimate A (given by equation 5.23) is larger than a threshold A_T ($A_T = 20$ here), a corresponding projector coordinate x_p (given by equation 5.22). The final 3D shape estimation stage can then be performed for all of those points (see section 5.2) resulting in the three-dimensional shape of the scenery. The next section 5.4 presents some reconstruction results.

5.4 Experimental results

Figure 5.7 shows the recovered projector coordinate map (x_p-map) over the entire image. The way it is shown is by gray-encoded the values of x_p in the 0-255 range.

Figure 5.8 shows a cross section of the x_p-map over all the pixels on the row $y_c = 350$ in the image. We can notice on that figure how linearly the projector coordinate increases while going from the left to the right side of the scenery. This is naturally expected since the observed object is planar.

After triangulation, we obtain the 3D coordinates of the observed points in the scenery in the camera reference frame: $\overline{X}_c = [X_c \quad Y_c \quad Z_c]^T$. Figure 5.9 shows the depth coordinates Z_c values for every pixel $\overline{x}_c$ in the image, in a gray-encoded fashion. Darker pixels correspond to closer points to the camera, except for the completely black region which is the rejected area after thresholding of A.

Figure 5.10 shows two synthetic views of the final set 3D reconstructed points from the left and the from the right of the initial camera location. Figure 5.11 shows two views of the 3D surface mesh generated from the cloud of points. Since reconstruction is achieved densely in the image plane, connectivity is directly established in pixel coordinates.

5.5 Error analysis

The method we propose here allows to compute the scene depth Z_c at every pixel $\overline{x}_c$ in the image. As we noticed in the experimental section, the final 3D reconstruction results are not perfect. Depth estimates are corrupted with errors that are significantly noticeable on the final 3D structure (see figure 5.10).

Those errors are due to brightness noise in the input image, as well as errors in the temporal model described by equation 5.16. Since it is quite difficult to characterize the errors introduced by the simplified sinusoidal model, let us suppose that the only source of noise is the one attached to the input images. Let us model that brightness noise by an additive Gaussian random variable with zero-mean and variance σ_I^2 (uni-

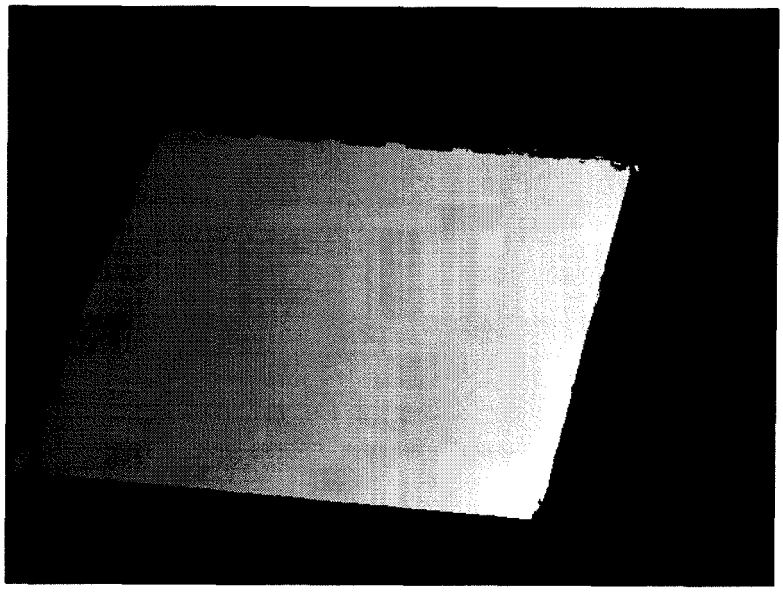


Figure 5.7: Projector coordinate x_p map: The projector coordinates x_p are computed for the pixels $\overline{x}_c$ whose amplitude A is larger than $A_T = 20$. This image shows x_p in a gray value encoded fashion. Notice that, as expected, the pixel brightness gently increases while going from the left the the right portion of the plane. The completely black regions of the image corresponds to rejected points after thresholding of the amplitude A.

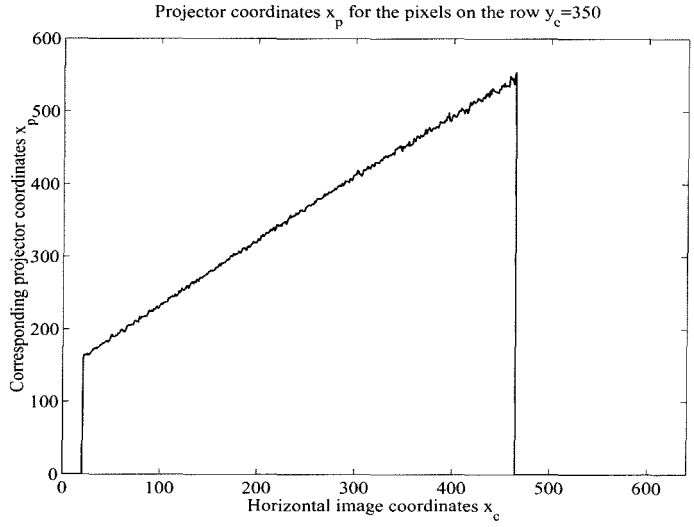


Figure 5.8: Projector coordinates x_p of the pixels on row 350: This figure shows a section of the x_p-map of figure 5.7 at row $y_c = 350$. Notice that the projector coordinate varies linearly with the horizontal pixel coordinate x_c. This is expected since the observed object is a plane. The $x_p = 0$ pixels simply correspond to rejected area in the image after thresholding of A.

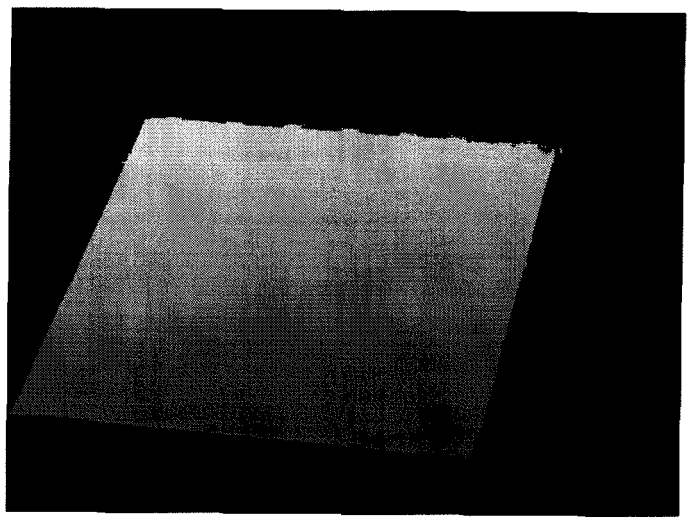
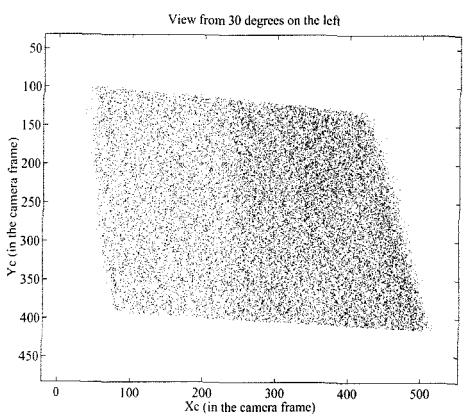


Figure 5.9: The recovered depth map: After triangulation, every pixel in the image has an associated point in 3D whose coordinate in the camera reference frame is $\overline{X}_c = [X_c \ Y_c \ Z_c]^T$. This figure shows for every pixel the recovered depths Z_c, which is the quantity estimated by triangulation. The values are linearly gray-encoded in the 0-255 range, and points further away from the camera have a larger depth, and therefore have a brighter associated gray value. The completely black region still corresponds to rejected points in the image.

(a)



(b)

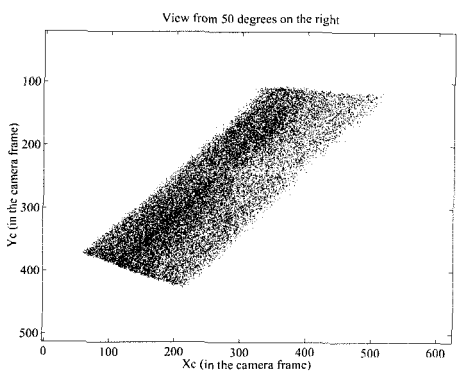


Figure 5.10: The final 3D reconstructed shape: Figures (a) and (b) are synthetic views of the 3D structure after rotation of the camera to the left and to the right respectively. There are 124106 points covering the surface. One can appreciate on that figure the type of uncertainties we are achieving on the final shape estimate. We fit a plane across the points in space and then looked at the residual algebraic distance of the points to the plane. The standard deviation of those distances is approximately 6mm (the overall size of the scene is approximately 30×30 cm^2).

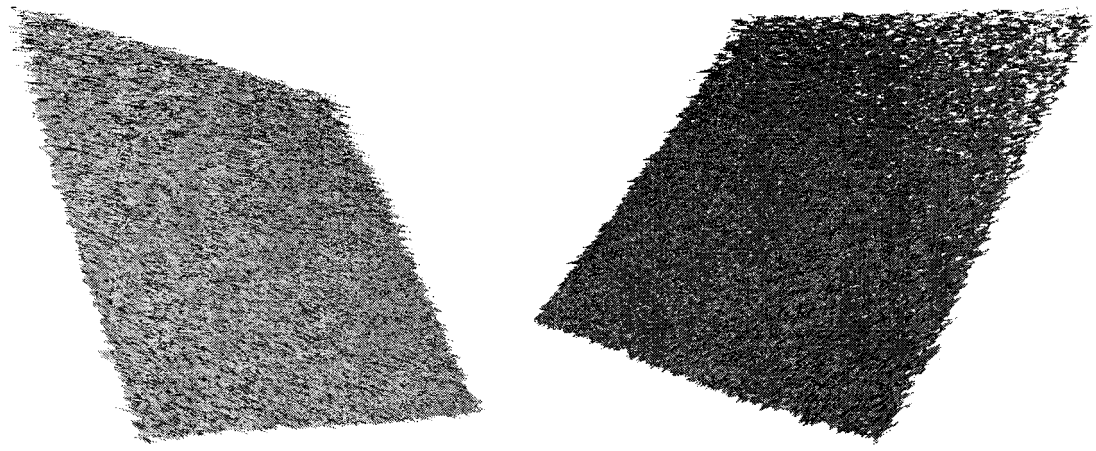


Figure 5.11: The final 3D reconstructed shape: Two views of the 3D mesh generated by the cloud of points shown on figure 5.10.

form across the whole image). The goal of a noise sensitivity analysis is to compute the effect of this input brightness noise onto the final depth measurement Z_c.

Let $\sigma_{x_p}^2$ be the variance of the error on the projector coordinate x_p. An intermediate step into the analysis will be computing this variance $\sigma_{x_p}^2$ as a function of σ_I^2. Equation 5.22 provides an expression for the projector coordinate x_p (in pixel units) as a function of the N brightness measurements at pixel $\overline{x}_c$ across time $I(0), I(1), \ldots, I(N)$:

$$x_p = \frac{N_p}{2\pi} \arctan\left(-\frac{\langle I(n), \cos(\omega_0 n)\rangle}{\langle I(n), \sin(\omega_0 n)\rangle}\right) \doteq \mathcal{F}\left(I(0), I(1), \ldots, I(N)\right) \tag{5.24}$$

Since all the brightness measurements $I(0), I(1), \ldots, I(N)$ are supposed to carry the same noise term of variance σ_I^2, the final projector coordinate variance $\sigma_{x_p}^2$ may be approximated by the following expression:

$$\sigma_{x_p}^2 = \sum_{n=0}^{N} \left(\frac{\partial \mathcal{F}}{\partial I(n)}\right)^2 \sigma_I^2 \tag{5.25}$$

After straightforward derivation, we may show that:

$$\frac{\partial \mathcal{F}}{\partial I(n)} = \frac{N_p}{\pi\, A\, N} \cos(\omega_0 n + \Phi) \tag{5.26}$$

where Φ and A are defines in equation 5.16. Therefore, the variance of the error on projector coordinate x_p may be written:

$$\sigma_{x_p}^2 = \left(\frac{N_p\,\sigma_I}{\pi\,A\,N}\right)^2 \sum_{n=0}^{N} \cos^2(\omega_0 n + \Phi) \tag{5.27}$$

Recalling that $\omega_0 = 2\pi/N$, one may show that the following relation holds:

$$\forall \Phi \in \mathbb{R}, \qquad \sum_{n=0}^{N} \cos^2(\omega_0 n + \Phi) = \frac{N}{2} \tag{5.28}$$

Therefore, the variance of the error attached to the projector coordinate x_p takes the following compact form (in pixel units):

$$\sigma_{x_p}^2 = \frac{N_p^2}{2\pi^2\,A^2\,N}\,\sigma_I^2 \tag{5.29}$$

A set of three fundamental observations may be drawn from that relation. First, notice that $\sigma_{x_p}^2$ is inversely proportional to the number of projected patterns N. This is quite intuitive: as the number of patterns increases, the accuracy of phase estimate Φ increases, which in turns makes the projector coordinate x_p more accurate. Second, notice that $\sigma_{x_p}^2$ is inversely proportional to the square of the brightness amplitude A. This supports the fact that pixels with larger temporal brightness variations (larger contrasts) are more reliable for phase estimation than ones with smaller brightness variations. This also adds a supportive argument in favor of the thresholding technique for rejecting too unreliable pixels. Finally, observe that as the projecting image width N_p increases, the accuracy in estimating x_p decreases. That is also quite intuitive: for a given error in phase estimate Φ (in radians), the corresponding error in estimating x_p (in pixels) will be larger on wider projecting images. In consequence, for a given pattern width, it would be beneficial to project patterns with more than one sinusoidal period. Doing so, the effective width of a period is smaller ($N_p \rightarrow N_p/k$ where k is the number of periods) decreasing the transferred error onto the projector pixel coordinate x_p (the proportionality factor between Φ and x_p is smaller - see

equation 5.17). However if the projecting patterns contain more than a single period, an ambiguity in corresponding phase information to pixel coordinates is introduced (if $k = 2$, there are two valid pixel coordinates associated to any given phase value Φ). In order to solve for that ambiguity, one could use a combination of high and low frequency patterns. The low frequency patterns (that could be either grayscale, or strict black and white stripes) would help disambiguate the high frequency ones providing the high local resolution. That is subject for future work.

Observe that before computing depth Z_c by triangulation, the projector coordinate x_p needs to be normalized (from pixel coordinates to homogeneous coordinates). It is therefore necessary to divide the pixel coordinate by the projector focal length in pixels. Denote that focal length f_p. Then, after normalization, the variance of the error on the normalized projector coordinate becomes:

$$\sigma_{x_p}^2 = \frac{N_p^2}{2\pi^2 A^2 N f_p^2} \sigma_I^2 \tag{5.30}$$

Observe that the units are preserved: both N_p and f_p are in pixels, and A and σ_I are in units of brightness values. Notice that the ratio N_p/f_p is directly related to the horizontal field of view angle of the projector. Then, the variance $\sigma_{Z_c}^2$ of the error on the depth estimate Z_c may be written:

$$\sigma_{Z_c}^2 = \left(\frac{\partial Z_c}{\partial x_p}\right)^2 \sigma_{x_p}^2 \tag{5.31}$$

The Jacobian matrix $(\partial Z_c/\partial x_p)$ may be decomposed as follows:

$$\frac{\partial Z_c}{\partial x_p} = \left(\frac{\partial Z_c}{\partial \overline{\omega}}\right)\left(\frac{\partial \overline{\omega}}{\partial x_p}\right) \tag{5.32}$$

where $\overline{\omega}$ is the coordinate vector of the plane Π spanned by the projector in dual-space (see figure 5.2). The expression for this vector is given by equation 5.12. From that equation, one may derive an expression for the second Jacobian matrix appearing in

equation 5.32:

$$\frac{\partial \overline{\omega}}{\partial x_p} = \frac{T_1 R_{[3]} - T_3 R_{[1]}}{(T_1 - x_p T_3)^2} \tag{5.33}$$

From the triangulation equation 5.11, an expression for the first Jacobian matrix may also be derived:

$$\frac{\partial Z_c}{\partial \overline{\omega}} = Z_c^2 \overline{x}_c^T \tag{5.34}$$

Equations 5.33 and 5.34 yield:

$$\frac{\partial Z_c}{\partial x_p} = Z_c^2 \frac{\langle T_1 R_{[3]} - T_3 R_{[1]}, \overline{x}_c \rangle}{(T_1 - x_p T_3)^2} = \frac{\langle T_1 R_{[3]} - T_3 R_{[1]}, \overline{x}_c \rangle}{\langle -R_{[1]} + x_p R_{[3]}, \overline{x}_c \rangle^2} \tag{5.35}$$

Notice that as the quantity $(T_1 - x_p T_3)^2$ decreases (towards zero), the sensitivity to noise increases. This is quite an intuitive trend, since in the limit $(T_1 - x_p T_3)^2 = 0$ corresponds to a degenerate case where triangulation is impossible (the projector plane Π contains the center projection O_c). In addition, observe that $(\partial Z_c / \partial x_p)$ is proportional to Z_c^2. This intuitively means that points further away from the camera are more sensitive to noise that ones closer to the camera.

After merging equations (5.35), (5.31) and (5.30), we reach to a final expression for the variance $\sigma_{Z_c}^2$ of the error on the depth estimate Z_c:

$$\sigma_{Z_c}^2 = \frac{N_p^2}{2\pi^2 A^2 N f_p^2} \frac{\langle T_1 R_{[3]} - T_3 R_{[1]}, \overline{x}_c \rangle^2}{\langle -R_{[1]} + x_p R_{[3]}, \overline{x}_c \rangle^4} \sigma_I^2 \tag{5.36}$$

Observe that in that equation, both vectors $\overline{x}_c$ and x_p are assumed to be normalized.

We applied the projector coordinate variance expression 5.30 on our data set. We first acquired a set of 10 images to compute experimentally the image brightness noise: $\sigma_I \approx 2$ brightness units. Then, equation 5.30 returned an estimate for σ_{x_p} between 1.1 and 1.4 pixels. That error estimate is very similar to that computed experimentally: between 1 and 1.6 pixels. Notice that we experimentally computed σ_{x_p} by first fitting planes to local 20×20 neighborhood of the x_p-map (see figure 5.7) and then computing

the residual deviations of all the points of the neighborhood to the plane. From that result, we conclude that our error model is sufficiently accurate to capture reality. Finally, according to the noise model of equation 5.36, the standard deviation σ_{Z_c} of the error on the depth reconstruction varies from 3 and 5mm. This estimate appears to be slightly below the actual measured errors on the final reconstruction: 6mm (see figure 5.10).

5.6 Conclusions

In this chapter, we have presented a new method for acquiring three-dimensional shape based on structured lighting technology. This technique consists of projecting a sequence of grayscale patterns using a conventional LCD projector, and computing the depth of every pixel in the image by temporal brightness processing. The main advantage of that technique compared to standard structured lighting techniques is that the outcome is a dense reconstruction of the scene in the pixel image. Consequently, meshing and texture mapping are straightforward tasks. A complete error analysis of the reconstruction method was also presented.

In that present description, we used a sequence of patterns with sinusoidal profiles for establishing dense correspondence between camera image and projector image. It is worth noticing that an identical image processing technique could be applied when using other kind of periodic profiles (such as triangular profile). We intend to test the method with several other grayscale pattern profiles, at potentially different frequencies (in order to achieve best reconstruction accuracies).

The main practical drawback of this structured lighting technique is that an external active device is necessary (the LCD projector). In addition, that device is required to be calibrated (intrinsically and extrinsically) as well as controlled. In the next chapter, we propose another technique for 3D scanning - also based on the philosophy of active lighting - that does not require any active device. We call this new technology 'weak structured lighting.'

Chapter 6 Weakly structured lighting - Scanning using shadows

6.1 Introduction and motivation

In designing a system for recovering three-dimensional shape, different engineering tradeoffs are proposed by each application. The main parameters to be considered are cost, accuracy, ease of use and speed of acquisition. So far the commercial 3D scanners (e.g., the Cyberware scanner) have emphasized accuracy over the other parameters. Active illumination systems are popular in industrial applications where a fixed installation with controlled lighting is possible. These systems use motorized transport of the object and active (laser, LCD projector) lighting of the scene which makes them very accurate, but unfortunately expensive [4, 5, 6, 7, 8]. Furthermore these systems cannot be used in outdoors where lighting is difficult to control and high power would be needed for large objects.

An interesting challenge for vision scientists is to take the opposite point of view: emphasize low cost and simplicity and design 3D scanners that demand little more hardware than a PC and a video camera by making better use of the data that is available in the images.

A number of passive cues have long been known to contain information on 3D shape: stereoscopic disparity, texture, motion parallax, (de)focus, shadows, shading and specularities, occluding contours and other surface discontinuities. At the current state of vision research stereoscopic disparity is the single passive cue that reliably gives reasonable accuracy. Unfortunately it has two major drawbacks: it requires two cameras thus increasing complexity and cost, and it cannot be used on untextured surfaces, which are common for industrially manufactured objects.

We propose a method for capturing 3D surfaces that is based on what we call

'weakly structured lighting.' It yields good accuracy and requires minimal equipment besides a computer and a camera: a stick, a checkerboard, and a point light source. The light source may be a desk lamp for indoor scenes, and the sun for outdoor scenes. A human operator, acting as a low precision motor, is also required. See [73, 74, 75, 76, 77].

We start with the description of the scanning method in Sec. 6.2, followed in Sec. 6.3 by an complete error analysis of our reconstruction technique. In Sec. 6.4, a simple algorithm for optimally merging multiple 3D scans is presented, followed in Sec. 6.5 by a short description of a real-time implementation of the 3D scanner. The following Sec. 6.6 reports a number of experiments that assess the convenience and accuracy of the system in indoor as well as outdoor scenarios. We close this chapter with discussions and conclusions in Sec. 6.7.

6.2 Description of the method

The general principle consists of casting a moving shadow with a stick onto the scene, and estimating the three-dimensional shape of the scene from the sequence of images of the deformed shadow. Figures 6.1 and 6.2 show two incarnations of the scanning setup. The objective is to extract scene depth at every pixel in the image. The point light source and the leading edge of the stick define, at every time instant, a plane; therefore, the boundary of the shadow that is cast by the stick on the scene is the intersection of this plane with the surface of the object. We exploit this geometrical insight for reconstructing the 3D shape of the object. Figure 6.3 illustrates the geometrical principle of the method. Approximate the light source with a point S, and denote by Π_h the horizontal plane (ground, or desk plane) and Π_v a vertical plane orthogonal to Π_h (this plane is not present in the scanning scenario shown on figure 6.1). Assume that the position of the plane Π_h in the camera reference frame is known from calibration (sec. 6.2.1). We infer the location of Π_v from the projection λ_i (visible in the image) of the intersection line Λ_i between Π_h and Π_v (sec. 6.2.2). The goal is to estimate the 3D location of the point P in space corresponding to every

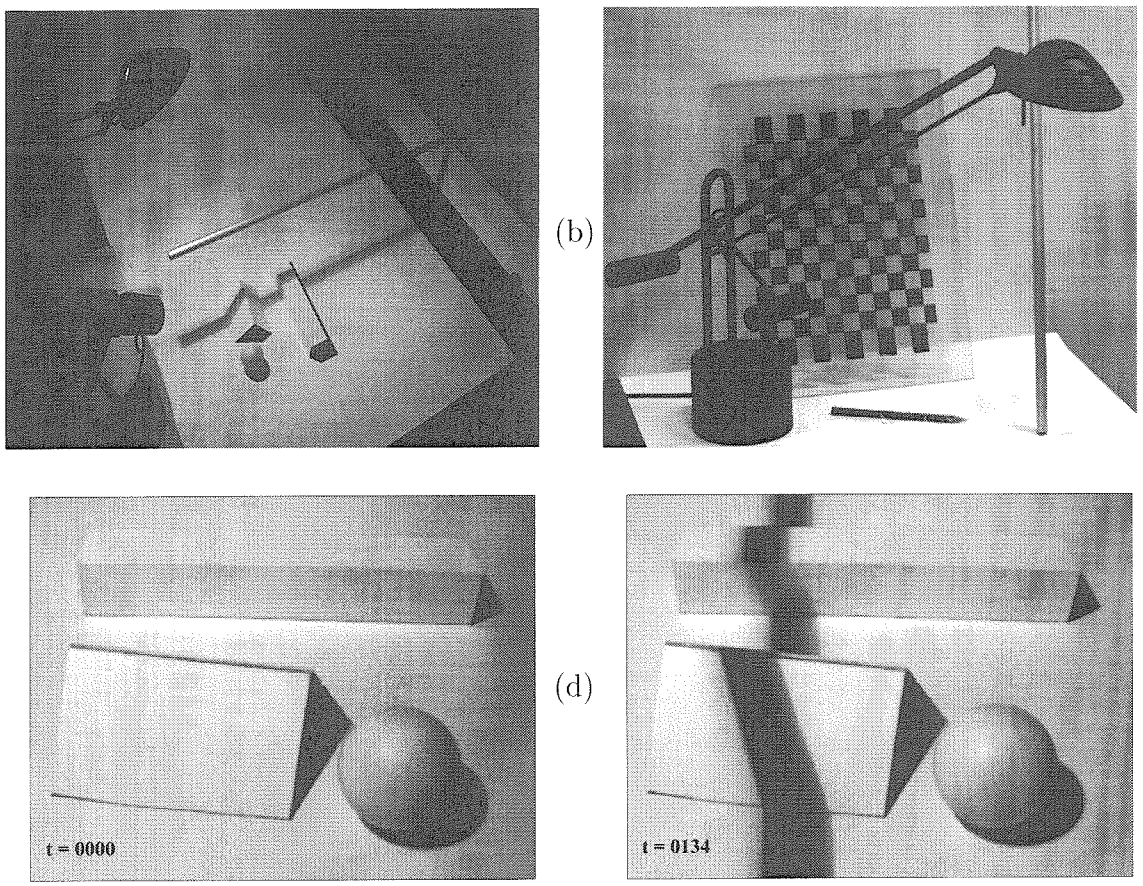


Figure 6.1: The general setup of the method: (a) The camera is facing the scene illuminated by the light source. The objects to scan are positioned on the desk (horizontal plane). Figure (c) is an initial camera view of the scene. When an operator freely moves a stick in front of the light, a shadow is cast on the scene. The camera acquires a sequence of images $I(x, y, t)$ as the operator moves the stick so that the shadow scans the entire scene. A sample image is shown on figure (d). This constitutes the input data to the 3D reconstruction system. The three-dimensional shape of the scene is reconstructed using the spatial and temporal properties of the shadow boundary throughout the input sequence. Figure (b) shows the necessary equipment besides the camera: a desk lamp, a calibration grid and a pencil for calibration, and a stick casting the shadow. One could use the pencil instead of the stick.

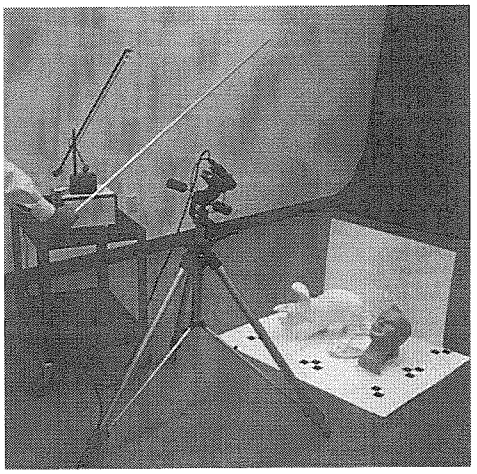
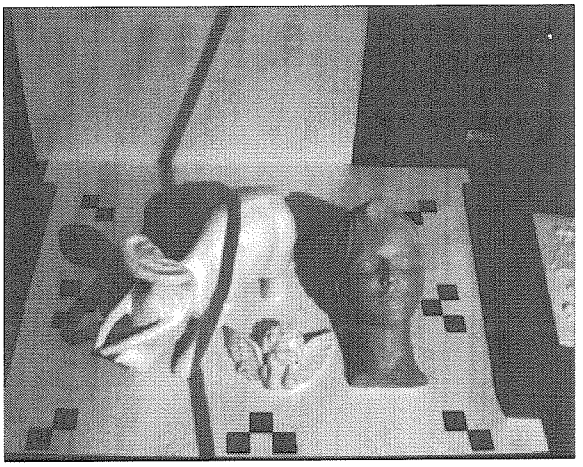
(a)  (b) 

Figure 6.2: Alternative geometrical setup: In this other configuration, the background consists of two orthogonal planes (horizontal and vertical planes). Figure (a) shows the general system setup where the light source is the same as in figure 6.1 without its reflector (this is why its look is different). A sample image acquired during scanning is shown on figure (b). Notice that the shadow is seen on both background planes. In that other incarnation, it will be shown that the light source position does not need to be known for scanning. This will be useful when using the sun as light source for outdoor scanning.

pixel p (of coordinates $\overline{x}_c$) in the image. Call t the time when the shadow boundary passes by a given pixel $\overline{x}_c$ (later referred to as the *shadow time*). Denote by $\Pi(t)$ the corresponding shadow plane at that time t. Assume that two portions of the shadow projected on the two planes Π_h and Π_v are visible on the image: $\lambda_h(t)$ and $\lambda_v(t)$. After extracting these two lines, we deduce the location in space of the two corresponding lines $\Lambda_h(t)$ and $\Lambda_v(t)$ by intersecting the planes $(O_c, \lambda_h(t))$ and $(O_c, \lambda_v(t))$ with Π_h and Π_v respectively. The shadow plane $\Pi(t)$ is then the plane defined by the two non-collinear lines $\Lambda_h(t)$ and $\Lambda_v(t)$ (sec. 6.2.5). In the case where the vertical plane is not used for scanning (as in figure 6.1), the line $\lambda_v(t)$ is not available. In that case, the plane $\Pi(t)$ may be inferred by the only available line $\lambda_h(t)$ and the point light source S (which then needs to be at a fixed and known location in space - see sec. 6.2.3). Finally, the point P corresponding to $\overline{x}_c$ is retrieved by intersecting $\Pi(t)$ with the optical ray (O_c, p). This final stage is called triangulation (sec. 6.2.6). Notice that the key steps are: (a) estimate the shadow time $t_s(\overline{x}_c)$ at every pixel $\overline{x}_c$ (*temporal processing*), (b) locate the two reference lines $\lambda_h(t)$ and $\lambda_v(t)$ at every time instant t

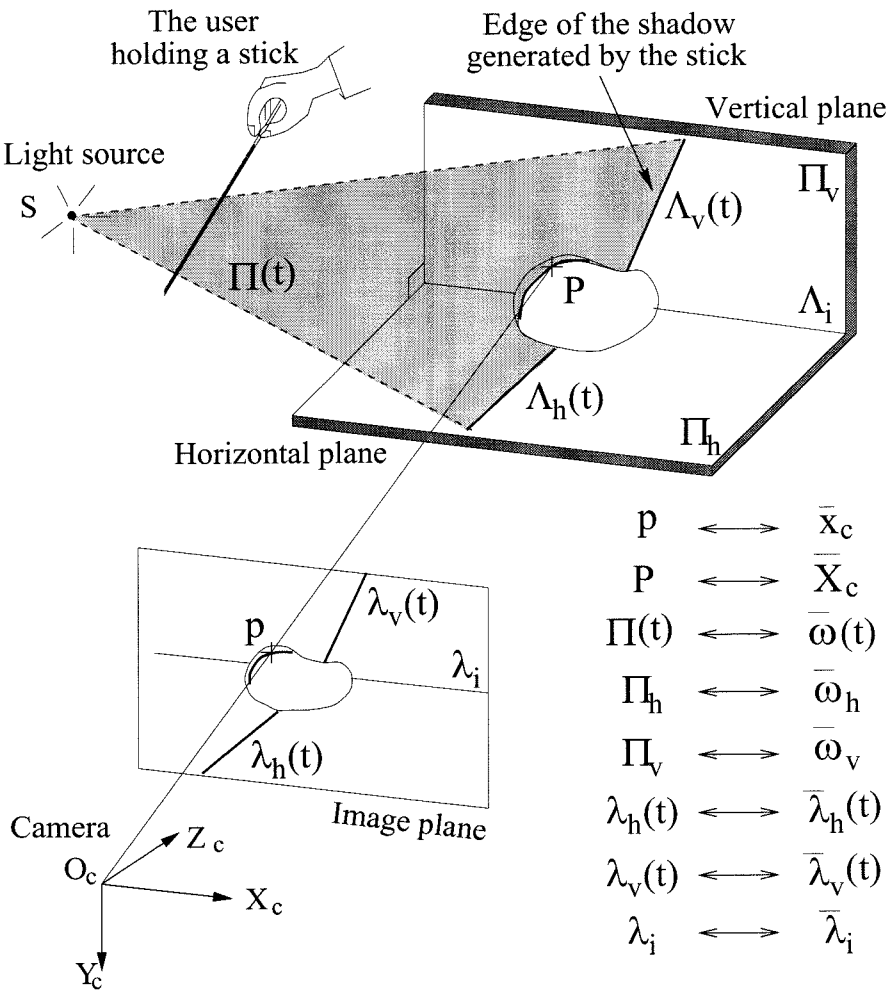


Figure 6.3: Geometrical principle of the method

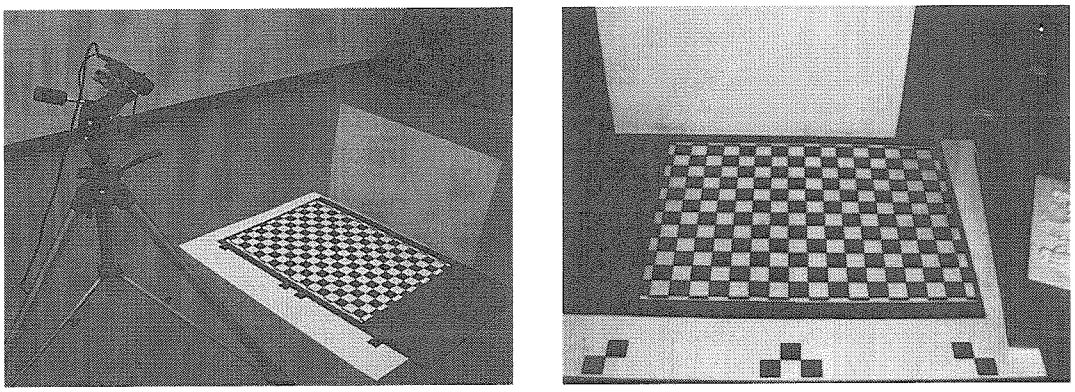


Figure 6.4: Camera calibration

(*spatial processing*), (c) calculate the shadow plane, and (d) triangulate and calculate depth. These tasks are described in sections 6.2.4, 6.2.5 and 6.2.6.

Goshtasby *et al.* [78] also designed a range scanner using a shadow generated by a fine wire in order to reconstruct the shape of dental casts. In their system, the wire was motorized and its position calibrated.

Observe that if the light source is at a known location in space, then the shadow plane $\Pi(t)$ may be directly inferred from the point S and the line $\Lambda_h(t)$. Consequently, in such cases, the additional plane $\Pi_v(t)$ is not required. We describe here two versions of the setup: one containing two calibrated planes and an uncalibrated (possibly moving) light source; the second containing one calibrated plane and a calibrated light source.

6.2.1 Camera calibration

The goal of calibration is to recover the location of the ground plane Π_h and the *intrinsic* camera parameters (focal length, principal point and radial distortion factor). The procedure consists of first placing a planar checkerboard pattern on the ground in the location of the objects to scan (see figure 6.4-left). From the image captured by the camera (figure 6.4-right), we infer the intrinsic and extrinsic parameters of the camera, by matching the projections onto the image plane of the known grid corners with the expected projection directly measured on the image (extracted corners of the grid); the method is proposed by Tsai in [18]. We use a first-order symmetric radial

distortion model for the lens, as proposed in [17, 18] (see equation 3.4). When using a single image of a planar calibration rig, the principal point (i.e., the intersection of the optical axis with the image plane) cannot be recovered. In that case it is assumed to be identical to the image center. In order to fit a full camera model (principal point included), we propose to extend that approach by integrating multiple images of the planar grid positioned at different locations in space (with different orientations). Theoretically, a minimum of two images is required to recover two focals (along x and y), the principal point coordinates, and the lens distortion factor. We recommend to use that method with three or four images for best accuracies on the intrinsic parameters. In our experience, in order to achieve good 3D reconstruction accuracies, it is sufficient to use the simple approach with a single calibration image without estimating the camera principal point. In other words, the quality of reconstruction is quite insensitive to errors on the principal point position. A whole body of work supporting that observation may be found in the literature. We especially advise the reader most interested in issues on sensitivity of 3D Euclidean reconstruction results with respect to intrinsic calibration errors, to refer to recent publications on self-calibration, such as Bougnoux [79] or Pollefeys *et al.* [80, 81, 31].

A detailed description of the fundamental calibration algorithm may be found in chapter 3.

For more general insights on calibration techniques, we refer the reader to the work of Faugeras [10] and others [82, 17, 22, 24, 83, 28]. A 3D rig should be used for achieving maximum accuracy.

6.2.2 Vertical plane localization Π_v

Call $\overline{\omega}_h$ and $\overline{\omega}_v$ respectively the coordinate vectors of Π_h and Π_v (see figure 6.3 and section 2.2.1 for notation). After calibration, $\overline{\omega}_h$ is known. The two planes Π_h and Π_v intersect along the line Λ_i observed on the image plane at λ_i. Therefore, according to proposition 1 in section 2.2.2, $\overline{\omega}_h - \overline{\omega}_v$ is parallel to $\overline{\lambda}_i$, coordinate vector of λ_i, or equivalently, there exists a scalar α such that $\overline{\omega}_v = \overline{\omega}_h + \alpha\overline{\lambda}_i$. Since the two planes

Π_h and Π_v are by construction orthogonal, we have $\langle \overline{\omega}_h, \overline{\omega}_v \rangle = 0$. That leads to the closed-form expression for calculating $\overline{\omega}_v$:

$$\overline{\omega}_v = \overline{\omega}_h - \frac{\langle \overline{\omega}_h, \overline{\omega}_h \rangle}{\langle \overline{\lambda}_i, \overline{\omega}_h \rangle} \overline{\lambda}_i. \tag{6.1}$$

Notice that in every realistic scenario, the two planes Π_h and Π_v do not contain the camera center O_c. Their coordinate vectors $\overline{\omega}_h$ and $\overline{\omega}_v$ in dual-space are therefore always well defined (see sections 2, 6.2.6 and 6.3 for further discussions).

6.2.3 Light source calibration

When using a single reference plane for scanning (for example Π_h without Π_v), it is required to know the location of the light source in order to infer the shadow plane location $\Pi(t)$ (see section 6.2.5 for details). Figure 6.5 illustrates a simple technique for calibrating the light source that requires minimal extra equipment: a pencil of known length. The operator stands a pencil on the reference plane Π_h (see fig. 6.5-top-left). The camera observes the shadow of the pencil projected on the ground plane. The acquired image is shown on figure 6.5-top-right. From the two points $\overline{b}$ and $\overline{t}_s$ on this image, one can infer the positions in space of B and T_s, respectively the base of the pencil, and the tip of the pencil shadow (see bottom figure). This is done by intersecting the optical rays $(O_c, \overline{b})$ and $(O_c, \overline{t}_s)$ with Π_h (known from camera calibration). In addition, given that the height of the pencil h is known, the coordinates of its tip T can be directly inferred from B. The point light source S has to lie on the line $\Delta = (T, T_s)$ in space. This yields one linear constraint on the light source position. By taking a second view, with the pencil at a different location on the plane, one retrieves a second independent constraint with another line Δ'. A closed form solution for the 3D coordinate of S is then derived by intersecting the two lines Δ and Δ' (in the least squares sense). Notice that since the problem is linear, one can integrate the information from more than 2 views and make the estimation more accurate. If $N > 2$ images are used, one can obtain a closed form solution for the closest point $\tilde{S}$ to the N inferred lines (in the least squares sense). We also

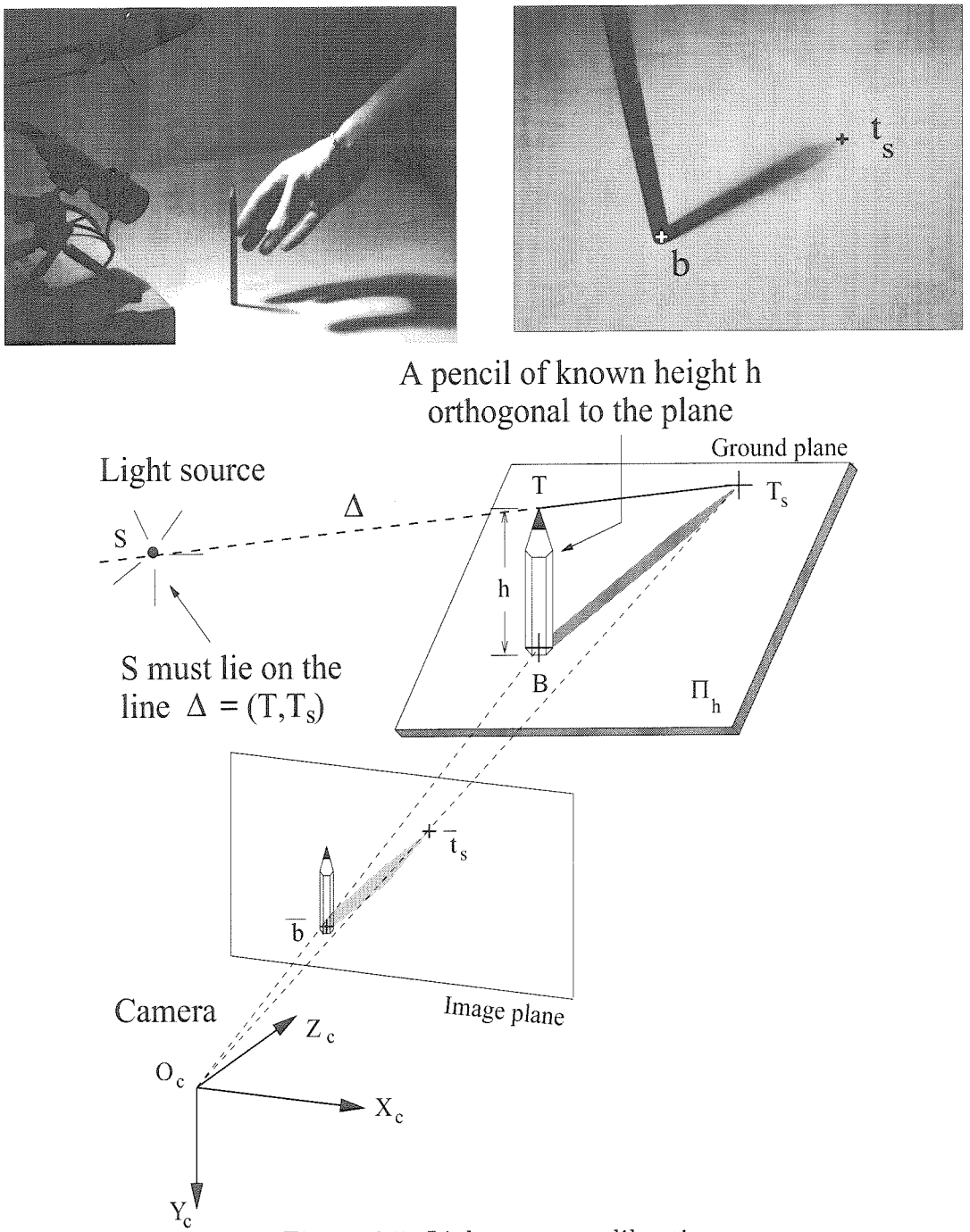


Figure 6.5: Light source calibration

estimate the uncertainty on that estimate from the distance of $\tilde{S}$ to each one of the Δ lines. That indicates how consistently the lines intersect a single point in space. The complete algorithm for intersecting a set of N lines in space may be found in section 4.1.2. In this section, the derivations were made in the context of another application (multi-view stereo triangulation). The same derivations may be found in [74].

6.2.4 Spatial and temporal shadow edge localization

A fundamental stage of the method is the detection of the lines of intersection of the shadow plane $\Pi(t)$ with the two planes Π_h and Π_v; a simple approach to extract $\overline{\lambda}_h(t)$ and $\overline{\lambda}_v(t)$ may be used if we make sure that a number of pixel rows at the top and bottom of the image are free from objects. Then the two tasks to accomplish are: **(a)** Localize the edges of the shadow that are directly projected on the two orthogonal planes $\lambda_h(t)$ and $\lambda_v(t)$ at every discrete time t (every frame), leading to the set of all shadow planes $\Pi(t)$ (see sec. 6.2.5), **(b)** Estimate the time $t_s(\overline{x}_c)$ (*shadow time*) where the edge of the shadow passes through any given pixel $\overline{x}_c = (x_c, y_c)$ in the image (see figure 6.6). Curless and Levoy [84] demonstrated that such a spatio-temporal approach is appropriate for preserving sharp discontinuities in the scene as well as reducing range distortions. A similar temporal processing for range sensing was used by Gruss, Tada and Kanade in [5, 85].

Both processing tasks correspond to finding the edge of the shadow, but the search domains are different: one operates on the spatial coordinates (image coordinates) and the other one on the temporal coordinate. Although independent in appearance, the two search procedures need to be compatible: if at time t_0 the shadow edge passes through pixel $\overline{x}_c = (x_c, y_c)$, the two searches should find the exact same point (x_c, y_c, t_0) (in space/time). One could observe that this property does not hold for all techniques. One example is the image gradient approach for edge detection (e.g., Canny edge detector [86]). Indeed, the maximum spatial gradient point does not necessarily match with the maximum temporal gradient point (which is a function of the scanning speed). In addition, the spatial gradient is a function both of

changes in illumination due to the shadow and changes in albedo and changes in surface orientation. Furthermore, it is preferable to avoid any spatial filtering on the images (e.g., smoothing) which would produce blending in the final depth estimates, especially noticeable at occlusions and surface discontinuities (corners for example). These observations were also addressed by Curless and Levoy in [84].

It is therefore necessary to use a unique criterion that would equally describe shadow edges in space (image coordinates) and time and is insensitive to changes in surface albedo and surface orientation. The simple technique we propose here that satisfies that property is spatio-temporal thresholding. This is based on the following observation: as the shadow is scanned across the scene, each pixel (x, y) sees its brightness intensity going from an initial maximum value $I_{\max}(x, y)$ (when there is no shadow yet) down to a minimum value $I_{\min}(x, y)$ (when the pixel is within the shadow) and then back up to its initial value as the shadow goes away. This profile is characteristic even when there is a fair amount of internal reflections in the scene [87, 88].

For any given pixel $\overline{x}_c = (x, y)$, define $I_{\min}(x, y)$ and $I_{\max}(x, y)$ as its minimum and maximum brightness throughout the entire sequence:

$$\begin{cases} I_{\min}(x, y) & \doteq & \min\limits_t \{I(x, y, t)\} \\ I_{\max}(x, y) & \doteq & \max\limits_t \{I(x, y, t)\} \end{cases} \tag{6.2}$$

We define the shadow edge to be the locations (in space-time) where the image $I(x, y, t)$ intersects with the threshold image $I_{\text{shadow}}(x, y)$ defined as the mean value between $I_{\max}(x, y)$ and $I_{\min}(x, y)$:

$$I_{\text{shadow}}(x, y) \doteq \frac{1}{2} \left(I_{\max}(x, y) + I_{\min}(x, y) \right) \tag{6.3}$$

This may be also regarded as the zero crossings of the difference image $\Delta I(x, y, t)$ defined as follows:

$$\Delta I(x, y, t) \doteq I(x, y, t) - I_{\text{shadow}}(x, y) \tag{6.4}$$

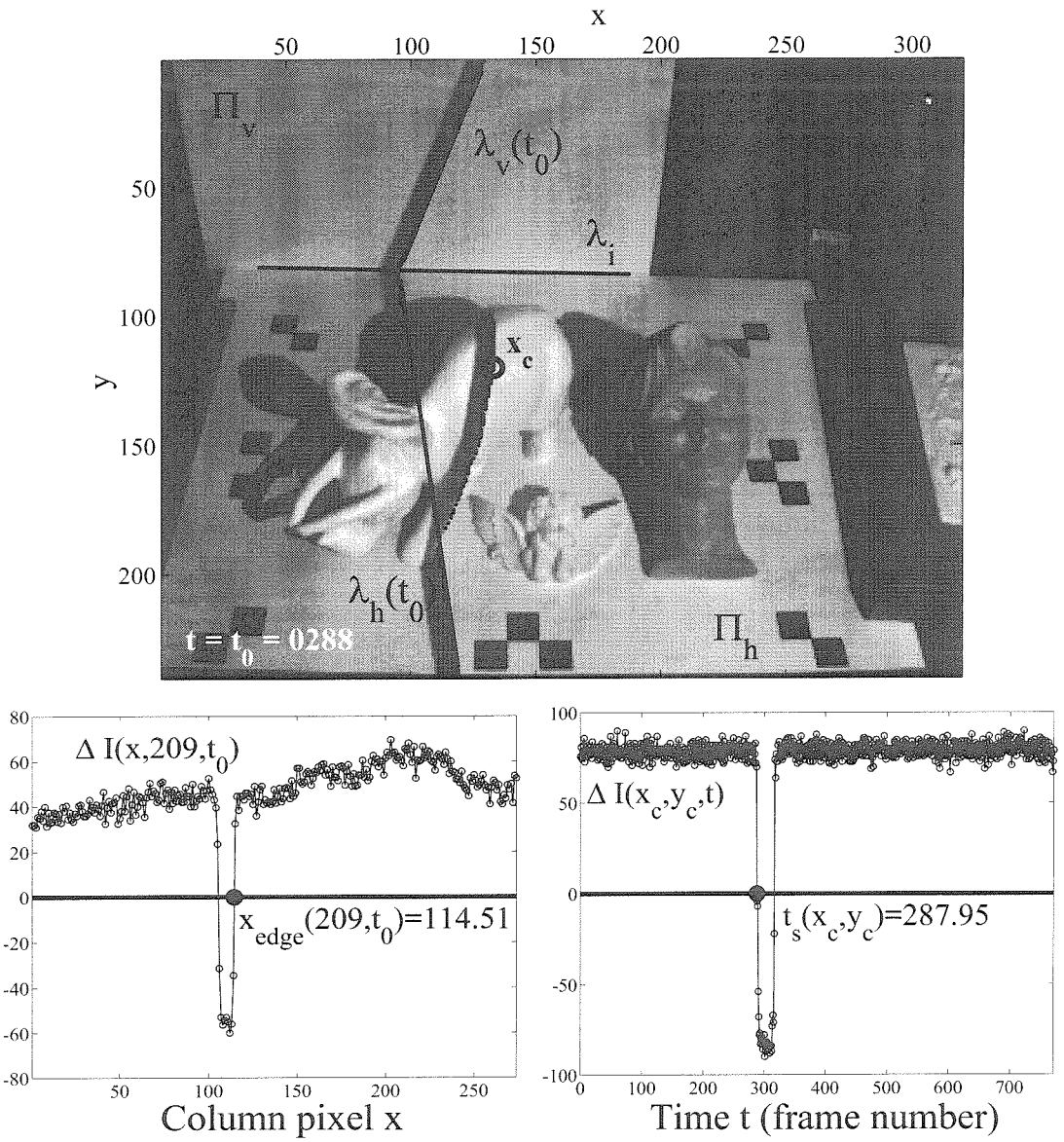


Figure 6.6: Spatial and temporal shadow localization

The two bottom plots of fig. 6.6 illustrate shadow edge detection in the spatial domain (to calculate $\lambda_h(t)$ and $\lambda_v(t)$) and in the temporal domain (to calculate $t_s(\overline{x}_c)$). The bottom-left plot shows the profile of $\Delta I(x, y, t)$ along row $y = 209$ at time $t = t_0 = 288$ versus the column pixel coordinate x. The second zero crossing of that profile corresponds to one point $\overline{x}_{\text{edge}}(t_0) = (114.51, 209)$ belonging to $\lambda_h(t_0)$, the right edge of the shadow (computed at subpixel accuracy by linear interpolation). Identical processing is applied on 39 other rows for $\lambda_h(t_0)$ and 70 rows for $\lambda_v(t_0)$ in order to retrieve the two edges (by least square line fitting across the two sets of points on the image). Similarly, the bottom-right figure shows the temporal profile $\Delta I(x_c, y_c, t)$ at the pixel $\overline{x}_c = (x_c, y_c) = (133, 120)$ versus time t (or frame number). The shadow time at that pixel is defined as the first zero crossing location of that profile: $t_s(133, 120) = 287.95$ (computed at sub-frame accuracy by linear interpolation). Notice that the right edge of the shadow corresponds to the front edge of the temporal profile, because the shadow was scanned from left to right in all experiments. Intuitively, pixels corresponding to occluded regions in the scene do not provide any relevant depth information. Therefore, we only process pixels with contrast value $I_{\text{contrast}}(x, y) \doteq I_{\max}(x, y) - I_{\min}(x, y)$ larger than a pre-defined threshold I_{thresh}. This threshold was 30 in all experiments reported in this chapter (recall that the intensity values are encoded from 0 for black to 255 for white). This threshold should be proportional to the level of noise in the image.

Due to the limited dynamic range of the camera, it is clear that one should avoid saturating the sensor, and that one would expect poor accuracy in areas of the scene that reflect little light towards the camera due to their orientation with respect to the light source and/or low albedo. Our experiments were designed to test the extent of this problem.

6.2.5 Shadow plane estimation $\Pi(t)$

Denote by $\overline{\omega}(t)$, $\overline{\lambda}_h(t)$ and $\overline{\lambda}_v(t)$ the coordinate vectors of the shadow plane $\Pi(t)$ and of the shadow edges $\lambda_h(t)$ and $\lambda_v(t)$ at time t. Since $\lambda_h(t)$ is the projection of the line

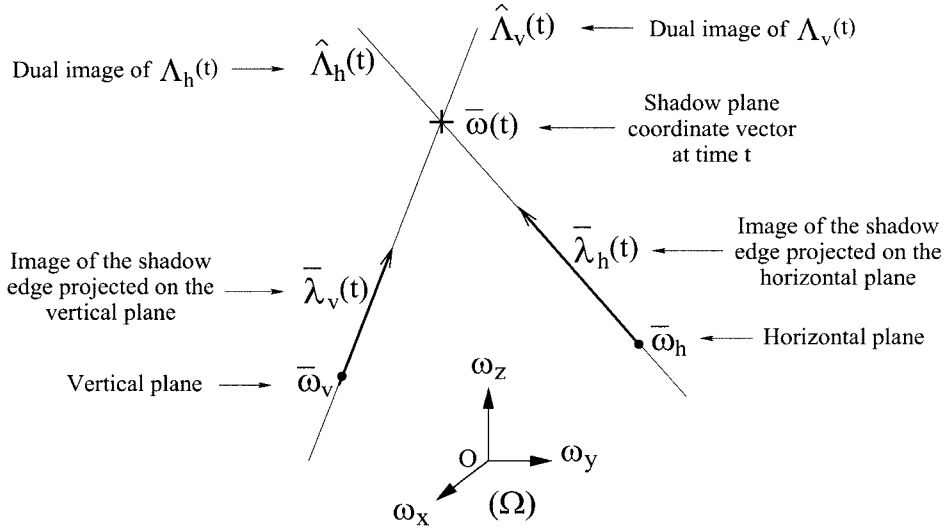


Figure 6.7: Shadow plane estimation using two planes: The coordinate vector of the shadow plane $\bar{\omega}(t)$ is the intersection point of the two dual lines $\hat{\Lambda}_h(t)$ and $\hat{\Lambda}_v(t)$ in dual-space (Ω). In presence of noise, the two lines do not intersect. The vector $\bar{\omega}(t)$ is then the best intersection point between the two lines (in the least squares sense).

of intersection $\Lambda_h(t)$ between $\Pi(t)$ and Π_h, then $\bar{\omega}(t)$ lies on the line passing through $\bar{\omega}_h$ with direction $\bar{\lambda}_h(t)$ in dual-space (from proposition 1 of section 2.2.2). That line, denoted $\hat{\Lambda}_h(t)$, is the dual image of $\Lambda_h(t)$ in dual-space (see section 2.2.2). Similarly, $\bar{\omega}(t)$ lies on the line $\hat{\Lambda}_v(t)$ passing through $\bar{\omega}_v$ with direction $\bar{\lambda}_v(t)$ (dual image of $\Lambda_v(t)$). Therefore, in dual-space, the coordinate vector of the shadow plane $\bar{\omega}(t)$ is at the intersection between the two known lines $\hat{\Lambda}_h(t)$ and $\hat{\Lambda}_v(t)$. In the presence of noise these two lines will not exactly intersect (equivalently, the 3 lines λ_i, $\lambda_h(t)$ and $\lambda_v(t)$ do not necessarily intersect at one point on the image plane, or their coordinate vectors $\bar{\lambda}_i$, $\bar{\lambda}_h(t)$ and $\bar{\lambda}_v(t)$ are not coplanar). However, one may still identify $\bar{\omega}(t)$ with the point that is the closest to the lines in the least-squares sense. The complete derivations for intersecting a set of lines in space may be found in section 4.1.2. When interesting the two lines $\hat{\Lambda}_h(t)$ and $\hat{\Lambda}_v(t)$ in space, the solution reduces to:

$$\bar{\omega}(t) = \frac{1}{2}\left(\bar{\omega}_1(t) + \bar{\omega}_2(t)\right), \tag{6.5}$$

with

$$\begin{aligned} \overline{\omega}_1(t) &\doteq \overline{\omega}_h + \alpha_h \overline{\lambda}_h(t) \\ \overline{\omega}_2(t) &\doteq \overline{\omega}_v + \alpha_v \overline{\lambda}_v(t) \end{aligned} \tag{6.6}$$

where

$$\begin{bmatrix} \alpha_h \\ \alpha_v \end{bmatrix} = \mathbf{A}^{-1}\mathbf{b} \tag{6.7}$$

where $\mathbf{A}$ is a 2×2 matrix and $\mathbf{b}$ is a 2-vector defined as follows (for clarity, the variable t is omitted):

$$\mathbf{A} \doteq \begin{bmatrix} \langle \overline{\lambda}_h, \overline{\lambda}_h \rangle & -\langle \overline{\lambda}_h, \overline{\lambda}_v \rangle \\ -\langle \overline{\lambda}_h, \overline{\lambda}_v \rangle & \langle \overline{\lambda}_v, \overline{\lambda}_v \rangle \end{bmatrix}, \quad \mathbf{b} \doteq \begin{bmatrix} \langle \overline{\lambda}_h, \overline{\omega}_v - \overline{\omega}_h \rangle \\ \langle \overline{\lambda}_v, \overline{\omega}_h - \overline{\omega}_v \rangle \end{bmatrix} \tag{6.8}$$

Note that the two vectors $\overline{\omega}_1(t)$ and $\overline{\omega}_2(t)$ are the orthogonal projections, in dual-space, of $\overline{\omega}(t)$ onto $\hat{\Lambda}_h(t)$ and $\hat{\Lambda}_v(t)$ respectively. The norm of the difference between these two vectors may be used as an estimate of the error in recovering $\Pi(t)$. If the two edges $\lambda_h(t)$ and $\lambda_v(t)$ are estimated with different reliabilities, a weighted least squares method may still be used.

Figure 6.7 illustrates the principle of shadow plane estimation in dual-space when using the two edges $\lambda_h(t)$ and $\lambda_v(t)$. This reconstruction method was used in experiments 1, 4 and 5.

Notice that the additional vertical plane Π_v enables us to extract the shadow plane location without requiring the knowledge of the light source position. Consequently, the light source is allowed to move during the scan (this may be the case of the sun, for example).

When the light source is of fixed and known location in space, the plane Π_v is not required. Then, one may directly infer the shadow plane position from the line $\lambda_h(t)$

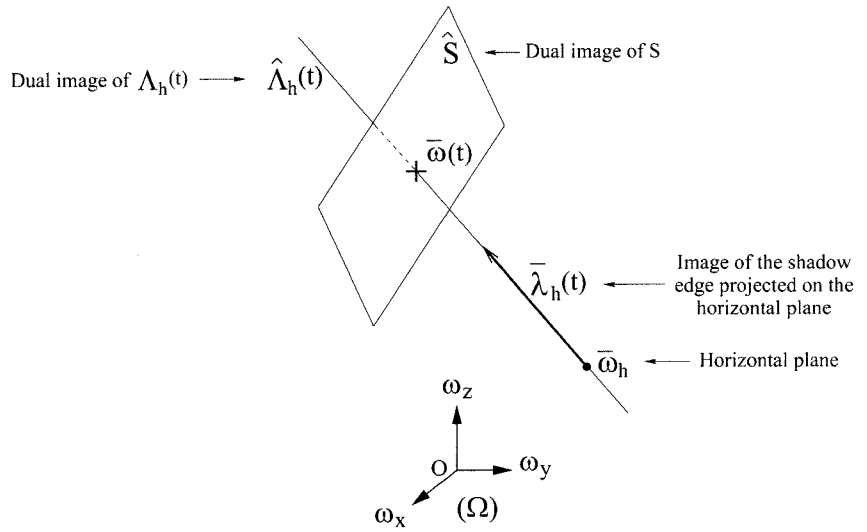


Figure 6.8: Shadow plane estimation using one plane and the light source position: In dual-space, the coordinate vector of the shadow plane $\overline{\omega}(t)$ is the intersection point of the line $\hat{\Lambda}_h(t)$ and the plane $\hat{S}$, dual image of the point light source S. This method requires the knowledge of the light source position. A light source calibration method is presented in section 6.2.3

and from the light source position S:

$$\overline{\omega}(t) = \overline{\omega}_h + \alpha_h \overline{\lambda}_h(t) \tag{6.9}$$

where

$$S \in \Pi(t) \;\Leftrightarrow\; \left\langle \overline{\omega}(t), \overline{X}_S \right\rangle = 1 \;\Leftrightarrow\; \alpha_h = \frac{1 - \left\langle \overline{\omega}_h, \overline{X}_S \right\rangle}{\left\langle \overline{\lambda}_h(t), \overline{X}_S \right\rangle} \tag{6.10}$$

where $\overline{X}_S = [X_S \;\; Y_S \;\; Y_S]^T$ is the coordinate vector of the light source S in the camera reference frame. In dual-space geometry, this corresponds to intersecting the line $\hat{\Lambda}_h(t)$ with the plane $\hat{S}$, dual image of the source point S. This process is illustrated in figure 6.8. Notice that $\left\langle \overline{\lambda}_h(t), \overline{X}_S \right\rangle = 0$ corresponds to the case where the shadow plane contains the camera center of projection O_c. This is singular configuration that makes the triangulation fail ($\|\overline{\omega}(t)\| \to \infty$). This approach requires an additional step of estimating the position of S. Section 6.2.3 describes a simple method for light source calibration. This reconstruction method was used in experiments 2 and 3.

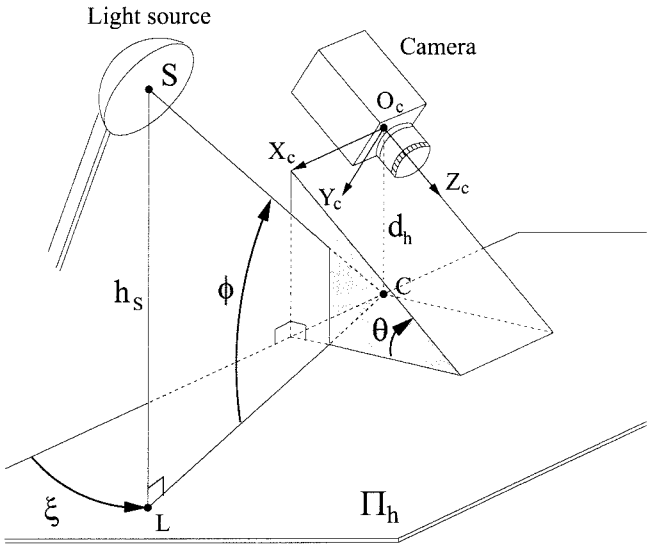


Figure 6.9: Geometric setup: The camera is positioned at a distance d_h away from the plane Π_h and tilted down towards it at an angle θ. The light source is located at a height h_S, with its direction defined by the azimuth and elevation angles ξ and ϕ in the reference frame attached to the plane Π_h. Notice that the sign of $\cos\xi$ directly relates to which side of the camera the lamp is standing: positive on the right, and negative on the left.

Claim: The quantity $1 - \langle \overline{\omega}_h, \overline{X}_S \rangle$ reduces to h_S/d_h where h_S and d_h are the orthogonal distances of the light source S and the camera center O_c to the plane Π_h (see figure 6.9).

Proof: Since $\overline{\omega}_h$ is the coordinate vector of the plane Π_h, the vector $\overline{n}_h = d_h\,\overline{\omega}_h$ is the normal vector of the plane Π_h in the camera reference frame (see figure 6.9). Let P be a point in Euclidean space (E) of coordinate vector $\overline{X}$. The quantity $d_h - \langle \overline{n}_h, \overline{X} \rangle$ is then the (algebraic) orthogonal distance of P to Π_h (positive quantity if the point P is on the side of the camera, negative otherwise). In particular, if P lies on Π_h, then $\langle \overline{n}_h, \overline{X} \rangle = d_h$, which is equivalent to $\langle \overline{\omega}_h, \overline{X} \rangle = 1$. The orthogonal distance of the light source S to Π_h is denoted h_S on figure 6.9. Therefore $h_S = d_h - \langle \overline{n}_h, \overline{X} \rangle$, or equivalently $1 - \langle \overline{\omega}_h, \overline{X}_S \rangle = h_S/d_h$. ■

According to that Claim, the constant α_h of equation 6.10 may be written as:

$$\alpha_h = \frac{h_S/d_h}{\langle \overline{\lambda}_h(t), \overline{X}_S \rangle} = \frac{1/d_h}{\langle \overline{\lambda}_h(t), \overline{X}_S/h_S \rangle} \tag{6.11}$$

This expression highlights the fact that the algebra naturally generalizes to cases where the light source is located at infinity (and calibrated). Indeed, in those cases, the ratio $\overline{X}_S/h_S$ reduces to $\overline{d}_S/\sin\phi$ where $\overline{d}_S$ is the normalized light source direction vector (in the camera reference frame) and ϕ the elevation angle of the light source with respect to the plane Π_h (defined on figure 6.9). In dual-space, the construction of the shadow plane vector $\overline{\omega}(t)$ remains the same: it is still at the intersection of $\hat{\Lambda}_h(t)$ with $\hat{S}$. The only difference is that the dual image $\hat{S}$ is a plane crossing the origin in dual-space. The surface normal of that plane is simply the vector $\overline{d}_S$.

6.2.6 Triangulation

Once the shadow time $t_s(\overline{x}_c)$ is estimated at a given pixel $\overline{x}_c = [x_c \quad y_c \quad 1]^T$ (in homogeneous coordinates), the corresponding shadow plane $\Pi(t_s(\overline{x}_c))$ is identified (its coordinate vector $\overline{\omega}_c \doteq \overline{\omega}(t_s(\overline{x}_c))$). The point P in space associated to $\overline{x}_c$ is then retrieved by intersecting $\Pi(t_s(\overline{x}_c))$ with the optical ray $(O_c, \overline{x}_c)$ (see figure 6.3):

$$Z_c = \frac{1}{\langle \overline{\omega}_c, \overline{x}_c \rangle} \implies \overline{X}_c = Z_c \overline{x}_c = \frac{\overline{x}_c}{\langle \overline{\omega}_c, \overline{x}_c \rangle}, \tag{6.12}$$

if $\overline{X}_c = [X_c \quad Y_c \quad Z_c]^T$ is defined as the coordinate vector of P in the camera reference frame. This equation was first introduced in section 2.2.3 (eq. 2.33).

Notice that the shadow time $t_s(\overline{x}_c)$ acts as an index to the shadow plane list $\Pi(t)$. Since $t_s(\overline{x}_c)$ is estimated at sub-frame accuracy, the plane $\Pi(t_s(\overline{x}_c))$ (actually its coordinate vector $\overline{\omega}_c$) results from linear interpolation between the two planes $\Pi(t_0 - 1)$ and $\Pi(t_0)$ if $t_0 - 1 < t_s(\overline{x}_c) < t_0$ and t_0 integer:

$$\overline{\omega}_c = \Delta t\, \overline{\omega}(t_0 - 1) + (1 - \Delta t)\, \overline{\omega}(t_0), \tag{6.13}$$

where $\Delta t = t_0 - t_s(\overline{x}_c)$, $0 \leq \Delta t < 1$ (see figure 6.12).

Once the range data are recovered, a mesh is generated by connecting neighboring points in triangles. The connectivity is directly given by the image: two vertices are neighbors if their corresponding pixels are neighbors in the image. In addition, since

every vertex corresponds to a unique pixel, texture mapping is also a straightforward task. Figures 6.13, 6.15 , 6.17, 6.18, 6.19, 6.20 and 6.21 show experimental results.

Similarly to stereoscopic vision, when the baseline becomes shorter, as the shadow plane moves closer to the camera center triangulation becomes more and more sensitive to noise. In the limit, if the plane crosses the origin (or equivalently $||\overline{\omega}_c|| \rightarrow \infty$) triangulation becomes impossible. This is why it is not a big loss that we cannot represent planes going through the origin with our parameterization. This observation will appear again in the section on error analysis (sec. 6.3).

6.2.7 Summary of the global geometry in dual-space

The global geometrical principle of the reconstruction technique may be summarized in a very compact fashion into a single diagram in dual-space. This diagram is shown on figure 6.10. On this figure, correspondence between Euclidean space and dual-space is given for objects in 2D (lines and points on the image plane) as well as objects in 3D (planes, lines and points in space).

Observe that the calibration of the vertical plane Π_v is also illustrated in the dual-space diagram: its coordinate vector $\overline{\omega}_v$ is at the intersection of the line $\hat{\Lambda}_i$ and the set of plane vectors orthogonal to $\overline{\omega}_h$ (defining a plane in dual-space). The line Λ_i is at the intersection of the two planes Π_h and Π_v, and its dual image $\hat{\Lambda}_i$ is uniquely defined by the horizontal plane vector $\overline{\omega}_h$ and the vector $\overline{\lambda}_i$, coordinate vector of the line λ_i observed on the image plane. This calibration process is described in section 6.2.2.

Once $\overline{\omega}_v$ is known, the shadow plane vector $\overline{\omega}(t)$ associated to the shadow edge configuration at time t is at the intersection between the two lines $\hat{\Lambda}_h(t)$ and $\hat{\Lambda}_v(t)$, dual images of $\Lambda_h(t)$ and $\Lambda_v(t)$. Those two dual lines are defined by the two reference plane vectors $\overline{\omega}_h$ and $\overline{\omega}_v$ and the direction vectors $\overline{\lambda}_h(t)$ and $\overline{\lambda}_v(t)$ (vector coordinates of the two image lines $\lambda_h(t)$ and $\lambda_v(t)$). This processing step is described in details in section 6.2.5.

The final step consisting of identifying the point P in space by intersecting the

optical ray (O_c, p) with the shadow plane Π is also illustrated on the dual-space diagram. In dual-space, that stage corresponds to finding the dual image $\hat{P}$ of P that is the unique plane in dual-space containing the point $\overline{\omega}(t)$ (shadow plane vector) with orthogonal vector $\overline{x}_c$ (homogeneous coordinate vector of the image point p). A description of this triangulation step may be found in section 6.2.6.

The other scanning setup consisting of using a single reference plane (Π_h without Π_v) with a calibrated light source is summarized on figure 6.11. The only difference between that figure on the previous one is in the procedure of estimating the shadow plane coordinate vector $\overline{\omega}(t)$. In that version, the vector $\overline{\omega}(t)$ is at the intersection of the dual line $\hat{\Lambda}_h(t)$ and the dual image of the light source $\hat{S}$. See section 6.2.5. The triangulation step remains unchanged.

Observe that on both figures 6.10 and 6.11, the dual-space diagrams are more compact than their corresponding Euclidean illustrations.

6.3 Error analysis - Design Issues

When designing the scanning system, it is important to choose a spatial configuration of the camera and the light source that maximizes the overall quality of reconstruction of the scene.

As first mentioned in section 6.2, the method proposes to associate to every pixel $\overline{x}_c$ the time instant $t_s(\overline{x}_c)$ at which the shadow crosses that particular pixel. That given time corresponds to the shadow plane $\Pi(t_s(\overline{x}_c))$ in space (of coordinate vector $\overline{\omega}_c$), used at the triangulation step to retrieve the coordinates of the point P in space (see figure 6.3). In addition, at every time instant t, a shadow plane $\Pi(t)$ is estimated based on two line segments $\lambda_h(t)$ and $\lambda_v(t)$ extracted from the image plane (see section 6.2.4).

Therefore, one clearly identifies two possible sources of error affecting the overall reconstruction: errors in localizing the two edges $\lambda_h(t)$ and $\lambda_v(t)$ leading to error in estimating the shadow plane $\Pi(t)$ (or error on the vector $\overline{\omega}(t)$), and errors in finding the shadow time $t_s(\overline{x}_c)$ (at every pixel $\overline{x}_c$) leading to an error in shadow plane

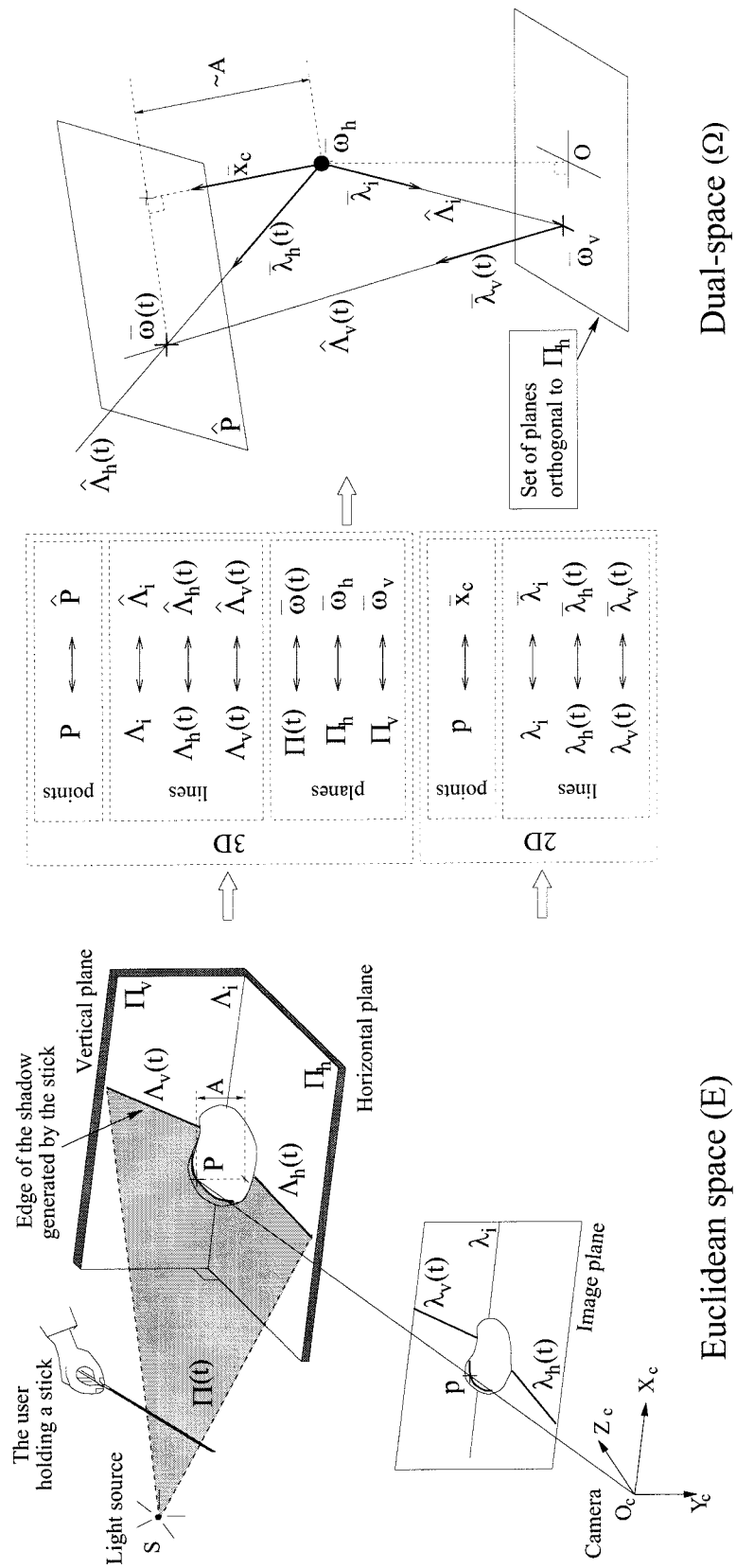


Figure 6.10: Summary of the global geometry of the scanning technique in Euclidean space and dual-space. In that setup, two background planes (Π_h and Π_v) are used and the light source is not calibrated.

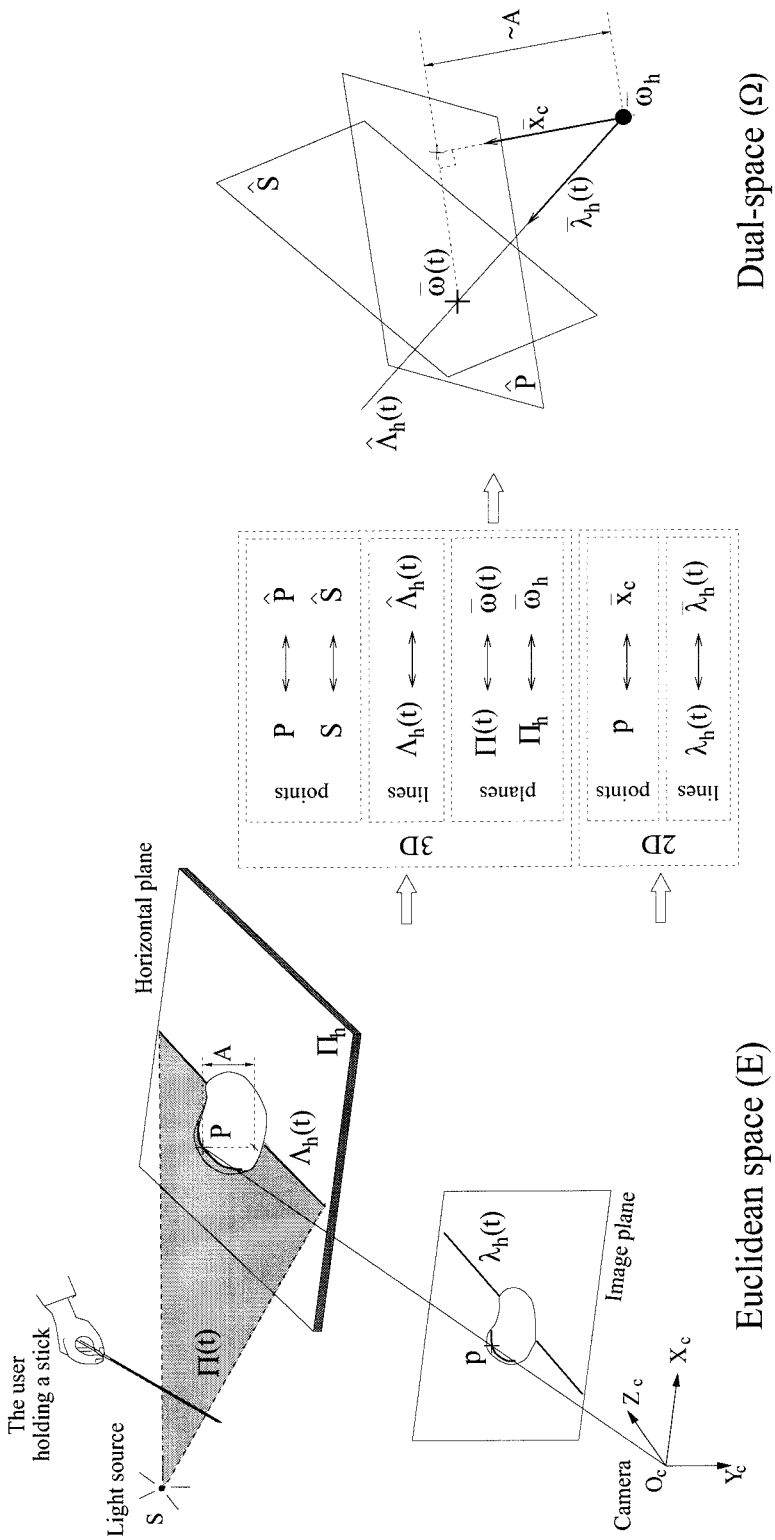


Figure 6.11: Summary of the global geometry of the scanning technique in Euclidean space and dual-space. In that setup, a single background plane is used (Π_v is not present) and the light source is assumed calibrated.

assignment.

Experimentally, we found that the error coming from spatial processing (shadow plane localization) was much smaller than the one coming from temporal processing (shadow time computation). In other words, in all the experiments we carried out, the shadow planes were localized to such a degree of accuracy that the errors induced by the noise on $\overline{\omega}_c$ were negligible compared to the errors induced by the noise on $t_s(\overline{x}_c)$. This experimental observation is reasonable because the shadow edges $\lambda_h(t)$ and $\lambda_v(t)$ are recovered by fitting lines through many points on the image plane (an order of 50 points per line) while shadow time $t_s(\overline{x}_c)$ is estimated on a basis of a single pixel. Notice that this is experiment dependent, and may very well not be true if fewer points were used to extract the shadow edges, or if the image were more noisy, or more distorted. In those cases, both error terms should be retained. In the present analysis, we propose to derive an expression of the variance of the error in depth estimation $\sigma^2_{Z_c}$ assuming that the main source of noise comes from temporal processing. In the experimental section, we verify that the final variance expression agrees numerically with accuracies achieved on real scan data.

6.3.1 Derivation of the depth variance $\sigma^2_{Z_c}$

Every pixel $\overline{x}_c$ on the image sees the shadow passing at time a $t_s(\overline{x}_c)$, called the shadow time, that is estimated through temporal processing (see section 6.2.4). This estimation is naturally subject to errors, leading to inaccuracies in the final 3D reconstruction. The purpose of that analysis is to study how damaging those errors truly are on the final structure, and quantify them. Assume that for a given pixel $\overline{x}_c$, an additive temporal error $\delta t_s(\overline{x}_c)$ is made on its shadow time estimate: $\tilde{t}_s(\overline{x}_c) = t_s(\overline{x}_c) + \delta t_s(\overline{x}_c)$. This typically leads the algorithm to assign to the pixel $\overline{x}_c$ the "wrong" shadow plane $\Pi(t_s(\overline{x}_c) + \delta t_s(\overline{x}_c))$ for the geometrical triangulation step. Equivalently, one can think that the plane $\Pi(t_s(\overline{x}_c) + \delta t_s)$ has been associated with the "wrong" pixel $\overline{x}_c$ in the image. Although it does not change anything to the problem, that way of centering the reasoning onto the shadow plane instead of the pixel actually significantly sim-

plifies the whole analysis. Indeed, as we will show in the following, if we assign the noise to the pixel location itself, the time variable can then be omitted.

To be more precise, let us first define $\overline{v}(\overline{x}_c) = [v_x(\overline{x}_c) \;\; v_y(\overline{x}_c)]^T$ to be the velocity vector of the shadow at the pixel $\overline{x}_c$ that is orthogonal to the shadow edge. Then, the closest point to $\overline{x}_c$ that has truly been lit by the shadow plane $\Pi(t_s(\overline{x}_c) + \delta t_s(\overline{x}_c))$ is $\overline{x}_c + \delta t_s(\overline{x}_c)\,\overline{v}(\overline{x}_c)$. Therefore, by picking $\overline{x}_c$ instead, we introduce an additive pixel error $\delta\overline{x}_c \doteq -\delta t_s(\overline{x}_c)\,\overline{v}(\overline{x}_c)$. This is the equivalent noise that can be attributed to the pixel location $\overline{x}_c$ before triangulation.

One can then see that this equivalent image coordinate noise is naturally related to the speed of the shadow. Indeed, even if we assume that the time estimation error δt_s is identical for every pixel in the image, the corresponding pixel error $\delta\overline{x}_c$ is generally not uniform, neither in direction, nor in magnitude. Typically, fast moving shadow regions will be subject to larger errors than slow moving shadow regions. Variations in apparent shadow speed can be caused by a change in the actual speed at which the stick is moved, a change in local surface orientation of the scene, or both.

Before triangulation, the pixel coordinates have to be normalized by the intrinsic parameters of the camera. Let us assume, for simplicity in the notation, that $\overline{x}_c = [x_c \;\; y_c \;\; 1]^T$ is directly the normalized, homogeneous coordinate vector associated to the pixel. The two coordinates x_c and y_c are affected by the error vector $\delta\overline{x}_c$ whose variance-covariance matrix is denoted $\Sigma_{\overline{x}_c}$ (a 2×2 matrix). Let us derive an expression for that matrix as a function of the image brightness noise.

Lemma: Let σ_I be the standard deviation of the image brightness noise (estimated experimentally). We can write $\Sigma_{\overline{x}_c}$ as a function of the image gradient $\overline{\nabla}I(\overline{x}_c)$ at pixel $\overline{x}_c$ at time $t = t_s(\overline{x}_c)$:

$$\Sigma_{\overline{x}_c} = \frac{\sigma_I^2}{f_c^2\,\|\overline{\nabla}I(\overline{x}_c)\|^2} \begin{bmatrix} \cos^2\varphi & \cos\varphi\sin\varphi \\ \cos\varphi\sin\varphi & \sin^2\varphi \end{bmatrix} \tag{6.14}$$

where f_c is the focal length of the camera (in pixels), $\overline{\nabla}I(\overline{x}_c)$ is the gradient vector of the image brightness at the shadow, and φ the orientation angle of that vector

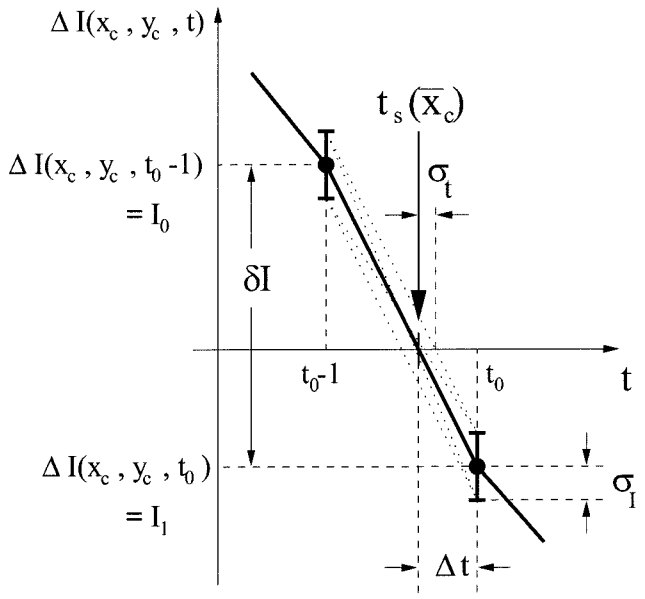


Figure 6.12: Estimation error on the shadow time: The shadow time $t_s(\overline{x}_c)$ is estimated by linearly interpolating the difference temporal brightness function $\Delta I(x_c, y_c, t)$ between times $t_0 - 1$ and t_0. The pixel noise (of standard deviation σ_I) on $I_0 \doteq \Delta I(x_c, y_c, t_0 - 1)$ and $I_1 \doteq \Delta I(x_c, y_c, t_0)$ induces errors on the estimation of Δt, or equivalently $t_s(\overline{x}_c)$. This error has variance σ_t^2.

(orientation of the shadow edge at pixel $\overline{x}_c$):

$$\overline{\nabla} I(\overline{x}_c) = \begin{bmatrix} I_x(\overline{x}_c) \\ I_y(\overline{x}_c) \end{bmatrix} = \|\overline{\nabla} I(\overline{x}_c)\| \begin{bmatrix} \cos\varphi \\ \sin\varphi \end{bmatrix} \tag{6.15}$$

where:

$$I_x(\overline{x}_c) \doteq \left.\frac{\partial I(\overline{x}, t)}{\partial x}\right|_{\overline{x}=\overline{x}_c, t=t_s(\overline{x}_c)} \tag{6.16}$$

$$I_y(\overline{x}_c) \doteq \left.\frac{\partial I(\overline{x}, t)}{\partial y}\right|_{\overline{x}=\overline{x}_c, t=t_s(\overline{x}_c)} \tag{6.17}$$

Proof of lemma (eq. 6.14): Figure 6.12 shows the principle of computing the shadow time $t_s(\overline{x}_c)$ from the difference image ΔI (refer to section 6.2.5). For clarity in the notation, define $I_0 \doteq \Delta I(x_c, y_c, t_0 - 1)$ and $I_1 \doteq \Delta I(x_c, y_c, t_0)$. Then, the

shadow time $t_s(\overline{x}_c)$ is given by:

$$t_s(\overline{x}_c) = t_0 - \Delta t \tag{6.18}$$

where:

$$\Delta t \doteq \frac{I_1}{I_1 - I_0} \tag{6.19}$$

Let σ_t^2 be the variance of the error $\delta t_s(\overline{x}_c)$ attached to the shadow time $t_s(\overline{x}_c)$. In normal sampling conditions (if the temporal brightness is sufficiently sampled within the shadow transition area), the same error is on the variable Δt, and therefore σ_t may be directly expressed as a function of σ_I, the variance of pixel noise on I_0 and I_1:

$$\sigma_t^2 = \left(\left(\frac{\partial \Delta t}{\partial I_0} \right)^2 + \left(\frac{\partial \Delta t}{\partial I_1} \right)^2 \right) \sigma_I^2 \tag{6.20}$$

$$\sigma_t^2 = \frac{I_0^2 + I_1^2}{\delta I^4} \sigma_I^2 \tag{6.21}$$

where $\delta I \doteq I_1 - I_0$ is the temporal brightness variation at the zero crossing (or equivalently at the shadow time). One may notice from equation 6.21 that, as the brightness difference δI increases, the error in shadow time decreases. That is a very intuitive behavior given that higher shadow contrasts should give rise to better accuracies. Notice however that the variance σ_t^2 is not only a function of δI but also of the absolute brightness values I_0 and I_1. One may then consider the maximum value of σ_t^2 for a fixed δI over all I_0 and I_1, subject to the constraint $I_1 = I_0 + \delta I$:

$$\sigma_t^2 = \max_{0<I_0<-\delta I} \left\{ \frac{2\,I_0^2 + 2\,I_0\,\delta I + \delta I^2}{\delta I^4} \right\} \sigma_I^2 \tag{6.22}$$

leading to the following simplified expression for σ_t^2:

$$\sigma_t^2 = \frac{\sigma_I^2}{\delta I^2} \tag{6.23}$$

To motivate that simplification, one may notice that the minimum and maximum values of σ_t^2 over all values I_0 and I_1 are quite similar anyway: $\sigma_I^2/(2\,\delta I^2)$ (minimum) and $\sigma_I^2/\delta I^2$ (maximum). The maximum may be thought as an upper bound on the error. Notice that δI is nothing but the first temporal derivative of the image brightness at the pixel $\overline{x}_c$, at the shadow time:

$$\delta I = \left.\frac{\partial I(\overline{x},t)}{\partial t}\right|_{\overline{x}=\overline{x}_c, t=t_s(\overline{x}_c)} \tag{6.24}$$

This temporal derivative may also be expressed as a function of the image gradient vector $\overline{\nabla} I(\overline{x}_c) = [I_x(\overline{x}_c) \;\; I_y(\overline{x}_c)]^T$ and the shadow edge velocity vector $\overline{v}(\overline{x}_c) = [v_x(\overline{x}_c) \;\; v_y(\overline{x}_c)]^T$:

$$\delta I = -\overline{\nabla} I(\overline{x}_c)^T\, \overline{v}(\overline{x}_c) = -I_x(\overline{x}_c)\, v_x(\overline{x}_c) - I_y(\overline{x}_c)\, v_y(\overline{x}_c) \tag{6.25}$$

By definition, the edge velocity vector $\overline{v}(\overline{x}_c)$ is orthogonal to the shadow edge. Therefore it may be also written as a direct function of the gradient vector $\overline{\nabla} I(\overline{x}_c)$:

$$\overline{v}(\overline{x}_c) = s\,\|\overline{v}(\overline{x}_c)\|\,\frac{\overline{\nabla} I(\overline{x}_c)}{\|\overline{\nabla} I(\overline{x}_c)\|} = s\,\|\overline{v}(\overline{x}_c)\| \begin{bmatrix} \cos\varphi \\ \sin\varphi \end{bmatrix} \tag{6.26}$$

where s is either $+1$ or -1 depending on the direction of motion of the edge. Therefore,

$$\delta I = (-s)\,\frac{\overline{\nabla} I(\overline{x}_c)^T\, \overline{\nabla} I(\overline{x}_c)}{\|\overline{\nabla} I(\overline{x}_c)\|}\,\|\overline{v}(\overline{x}_c)\| \tag{6.27}$$

$$\delta I = (-s)\,\|\overline{\nabla} I(\overline{x}_c)\|\,\|\overline{v}(\overline{x}_c)\| \tag{6.28}$$

Consequently, by substituting (6.28) into (6.23), we obtain a new expression for the temporal variance σ_t^2:

$$\sigma_t^2 = \frac{\sigma_I^2}{\|\overline{\nabla} I(\overline{x}_c)\|^2\,\|\overline{v}(\overline{x}_c)\|^2} \tag{6.29}$$

Then, the error vector $\delta\overline{x}_c$ transfered on the image plane is also related to the shadow

edge velocity $\overline{v}(\overline{x}_c)$ and the temporal error $\delta t_s(\overline{x}_c)$:

$$\delta\overline{x}_c = -\delta t_s(\overline{x}_c)\,\overline{v}(\overline{x}_c) \tag{6.30}$$

$$\delta\overline{x}_c = (-s)\,\|\overline{v}(\overline{x}_c)\|\,\delta t_s(\overline{x}_c)\begin{bmatrix}\cos\varphi\\ \sin\varphi\end{bmatrix} \tag{6.31}$$

Then, the variance-covariance matrix of the noise $\delta\overline{x}_c$ is (recall that $s^2 = 1$):

$$\Sigma_{\overline{x}_c} = \|\overline{v}(\overline{x}_c)\|^2\,\sigma_t^2\begin{bmatrix}\cos^2\varphi & \cos\varphi\sin\varphi\\ \cos\varphi\sin\varphi & \sin^2\varphi\end{bmatrix} \tag{6.32}$$

$$\Sigma_{\overline{x}_c} = \frac{\sigma_I^2}{\|\overline{\nabla}I(\overline{x}_c)\|^2}\begin{bmatrix}\cos^2\varphi & \cos\varphi\sin\varphi\\ \cos\varphi\sin\varphi & \sin^2\varphi\end{bmatrix} \tag{6.33}$$

Finally, note that this relation is valid if x_c is expressed in pixel coordinates. After normalization, this variance must be scaled by the square of the inverse of focal length f_c:

$$\Sigma_{\overline{x}_c} = \frac{\sigma_I^2}{f_c^2\,\|\overline{\nabla}I(\overline{x}_c)\|^2}\begin{bmatrix}\cos^2\varphi & \cos\varphi\sin\varphi\\ \cos\varphi\sin\varphi & \sin^2\varphi\end{bmatrix} \tag{6.34}$$

which ends the proof of the lemma (eq. 6.14). ■

Notice that if the shadow edge is roughly vertical on the image, one may assume $\varphi = 0$, and therefore simplify quite significantly the variance expression:

$$\Sigma_{\overline{x}_c} = \frac{\sigma_I^2}{f_c^2\,I_x^2(\overline{x}_c)}\begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix} \tag{6.35}$$

In that case, we reach the very intuitive result that only the first coordinate of $\overline{x}_c$ is affected by noise.

Since $\Sigma_{\overline{x}_c}$ in inversely proportional to the image gradient, accuracy improves with shadow edge sharpness. In addition, observe that $\Sigma_{\overline{x}_c}$ does not directly depend upon the local shadow speed. Therefore, decreasing the scanning speed would not increase

accuracy. However, for the analysis leading to equation 6.14 to remain valid, the temporal pixel profile must be sufficiently sampled within the transition area of the shadow edge (the penumbra). Therefore, if the shadow edge were sharper, the scanning should also be slower so that the temporal profile at every pixel would be properly sampled. Further discussions may be found at the end of those derivations (in the same section). Another consequence of equation 6.14 is that one may experimentally compute the variance $\Sigma_{\overline{x}_c}$ of the transfered error directly from the original input sequence: $\overline{\nabla} I(\overline{x}_c)$ is the image gradient at the shadow edge and σ_I is the pixel noise on the image. In addition, assuming that the sharpness of the shadow is approximately uniform over the entire image, then $\Sigma_{\overline{x}_c}$ may also be assumed to be uniform to a first approximation. That constitutes an additional simplification that does not have to be retained in practice.

The final expression of the variance $\sigma^2_{Z_c}$ of the error attached to the depth estimate Z_c may be written as follows:

$$\sigma^2_{Z_c} = \left(\frac{\partial Z_c}{\partial \overline{x}_c}\right) \Sigma_{\overline{x}_c} \left(\frac{\partial Z_c}{\partial \overline{x}_c}\right)^T \tag{6.36}$$

One may derive the expression for the Jacobian matrix $\left(\frac{\partial Z_c}{\partial \overline{x}_c}\right)$ from the triangulation equation 6.12:

$$Z_c = \frac{1}{\langle \overline{\omega}_c, \overline{x}_c \rangle} \implies \frac{\partial Z_c}{\partial \overline{x}_c} = Z_c^2 \begin{bmatrix} \omega_x & \omega_y \end{bmatrix} \tag{6.37}$$

where ω_x and ω_y are the two first coordinates of the shadow plane vector $\overline{\omega}_c$. This allows to expand the expression of $\sigma^2_{Z_c}$:

$$\sigma^2_{Z_c} = Z_c^4 \left(\frac{\omega_x \cos\varphi + \omega_y \sin\varphi}{f_c \, \|\overline{\nabla} I(\overline{x}_c)\|}\right)^2 \sigma_I^2 \tag{6.38}$$

This expression is directly computable from the original input sequence, and used for scan merging (described in section 6.4).

Three observations may be drawn from equation 6.38. First, since $\sigma^2_{Z_c}$ is inversely

proportional to $\|\overline{\nabla} I(\overline{x}_c)\|^2$, the reconstruction accuracy increases with the sharpness of the shadow boundary. This behavior has been observed in past experiments, and discussed in [73]. This explains why scans obtained using the sun (experiments 4 and 5) are more noisy that those with a desk lamp (as the penumbra is larger with the sun by a factor of approximately 5). Second, notice that $\sigma^2_{Z_c}$ is proportional to $\|\overline{\omega}_c\|^2$ (through the terms ω_x^2 and ω_y^2), or, equivalently, inversely proportional to the square of the distance of the shadow plane to the camera center O_c. Therefore, as the shadow plane moves closer to the camera, triangulation becomes more and more sensitive to noise (see discussion in section 6.2.6). The third observation is less intuitive: one may notice that σ_{Z_c} does not explicitly depend on the local shadow speed at $\overline{x}_c$, at time $t = t_s(\overline{x}_c)$. Therefore, decreasing the scanning speed would not increase accuracy. However, for the analysis leading to equation 6.38 to remain valid, the temporal pixel profile must be sufficiently sampled within the transition area of the shadow edge (the penumbra). Therefore, if the shadow edge were sharper, the scanning should also be slower so that the temporal profile at every pixel would be properly sampled. Decreasing further the scanning speed would benefit the accuracy only if the temporal profile were appropriately low-pass filtered or otherwise interpolated before extraction of $t_s(\overline{x}_c)$. This is an issue for future research.

An experimental validation of the variance expression (6.38) is reported in section 6.6 (figure 6.14).

6.3.2 System design issues

In the case where the light source position is known, it is possible to write the "average" depth variance as a direct function of the variables defining the geometry of the system. For that purpose, let us consider the scanning setup as it is presented on figure 6.9 where scanning is done roughly vertically. In that case, $\varphi \approx 0$, and $I_y^2(\overline{x}_c) \ll I_x^2(\overline{x}_c)$ (see figure 6.14). Then, the depth variance expression (6.38) may be

further simplified to:

$$\sigma_{Z_c}^2 \approx \frac{Z_c^4\,\omega_x^2}{f_c^2\,I_x^2(\overline{x}_c)}\,\sigma_I^2 \tag{6.39}$$

It appears then that the first coordinate ω_x of the shadow plane vector $\overline{\omega}_c$ carries most of the variations in accuracy of reconstruction within a given scan. When designing the scanning system, an important issue is to choose the spatial configurations of the camera and the light source that maximize the overall quality of reconstruction, or equivalently minimize $|\omega_x|$. In order to address this issue, it is necessary to further expand the term ω_x, and study its dependence upon the geometrical variables characterizing the system. Since the light source position is of interest here, let us consider the case where a single plane Π_h is used for scanning. In that case, the shadow plane vector $\overline{\omega}_c$ appears as a function of the light source position vector $\overline{X}_S$, as stated by equation 6.9. Assume that $\overline{\lambda}_h = [\lambda_x \quad \lambda_y \quad \lambda_z]^T$ is normalized such that $\lambda_x = 1$. In addition, assume that the (O_c, X_c) axis of the camera is approximately parallel to the plane Π_h (as suggested in figure 6.9). This implies that the first coordinate of $\overline{\omega}_h$ is zero. Then, the first coordinate ω_x of $\overline{\omega}_c$ reduces to:

$$\omega_x = \frac{1 - \langle \overline{\omega}_h, \overline{X}_S \rangle}{\langle \overline{\lambda}_h, \overline{X}_S \rangle} = \frac{h_S/d_h}{\langle \overline{\lambda}_h, \overline{X}_S \rangle} \tag{6.40}$$

where d_h and h_S are the respective orthogonal distances of the camera center O_c and the light source S to the plane Π_h.

For simplification purposes, let us assume that the shadow edge λ_h appears vertically on the image plane, and let x be its horizontal position (on the image). As the shadow moves from left to right, x varies from negative values to positive values, crossing zero when the shadow is at the center of the image. In that specific scenario, the shadow edge vector reduces to: $\overline{\lambda}_h = [1 \quad 0 \quad -x]^T$ simplifying equation 6.40:

$$\frac{1}{\omega_x} = \frac{d_h}{h_S}(X_S - x\,Z_S) \tag{6.41}$$

The problem of maximizing the reconstruction quality corresponds then to maximiz-

ing $|1/\omega_x|$. Since that quantity is function of the shadow edge location x, we may observe that the accuracy of reconstruction is not uniform throughout the scene for a given scan (unless the depth of the light source in the camera reference frame is zero: $Z_S = 0$). A better understanding of that relation may be achieved by expressing the light source coordinate vector $\overline{X}_S$ as a function of the angular coordinates θ, ϕ, and ξ defining the mutual positions of the camera and the light source with respect to the plane Π_h (see figure 6.9):

$$\overline{X}_S = \begin{bmatrix} X_S \\ Y_S \\ Z_S \end{bmatrix} = \begin{bmatrix} h_S \frac{\cos\xi}{\tan\phi} \\ -h_S \frac{\sin\theta\sin\xi}{\tan\phi} + (d_d - h_S)\cos\theta \\ h_S \frac{\cos\theta\sin\xi}{\tan\phi} + (d_d - h_S)\sin\theta \end{bmatrix} \tag{6.42}$$

Following this notation, the inverse of ω_x may be written as follows:

$$\frac{1}{\omega_x} = d_h \left(\frac{\cos\xi}{\tan\phi} - x \left(\frac{\cos\theta\sin\xi}{\tan\phi} + \frac{d_h - h_S}{h_S}\sin\theta \right) \right) \tag{6.43}$$

Since during scanning, the shadow edge coordinate x spans a range of values going from negative to positive values, we may consider that taking $x = 0$ gives us an indication of the "average" reconstruction quality:

$$\left.\frac{1}{\omega_x}\right|_{\text{average}} \approx \left.\frac{1}{\omega_x}\right|_{x=0} = d_h \frac{\cos\xi}{\tan\phi} \tag{6.44}$$

Equation 6.39 may then be used to infer an expression for the "average" depth variance:

$$\sigma^2_{Z_c}\big|_{\text{average}} \approx \frac{Z_c^4}{d_h^2} \frac{\tan^2\phi}{\cos^2\xi} \frac{\sigma_I^2}{f_c^2 I_x^2(\overline{x}_c)} \tag{6.45}$$

A next simplification step may be applied, by observing that the average depth of the scene is approximately related to the height d_h and the tilt angle θ of the camera

through the following expression:

$$Z_c|_{\text{average}} \approx \frac{d_h}{\sin\theta} \tag{6.46}$$

That relation leads us to a new expression for the "average" σ_{Z_c}:

$$\sigma_{Z_c}|_{\text{average}} \approx d_h \frac{\tan\phi}{\sin^2\theta\ |\cos\xi|} \frac{\sigma_I}{f_c\ |I_x(\overline{x}_c)|} \tag{6.47}$$

Notice that this quantity may be computed prior to scanning knowing the geometrical configuration of the system. From that expression, it is also possible to identify optimal configurations of the camera and the light source that maximize the overall quality of the reconstruction. In order to maximize the overall reconstruction quality, the position of the light source needs then to be chosen so that the norm of the ratio $\tan\phi/\cos\xi$ is minimized. Therefore, the two optimal values for the azimuth angle are $\xi = 0$ and $\xi = \pi$ corresponding to positioning the lamp either to the right ($\xi = 0$) or to the left ($\xi = \pi$) of the camera (see figure 6.9). Regarding the elevation angle ϕ, it would be beneficial to make $\tan\phi$ as small as possible. However, making ϕ arbitrarily small is not practically possible. Indeed, setting $\phi = 0$ would constrain the light source to lie on the plane Π_h which would then drastically reduce the effective coverage of the scene due to large amount of self-shadows cast on the scenery. A reasonable trade-off for ϕ is roughly between 60 and 70 degrees. Regarding the camera position, it would also be beneficial to make $\sin\theta$ as large as possible (ideally equal to one). However, it is very often not practical to make θ arbitrarily close to $\pi/2$. Indeed, having $\theta = \pi/2$ brings the reference plane Π_h parallel to the image plane. Then, standard visual camera calibration algorithms are known to fail (due to lack of depth perspective in the image). In most experiments, we set θ to roughly $\pi/4$.

Once again, for validation purposes, we may use equation 6.47 to estimate the reconstruction error of the scans performed in experiment 3 (figure 6.19). From a set of 10 images, we first estimate the average image brightness noise ($\sigma_I = 2$), and the shadow edge sharpness ($\|\overline{\nabla} I\| \approx 50$). After camera and light source calibration,

the following set of parameters is recovered: $f_c = 428$ pixels, $d_h = 22$ cm, $\theta = 39.60$ degrees, $h_S = 53.53$ cm, $\xi = -4.91$ degrees and $\phi = 78.39$ degrees. Equation 6.47 returns then an estimate of the reconstruction error ($\sigma_{Z_c} \approx 0.2$ mm) very close to the actual error experimentally measured on the final reconstructed surface (between 0.1 mm and 0.2 mm). The first expression given in equation 6.38 may also be used for obtaining a more accurate estimate of σ_{Z_c} specific to every point in the scene.

6.4 A simple merging technique

The range data can only be retrieved at pixels corresponding to regions in the scene illuminated by the light source and seen by the camera. In order to obtain better coverage of the scene, one may take multiple scans of the same scene with the light source at different locations each time, while keeping the camera position fixed. Consider the case of two scans with the lamp first on the right, and then on the left of the camera (see figure 6.13). Assume that at a given pixel $\overline{x}_c$ on the image, two shadow planes are available from the two scans: Π_c^L and Π_c^R. Denote by $\overline{\omega}_c^L$ and $\overline{\omega}_c^R$ their respective coordinate vectors. Then, two estimates Z_c^L and Z_c^R of the corresponding depth at $\overline{x}_c$ are available (from equation 6.12):

$$\begin{cases} Z_c^L &= 1/\left\langle \overline{\omega}_c^L, \overline{x}_c \right\rangle \\ Z_c^R &= 1/\left\langle \overline{\omega}_c^R, \overline{x}_c \right\rangle \end{cases} \tag{6.48}$$

One may then calculate the depth estimate at $\overline{x}_c$ by taking a weighted average of Z_c^L and Z_c^R:

$$Z_c \doteq \alpha_L \, Z_c^L + \alpha_R \, Z_c^R \tag{6.49}$$

where the weights α_L and α_R are computed based on the respective reliabilities of the two depth estimates. Assuming that Z_c^L and Z_c^R are random variables with independent noise terms, they are optimally averaged (in the minimum variance sense)

using the inverse of the variances as weights [89]:

$$\frac{\alpha_L}{\alpha_R} = \frac{\sigma_R^2}{\sigma_L^2} \implies \begin{cases} \alpha_L = \sigma_R^2/(\sigma_R^2 + \sigma_L^2) \\ \alpha_R = \sigma_L^2/(\sigma_R^2 + \sigma_L^2) \end{cases} \tag{6.50}$$

where σ_L^2 and σ_R^2 are the variances of the error attached to Z_c^L and Z_c^R respectively.

A sensitivity analysis of the method described in section 6.3 provides expressions for those variances (given in equation 6.38). This technique was used in experiment 1 and experiment 3 (see figures 6.13 and 6.16).

6.5 Real-time implementation

We implemented a real-time version of our 3D scanning algorithm in collaboration with Silvio Savarese of the university of Naples, Italy. In that implementation the process is divided into two main steps. In the first step, the minimum and maximum images $I_{\min}(x, y)$ and $I_{\max}(x, y)$ (eq. 6.2) are computed through a first fast shadow sweep over the scene (with no shadow edge detection). That step allows to pre-compute the threshold image $I_{\text{shadow}}(x, y)$ (eq. 6.3) which is useful to compute in real-time the difference image $\Delta I(x, y, t)$ (eq. 6.4) during the next stage: the scanning procedure itself. During scanning, temporal and spatial shadow edge detections are performed as described in section 6.2.4: As a new image $I(x, y, t_0)$ is acquired at time $t = t_0$, the corresponding difference image $\Delta I(x, y, t_0)$ is first computed. Then, a given pixel (x_c, y_c) is selected as a pixel lying on the edge of the shadow if $\Delta I(x_c, y_c, t)$ crosses zero between times $t = t_0 - 1$ and $t = t_0$. In order to make that decision, and then compute its corresponding sub-frame shadow time $t_s(x_c, y_c)$, only the previous image difference $\Delta I(x, y, t_0 - 1)$ needs to be stored in memory. Once a pixel (x_c, y_c) is activated and its sub-frame shadow time $t_s(x_c, y_c)$ computed, one may directly identify its corresponding shadow plane Π by linear interpolation between the current shadow plane $\Pi(t_0)$ and the previous one $\Pi(t_0 - 1)$ (see sec. 6.2.5). Therefore, the 3D coordinates of the point may be directly computed by triangulation (see sec. 6.2.6). As a result, in that implementation, neither the shadow times $t_s(x, y)$, nor

the entire list of shadow planes $\Pi(t)$ need to be stored in memory, only the previous difference image $\Delta I(x, y, t_0 - 1)$ and the previous shadow plane $\Pi(t_0 - 1)$. In addition, scene depth map (or range data) is computed in real-time. The final implementation that we designed also takes advantage of possible multiple passes of the shadow edge over a given pixel in the image by integrating all the successive depth measurements together based on their relative reliabilities (equations 6.48, 6.49 and 6.50 in section 6.4). Details of the implementation may be found in [90].

The real-time program was developed under Visual C++ and works at 30 frames a second on images of size 320×240 on a Pentium 300MHz machine: it takes approximately 30 seconds to scan a scene with a single shadow pass (i.e., $30 \times 30 = 900$ frames), and between one and two minutes for a refined scan using multiple shadow passes. The system uses the PCI frame grabber PXC200 from Imagenation, a NTSC black and white SONY XC-73/L camera (1/3 inch CCD) with a 6mm COSMICAR lens (leading to a 45^o horizontal field of view). Source code (`matlab` for calibration and `C` for scanning) and complete hardware references and specifications are available online at `http://www.vision.caltech.edu/bouguetj/ICCV98`. At the same location, a short demonstration movie (one minute long) of the working system is also available.

6.6 Experimental results

6.6.1 Calibration accuracy

Camera calibration. For a given setup, we acquired 5 images of the checkerboard pattern (see figure 6.4-right), and performed independent calibrations on them. The checkerboard, placed at different positions in each image, consisted of 187 visible corners on a 16×10 grid. We computed both mean values and standard deviations of all the parameters independently: the focal length f_c, radial distortion factor k_c and ground plane position Π_h. Regarding the ground plane position, it is convenient to look at its distance d_h to the camera origin O_c and its normal vector $\overline{n}_h$ expressed in the camera reference frame (recall: $\overline{\omega}_h = \overline{n}_h / d_h$). The following table summarizes the calibration results:

Parameters	Estimates	Relative errors
f_c (pixels)	426.8 ± 0.8	0.2%
k_c	-0.233 ± 0.002	1%
d_h (cm)	112.1 ± 0.1	0.1%
$\overline{n}_h$	$\begin{pmatrix} -0.0529 \pm 0.0003 \\ 0.7322 \pm 0.0003 \\ 0.6790 \pm 0.0003 \end{pmatrix}$	0.05%
$\overline{\omega}_h$ (m^{-1})	$\begin{pmatrix} -0.0472 \pm 0.0003 \\ 0.653 \pm 0.006 \\ 0.606 \pm 0.006 \end{pmatrix}$	0.1%

Lamp calibration. Similarly, we collected 10 images of the pencil shadow (like figure 6.5-top-right) and performed calibration of the light source on them. See section 6.2.3. Notice that the points $\overline{b}$ and t_s were manually extracted from the images. Define $\overline{X}_S$ as the coordinate vector of the light source in the camera reference frame. The following table summarizes the calibration results obtained for the setup shown in figure 6.5 (refer to figure 6.9 for notation):

Parameters	Estimates	Relative errors
$\overline{S}_c$ (cm)	$\begin{pmatrix} -13.7 \pm 0.1 \\ -17.2 \pm 0.3 \\ -2.9 \pm 0.1 \end{pmatrix}$	$\approx 2\%$
h_S (cm)	34.04 ± 0.15	0.5%
ξ (degrees)	146.0 ± 0.8	0.2%
ϕ (degrees)	64.6 ± 0.2	0.06%

The estimated lamp height agrees with the manual measure (with a ruler) of 34 ± 0.5 cm.

This accuracy is sufficient for not inducing any significant global distortion onto the final recovered shape, as we discuss in the next section.

6.6.2 Scene reconstructions

Experiment 1 - Indoor scene: We took two scans of the same scene with the desk lamp first on the right side and then on the left side of the camera. The two resulting meshes are shown on the top row of figure 6.13. The meshes were then merged together following the technique described in section 6.4. The bottom figure shows the resulting mesh composed of 66,579 triangles. We estimated the surface error (σ_{Z_c}) to approximately .7 mm in standard deviation over 50 cm large objects, leading to a relative reconstruction error of 0.15%. The white holes in the mesh images correspond to either occluded regions (not observed from the camera, or not illuminated) or very low albedo areas (such as the black squares on the horizontal plane). There was no significant global deformation in the final structured surface: after fitting a quadratic model through sets of points on the two planes, we only noticed a decrease of approximately 5% in standard deviation of the surface error. One may therefore conclude that the calibration procedure returns sufficiently accurate estimates. The original input sequences were respectively 665 and 501 frames long, each image being 320×240 pixels large, captured with a grayscale camera.

Figure 6.14 reports a comparison test between the theoretical depth variances

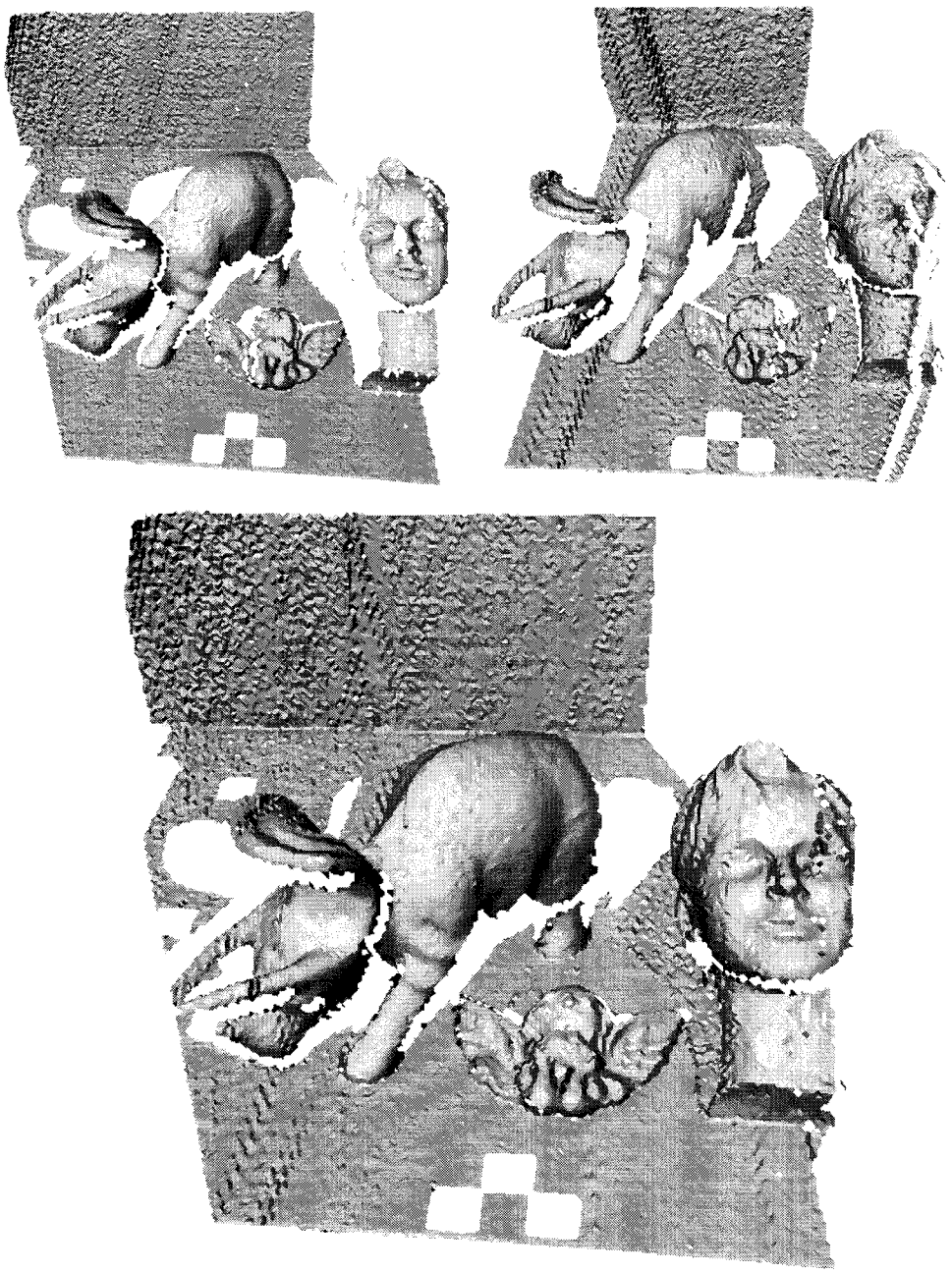
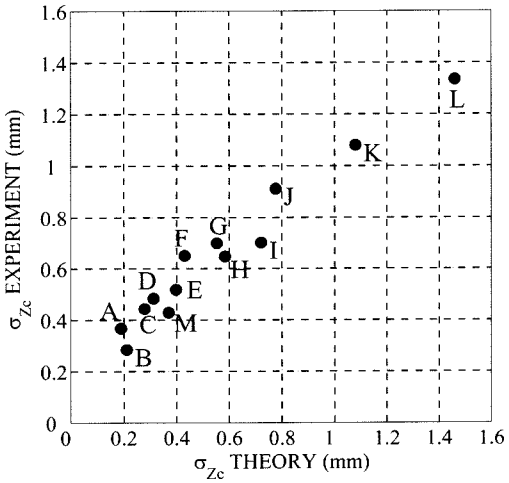
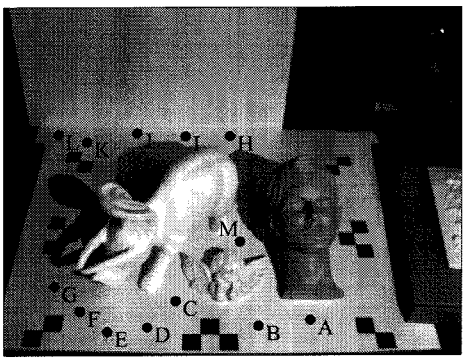


Figure 6.13: Experiment 1 - Indoor scene: The top figures are two scans of the scene with the light source at two different locations (on the right, and on the left of the camera). The bottom figure is the resulting scene surface after merging of the two scans.





p	$\overline{\nabla I}$	$[\omega_x \ \omega_y]^T$ (in m^{-1})	Z_c (in mm)	σ_{Z_c} theory (in mm)	σ_{Z_c} exp. (in mm)
A	71.5 18.0	1.6591 0.2669	1332.4	0.19	0.37
B	69.0 12.0	1.7755 0.3762	1317.2	0.21	0.28
C	61.0 11.0	1.9639 0.3576	1355.6	0.28	0.44
D	52.0 12.0	2.0788 0.3071	1300.0	0.31	0.48
E	40.5 14.0	2.2454 0.2170	1286.2	0.40	0.52
F	42.0 12.0	2.3455 0.1606	1318.6	0.43	0.65
G	37.5 10.0	2.5048 0.1101	1363.4	0.55	0.70
H	46.5 9.0	1.7752 0.3776	1800.8	0.58	0.65
I	38.5 9.5	1.8700 0.3608	1789.6	0.72	0.70
J	38.0 9.5	2.0038 0.3491	1786.1	0.78	0.91
K	28.0 7.5	2.1815 0.2523	1749.7	1.08	1.08
L	21.5 7.0	2.2834 0.1953	1769.0	1.46	1.34
M	51.0 10.0	1.7905 0.3765	1495.2	0.37	0.43

Figure 6.14: Comparison of measured and predicted reconstruction error σ_{Z_c}.

obtained from expression (6.38) and that computed from the reconstructed surface. This test was done on the first scan of the scene shown on figure 6.13-top-left. In that test, we experimentally compute the standard deviation σ_{Z_c} of the error on the depth estimate Z_c at 13 points $p = (A, B, \ldots, M)$ picked randomly on the horizontal plane Π_h of the scan data shown on figure 6.13-top-left. Figure 6.14-top-right shows the positions of those points in the scene. The standard deviation σ_{Z_c} at a given point p in the image is experimentally calculated by first taking the 9×9 pixel neighborhood around p resulting into a set of 81 points in space that should lie on Π_h. We then fit a plane across those 81 points (in the least squares sense) and set σ_{Z_c} as the standard deviation of the residual algebraic distances of the entire set of points to this best fit plane. The experimental estimates for σ_{Z_c} are reported in the last column of the table (in mm). The second last column reports the corresponding theoretical estimates of σ_{Z_c} (in mm) computed using equation 6.38. The terms involved in that equation are also given: $\overline{\nabla} I$ (in units of brightness per pixel), $[\omega_x \;\; \omega_y]^T$ (in m^{-1}) and Z_c (in mm). The image noise was experimentally estimated to $\sigma_I = 2$ brightness values (calculation based on 100 acquired images of the same scene), and the focal value used was $f_c = 426$ pixels. See sec. 6.3 for a complete description of those quantities. The top-left figure shows a plot of the theoretical standard deviations versus the experimental ones. Observe that the theoretical error model captures quite faithfully the actual variations in accuracy of reconstruction within the entire scene: as the point of interest moves from the left to the right part of the scenery, accuracy increases due to sharper edges, and a smaller shadow plane vector $\overline{\omega}_c$; in addition, deeper areas in the scene are more noisy mainly because of larger absolute depths Z_c and shallower shadow edges (smaller $\|\overline{\nabla} I\|$). We conclude from that experiment that equation 6.38 returns a valid estimate for σ_{Z_c}.

Experiment 2 - The plane/ball/corner scene: Figure 6.15 reports the scanning results achieved on a scene composed of simple geometrical objects (scene already seen on figure 6.1-cd). The original sequence was composed of 270 frames, 320×240 pixels each. Regarding the general setup, the camera was positioned at a distance of $d_h = 16.7$ cm away from the desk plane, tilted down by $\theta = 41.3$ degrees. The

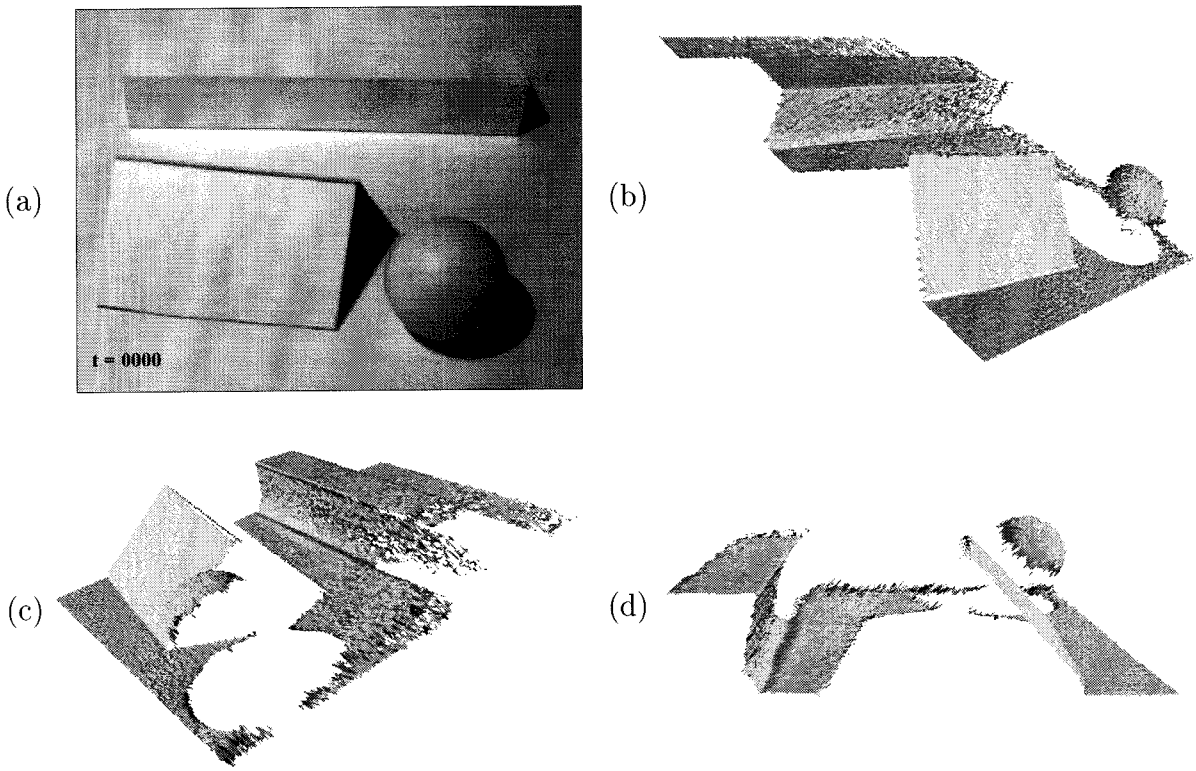


Figure 6.15: Experiment 2 - The plane/ball/corner scene: (a) The initial image of the scene before shadow scanning, and (b), (c) and (d) are different views of the mesh generated from the cloud of points obtained after triangulation.

calibrated light source was at height $h_S = 37.7$ cm, on the left of the camera at angles $\xi = 157.1$ degrees, and $\phi = 64.8$ degrees (see figure 6.9). Notice that in the experiment, the lamp had its reflector on (as seen on figure 6.1-a). As the scanning progresses from the left to the right part of the scene, the shadow boundary moves away from the light source, bringing the shadow plane $\Pi(t)$ closer to the camera center of projection O_c. This explains why the surface noise is larger on the right part of the scene, especially noticeable on figure 6.1-c. For more details on that issue, refer to the error analysis section 6.3.

In this experiment, we evaluated the accuracy of reconstruction based on the sizes and shapes of the plane at the bottom left corner and the corner object on the top of the scene (see figure 6.15a).

Planarity of the plane: We fit a plane across the points lying on the planar patch and estimated the standard deviation of the set of residual distances of the points to the plane to 0.23 mm. This corresponds to the granularity (or roughness) noise on the planar surface. The fit was done over a surface patch of approximate size 4 cm $\times$ 6 cm. This leads to a relative non planarity of approximately 0.23mm/5cm = 0.4%. To check for possible global deformations due to errors in calibration, we also fit a quadratic patch across those points. We only noticed a decrease of approximately 6% in residual standard deviation after quadratic warping. This leads us to believe that global geometric deformations are negligible compared to local surface noise. In other words, one may assume that the errors of calibration do not induce significant global deformations on the final reconstruction.

Geometry of the corner: We fit 2 planes to the corner structure, one corresponding to the top surface (the horizontal plane) and the other one to the frontal surface (vertical plane). We estimated the surface noise of the top surface to only 0.125 mm, and that of the frontal face to 0.8 mm (almost 7 times larger). This noise difference between the two planes can be observed on figure 6.15. Once again, after fitting quadratic patches to the two planar portions, we did not notice any significant global geometric distortion in the scene (from planar to quadratic warping, the residual noise decreased by only 5% in standard deviation). From the reconstruction, we estimated the height H and width D of the right angle structure, as well as the angle ψ between the two reconstructed planes, and compared them to their true values:

Parameters	Estimates	True values	Relative errors
H (cm)	2.57 ± 0.02	2.65 ± 0.02	3%
D (cm)	3.06 ± 0.02	3.02 ± 0.02	1.3%
ψ (degrees)	86.21	90	1%

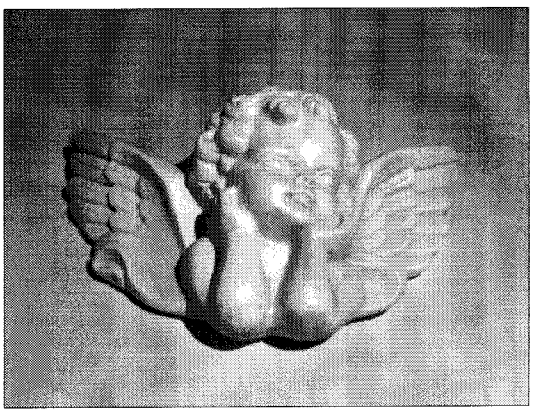
We can recover the height and width of the right angle structure, as well as

the angle between the two reconstructed planes: The reconstructed width D of the object is 3.06 ± 0.02 cm, to be compared to its real width 3.02 ± 0.02 cm (i.e., a relative error of 1.3%). For the height, we measure on the reconstruction 2.57 ± 0.02 cm, to be compared with the real height of the object being 2.65 ± 0.02 cm (i.e., relative error of 3%). Finally, the angle between the two reconstructed planes is estimated to be 86.21 degrees (the true value being 90 degrees), which means a deformation error of 4 degrees, or 4% relative angular deformation.

On average, the overall reconstructed structure does not have any major noticeable global deformation (it seems that the calibration process gives good enough estimates). The most noticeable source of errors is the surface noise due to local image processing. A figure of merit to keep in mind is a surface noise between 0.1 mm (for planes roughly parallel to the desk) and 0.8 mm (for frontal plane in the right corner). In most portions of the scene, the errors are of the order of 0.3 mm, i.e., less than 1%.

Experiment 3 - The angel scene: In this third experiment (shown on figure 6.16), we took two scans of a small sculpture of an angel, and then merged them together (to obtain a better coverage of the surface, and a cleaner reconstruction). Similarly to the second experiment, we used a scanning scenario where the light source is calibrated (and a single plane is used for background). Notice from figures 6.16a and 6.16b that between the two scans, we moved the lamp source from the left to the right side of the camera, keeping the camera position unchanged. Figures 6.16c and 6.16d show the meshes resulting from both scans. Notice that, as expected, the part of the scenery located on the side on the lamp (with respect to the camera) is always the most accurately reconstructed: left side for figure 6.17c, and right side for figure 6.17d. For a precise justification, refer to section 6.3 where a complete noise sensitivity analysis of the method is carried out.

Figure 6.17 shows different views of the resulted 3D mesh of the angel after merging of the two scans. The merging was performed according to the method described in section 6.4: at every pixel in the image, each scan gives one depth estimate, to-

(a)  (b) 

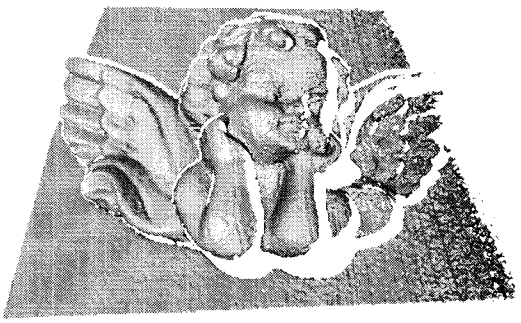
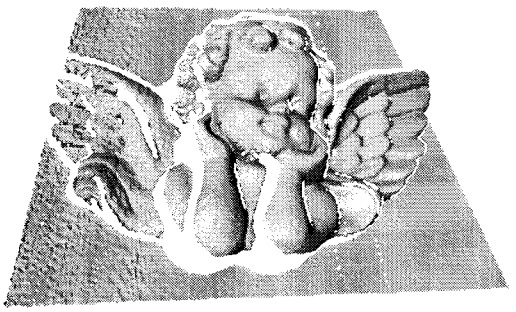
(c)  (d) 

Figure 6.16: Experiment 3 - The angel scene: In that experiment, we took two scans of the angel with the lamp first on the left side (a) and then on the right side (b) of the camera. The resulted meshes after scanning are shown on figures (c) and (d) (for respectively left and right illuminations).

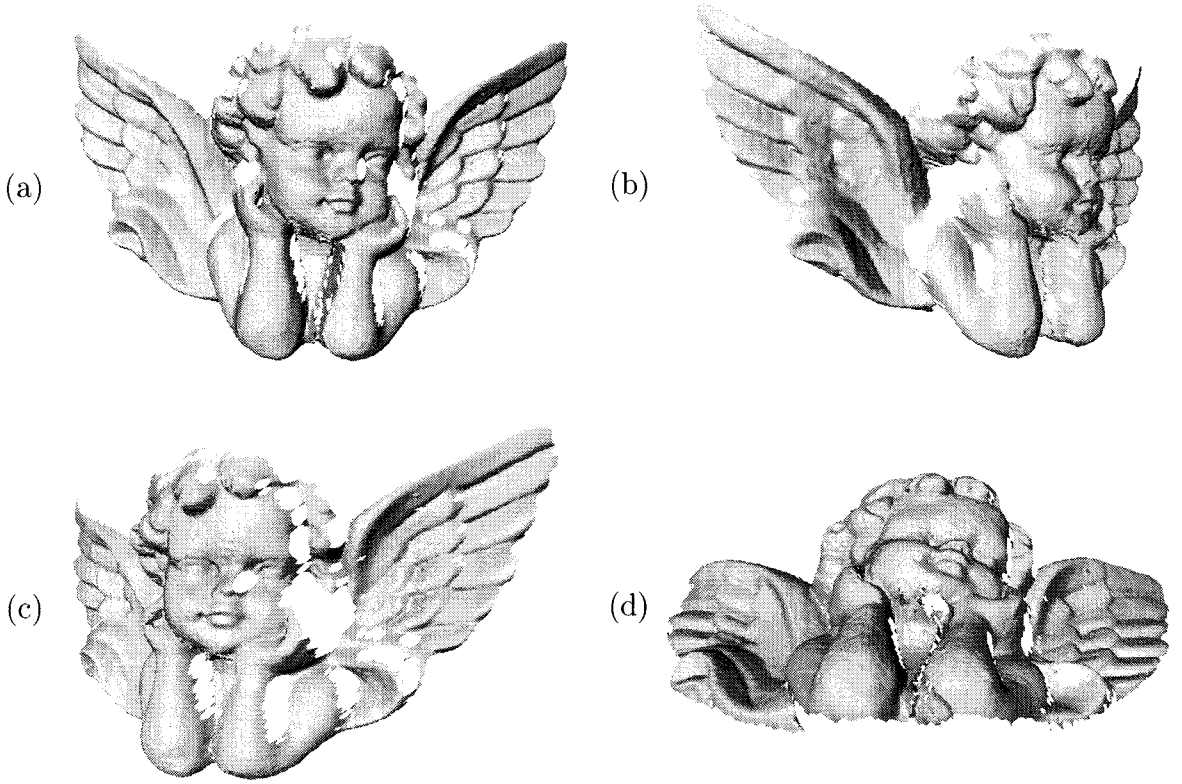


Figure 6.17: Experiment 3 - The final reconstruction: The resulting 3D model of the angel after merging of the two meshes 6.16c and 6.16d generated from the two scans independently. It is composed of 47076 triangles. Notice that most of the surface of the object is nicely reconstructed, except for few occluded portions of the scene (not observed from the camera or not illuminated by the camera) leaving small white holes here and there (look for example at the right side of the nose). Notice the very small surface noise: we estimated it to 0.09 mm throughout the entire reconstructed surface.

gether with an uncertainty measure (the variance of the estimation noise). Both depths are then combined by weighted average, resulting into the the final depth estimate (equation 6.49). Notice that most of the object surface is nicely reconstructed, except for few occluded portions of the scene (not observed from the camera or not illuminated by the light) leaving small white holes here and there (for example the right side of the nose).

Regarding the general setup, the camera was positioned at a height of $d_h = 22$ cm with respect to the desk, tilted down at an angle of approximately $\theta = 40$ degrees (see figure 6.9). Special care was taken in order to position the light source precisely

in the azimuth positions $\xi = 0$ degree and $\xi = 180$ degrees on the respective left and right hand scans. As shown in section 6.3, this configuration maximizes the average accuracies in shape estimation, by maximizing $|\cos\xi| = 1$. The elevation angle of the lighting direction was set to approximately $\phi = 70$ degrees for both scans. Notice that one could try smaller values in order to minimize $\tan\phi$ and thereby further improve global accuracies (see discussion in section 6.3). For completeness of the description, note that the lamp was positioned at an approximate height of $h_S = 62$ cm in both scans. Finally, notice that for that experiment, we decided to take the lamp reflector off, leaving the bulb naked. Consequently, we noticed a great improvement in the sharpness of the projected shadow (compared to the two first experiments). This was actually expected, since that way, the lamp became a better point light source. We believe that this operation was the main reason for the noticeable improvement in quality of reconstruction (compared with the second experiment): the surface noise was estimated to 0.09 mm in standard deviation throughout the entire surface. Over a depth variation of approximately 10 cm, this means a relative error of 0.1%. Once again, there was no significant global deformation in the final structured surface: we fit a quadratic model through the reconstructed set of points on the desk plane and noticed from planar to quadratic warping a decrease of only 2% on the standard deviation of surface noise.

Experiment 4 - Scanning of a textured skull: We took one scan of a small painted skull, using a single reference plane Π_h, with known light source position (pre-calibrated). Two images of the sequence are shown on the top row of figure 6.18. The recovered shape is presented on the second row (33,533 triangles), and the last row shows three views of the mesh textured by the top left image. Notice that the textured regions of the object are nicely reconstructed (although these regions have smaller contrast I_{contrast}). Small artifacts observable at some places on the top of the skull are due to the saturation of the pixel values to zero during shadow passage. This effect induces a positive bias on the threshold I_{shadow} (since I_{min} is not as small as it should be). Consequently, those pixels take on slightly too small shadow times t_s and are triangulated with shadow planes that are shifted to the left. In effect,

their final 3D location is slightly off the surface of the object. One possible solution to that problem consists of taking multiple scans of the object with different camera apertures, and retaining each time the range results for the pixels that do not suffer from saturation. The overall reconstruction error was estimated to approximately 0.1 mm over a 10 cm large object leading to a relative error of approximately 0.1%. In order to check for global distortion, we measured the distances between three characteristic points on the object: the tip of the two horns, and the top medium corner of the mouth. The values obtained from physical measurements on the object and the ones from the retrieved model agreed within the error of measurement (on the order of 0.5mm over distances of approximately 12 to 13cm). The sequence of images was 670 frames long, each image being 320×240 pixels large (acquired with a grayscale camera).

Experiment 5 - Textured and colored fruits: Figure 6.19 shows the reconstruction results on three textured and colored fruits. The second row shows the reconstructed shapes. The three meshes with the pixel images textured on them are shown on the third row. Similar reconstruction errors to the previous two experiments (Experiments 3 and 4) were estimated on that data set. Notice that both textured and colored regions of the objects were well reconstructed: the local surface errors was estimated between 0.1 mm and 0.2 mm, leading to relative errors of approximately 0.1%.

Experiment 6 - Outdoor scene: In this experiment, the sun was used as light source for scanning a small object. See figure 6.20. The final mesh is shown on the bottom figure (with $106,982$ triangles). The reconstruction error was estimated to 1mm in standard deviation, leading to a relative error of approximately 0.2%. The larger reconstruction error is possibly due to the fact that the sun is not well approximated by a point light source leading to shallower shadow edges (see discussion in Sec. 6.3). Once again, there was no noticeable global deformation induced by calibration. After fitting a quadratic model to sets of points on the planes, we only witnessed a decrease of approximately 5% on the standard deviation of the residual

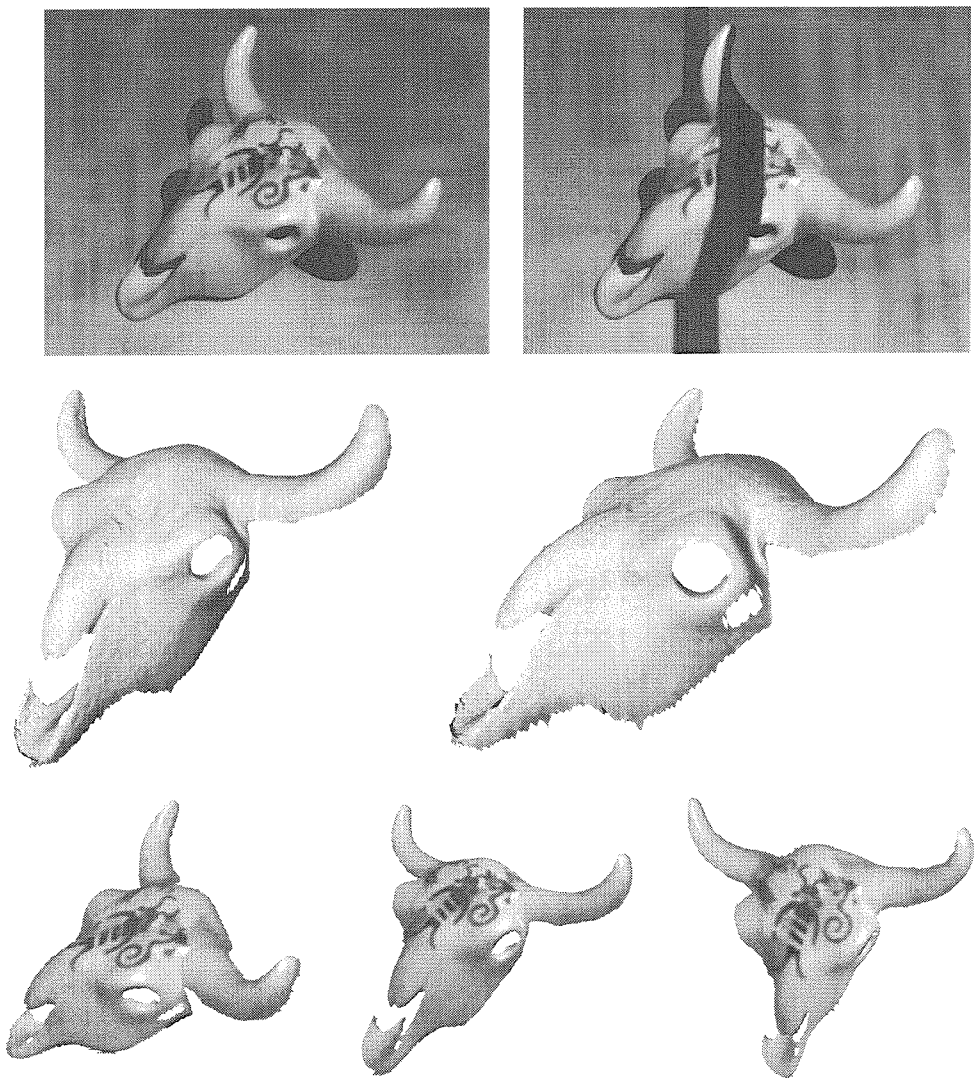


Figure 6.18: Experiment 4 - Scanning of a textured skull

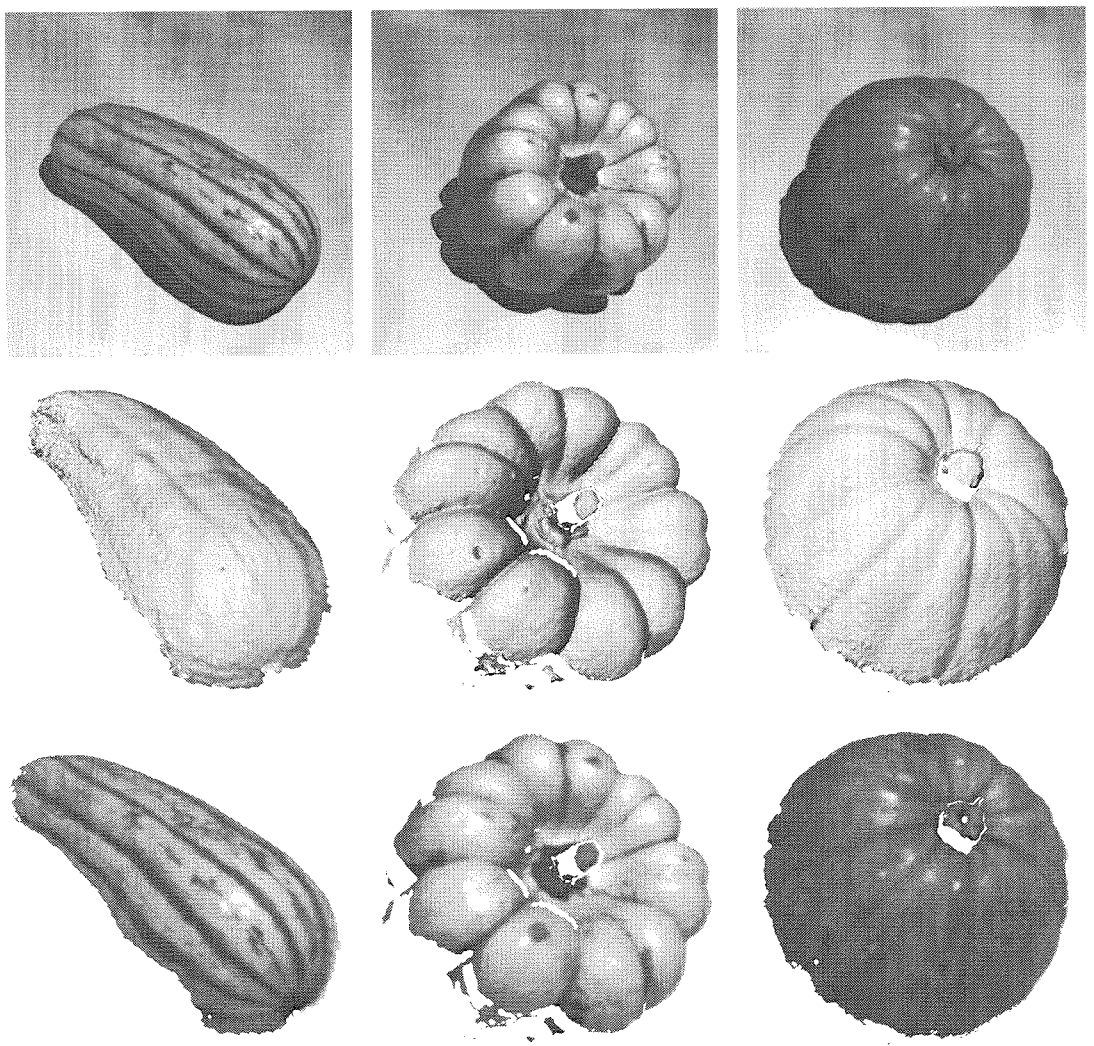


Figure 6.19: Experiment 5 - Textured and colored fruits

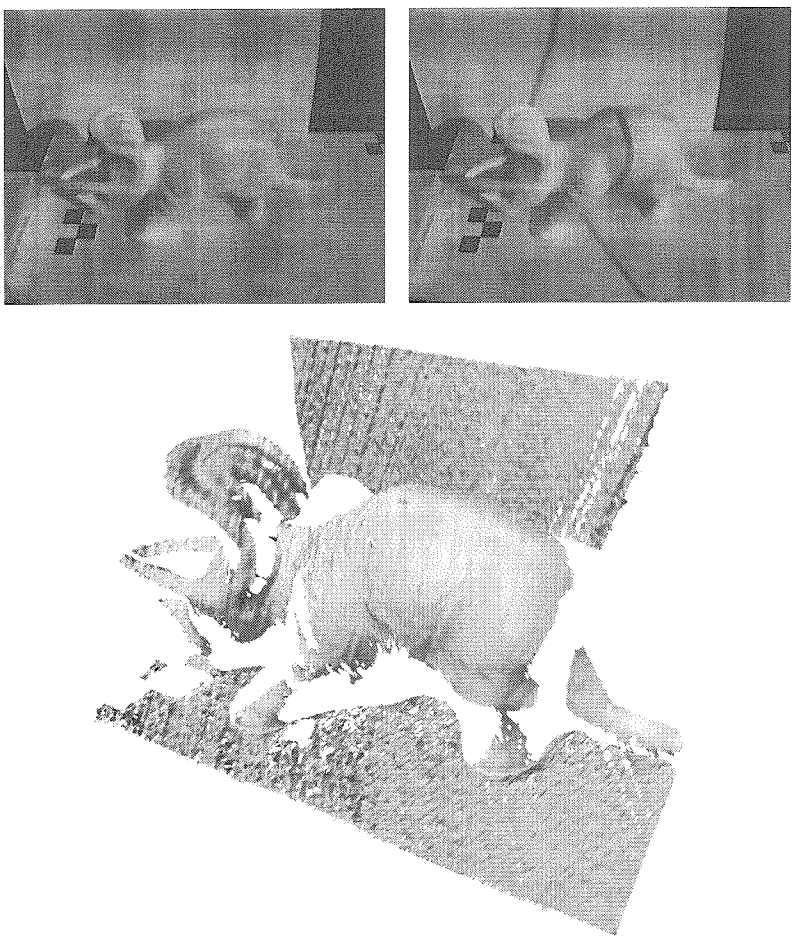


Figure 6.20: Experiment 6 - Outdoor scanning of an object

error. The original sequence was 790 images long acquired with a consumer electronics color camcorder (at 30 Hz). After digitization, and de-interlacing, each image was 640×240 pixel large. The different digitalization technique may also explain the larger reconstruction error.

Experiment 7 - Outdoor scanning of a car: Figure 6.21 shows the reconstruction results on scanning a car with the sun. The two planes (ground floor and back wall) approach was used to infer the shadow plane (without requiring the sun position). The initial sequence was 636 frames long acquired with a consumer electronics color video-camera (approximately 20 seconds long). Similarly to Experiment 4, the sequence was digitized resulting to 640×240 pixel large non-interlaced images. Two images of the sequence are presented on the top row, as well as two views of the reconstructed 3D

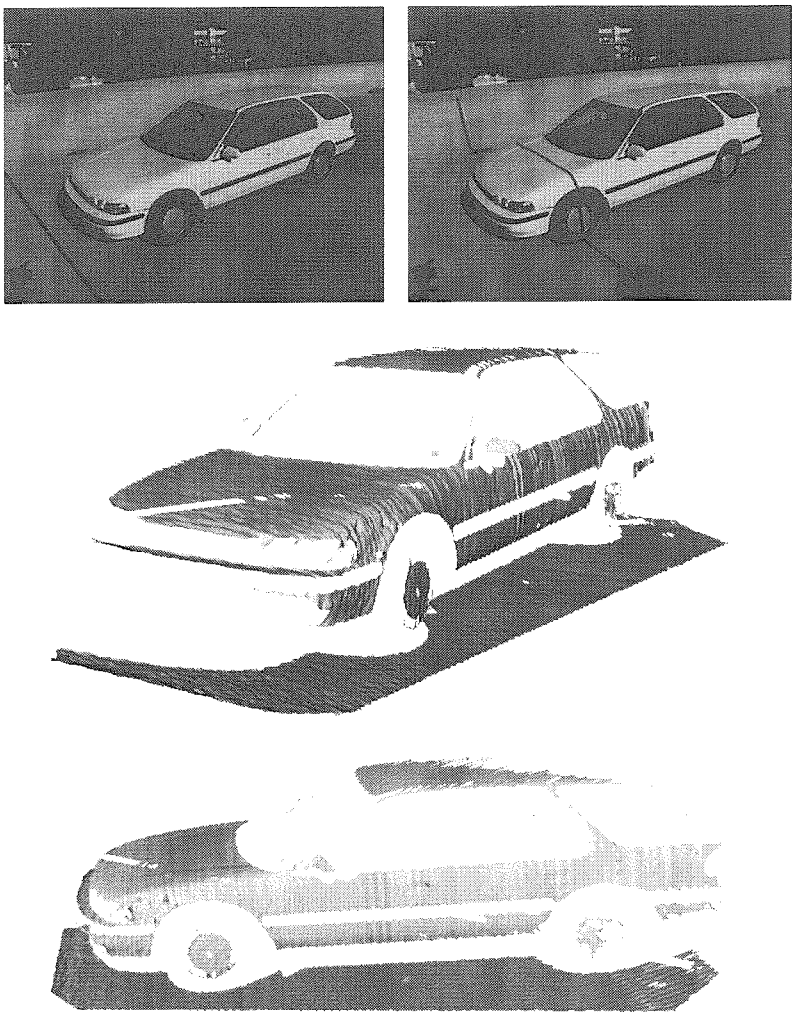


Figure 6.21: Experiment 5 - Outdoor scanning of a car

mesh after scanning. The reconstruction errors were estimated to approximately 1 cm, or 0.5% of the size of the car (approximately 3 meters).

All the reconstruction results presented in that section (and more) are available online in the form of VRML meshes at:

`http://www.vision.caltech.edu/bouguetj/ICCV98/gallery.html`

6.7 Conclusions

In this chapter, we have presented a simple, low cost system for 3D scanning. The system requires very little equipment (a light source, and a straight edge to cast

the shadow) and is very simple and intuitive to use and to calibrate. This technique scales well to large objects and may be used in brightly lit scenes where active lighting methods are impractical. In outdoor scenarios, the sun is used as light source and is allowed to move during a scan. The method requires very little processing and image storage and has been implemented in real time (30 Hz) on a Pentium 300MHz machine. The accuracies that we obtained on the final reconstructions are reasonable (error at most 0.5% of the size of the scene, or one part in 200). In addition, the final outcome is a dense and conveniently organized coverage of the surface (one point in space for each pixel in the image), allowing direct triangular meshing and texture mapping. We also showed that using dual-space geometry enables us to keep the mathematical formalism simple and compact throughout the successive steps of the method. An error analysis was presented together with a description of a simple technique for merging multiple 3D scans in order to obtain a better coverage of the scene, and reduce the estimation error. The overall calibration procedure, even in the case of multiple scans, is intuitive, simple, and accurate.

Our method may be used to construct complete 3D object models. One may take multiple scans of the object at different locations in space, and then align the sets of range images. For that purpose, a number of algorithms have been explored and shown to yield excellent results [71, 91, 92]. The final step consists of constructing the final object surface from the aligned views [93, 94, 92]. In order to apply these merging techniques, we believe it is necessary to further analyze (and control) all possible global deformation effects due to calibration errors. Indeed, even for standard structured lighting scanning systems, small errors on calibration may induce global distortions in the reconstructed scene that may forbid any alignment procedure to work at all.

It is also part of future work to incorporate a geometrical model of extended light source to the shadow edge detection process (although we do not believe that this will significantly improve the quality of the scans), in addition to developing an uncalibrated (projective) version of the method.

Observe that at least one reference (and calibrated) plane is necessary for 3D

shadow scanning. This plane is useful for directly computing the shadow plane location from the information contained in the image only. The following chapter explores an extension of this shadow scanning technique to cases where no reference plane is available in the scene.

Chapter 7 Geometry of planar shadows in B-dual-space

7.1 Motivation: Shadow scanning without any reference plane

In the previous chapter, we presented a method for capturing cheaply and quite accurately 3D surfaces based on projecting shadows onto the scene using a pencil (or another straight edge) and a conventional desk lamp [73, 77]. This approach has the advantage of being simple, and achieving full Euclidean reconstruction, however it requires the presence of a background plane used as a reference plane (assuming a calibrated light source). One question remains: Can we do without that reference surface? Or, what can we tell about the scene geometry from a set of projected shadows? In this paper, we demonstrate that, under the assumption that the light source position is known, planar shadows provide sufficient information for Euclidean 3D reconstruction (up to three global scalar parameters) and propose a simple algorithm for achieving such a reconstruction [95]. All the mathematical derivations will be using the dual-space formalism as fundamental tool.

We start with the description of the method in Sec. 7.2, followed in Sec. 7.3 by some experimental results. Preliminary results on the generalization to more than one light source are presented in the following section 7.4. We end with conclusions in Sec. 7.5.

7.2 Description of the method

Let us consider the scanning scenario as we presented in the previous chapter 6 and in [73]. Figure 7.1 recalls the complete geometry corresponding to this technique. On this figure, the plane Π_d is used as reference plane for scanning. Let us recall how 3D reconstruction is achieved in that geometrical setup: Assume that the positions of the light source S and the plane Π_d (reference plane) in the camera reference frame are known from calibration (see section 6.2.1). During scanning, the user casts a shadow on the scene observed as a curved edge $\mathcal{E}$ on the image. The goal is to estimate the 3D location of the point P in space corresponding to any point p on $\mathcal{E}$. Denote by Π the corresponding shadow plane. Assume that two portions of the shadow projected on the desk plane are visible on two given rows of the image (top and bottom rows on the figure). Consider the two points a and b lying at the intersection of $\mathcal{E}$ and the two reference rows. Their corresponding points A and B in the scene may be found by intersecting Π_d with the optical rays (O_c, a) and (O_c, b) respectively. The shadow plane Π is then inferred from the three points in space S, A and B (this technique is essentially described in section 6.2.5). Finally, the point P is retrieved by intersecting Π with the optical ray (O_c, p) (triangulation stage - see section 6.2.6).

The central observation is that for a given stick position, once the shadow plane Π is identified, so is the 3D position of the entire shadow edge $\mathcal{E}$ (by geometrical triangulation). The reference plane Π_d constitutes then a direct mean for locating the shadow plane in space (through the top and bottom reference rows) and achieve Euclidean reconstruction. This paper attempts to answer the question: Is there a way of recovering Π without Π_d?

We will first describe the scanning scenario. The light source, S, is at a known position with respect to the camera (from light source calibration - see section 6.2.3), and the camera is calibrated (see section 6.2.1 and chapter 3). During scanning, the user projects a succession of N shadows onto the scene (with a straight edge), generating N shadow edges $\mathcal{E}_i$ $(i = 1, \ldots, N)$ on the image plane. The problem of reconstructing scene geometry then leads to the problem of estimating the locations

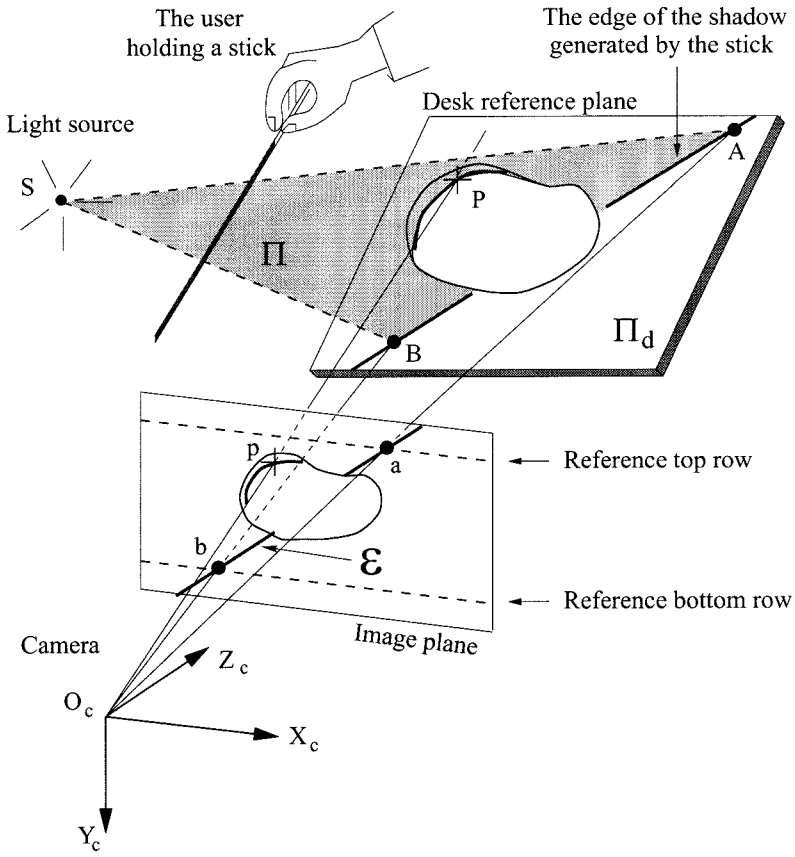


Figure 7.1: Scanning method using a reference plane Π_d.

of the associated N shadow planes Π_i in space. As one shadow edge is added to the list of edges, in principle three unknowns are added to the problem, corresponding to the three degrees of freedom of the associated shadow plane in space. Are there enough constraints imposed by the images that allow to estimate this total of $3N$ variables?

7.2.1 A constructive approach

Before getting into the mathematical details of the reconstruction algorithm, let us first build some intuition about the geometry of the problem.

For that purpose, let us state the two following properties.

Property 1 - Depth propagation along an edge: If the depths Z_A and Z_B of two distinct points a and b along a given shadow edge $\mathcal{E}$ are known, then so is the depth Z_P of any point p along that edge. In other words, depth information at two

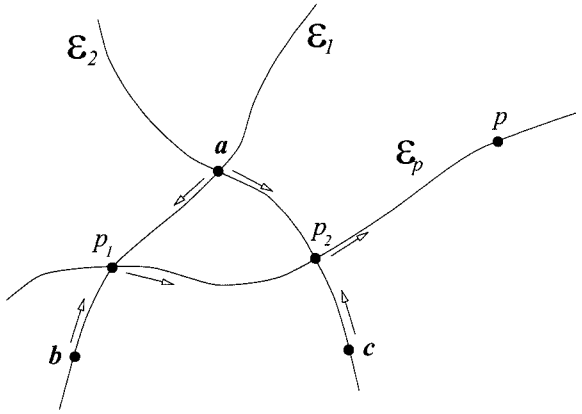


Figure 7.2: Depth propagation from edge to edge: Depth information propagates from the points a, b and c along the edges $\mathcal{E}_1$ and $\mathcal{E}_2$ to the points p_1 and p_2 and finally along $\mathcal{E}_p$ to p.

distinct points on an edge propagates along the entire edge.

Proof: Let $\overline{x}_A$ and $\overline{x}_B$ be the homogeneous coordinate vectors of the two points a and b on the image plane: $\overline{x}_A = [x_A \quad y_A \quad 1]^T$ and $\overline{x}_B = [x_B \quad y_B \quad 1]^T$. Then, if the two depths Z_A and Z_B are known, then so are the full coordinate vectors of the associated points A and B in the 3D scene: $\overline{X}_A = Z_A \overline{x}_A$ and $\overline{X}_B = Z_B \overline{x}_B$. Therefore, the associated shadow plane Π is the unique plane passing through the three points A, B and S. Once Π is recovered, any point p along the edge $\mathcal{E}$ may be triangulated leading to Z_P. ■

This approach was implicitly used in the scanning system presented in chapter 6 in connection with the reference plane and illustrated in figure 7.1. Consider Fig. 7.1, where the depths of the two points a and b are known due to the fact that they lie on the known reference plane Π_d. Consequently, following Property 1, depth information at a and b propagate to every point p along the edge $\mathcal{E}$.

Property 2 - Depth propagation from edge to edge (to the entire image): Let a, b and c be three distinct points on the image (in the scannable area). Assume their respective depths Z_A, Z_B and Z_C known. Then, by "appropriate" shadow scanning, one may retrieve the depth at any point p on the image (in scannable areas).

Proof: The proof of this property is constructive. First project a shadow edge $\mathcal{E}_1$

that goes through the points a and b, and another one ($\mathcal{E}_2$) through a and c (see figure 7.2). Given that two points (a and b) on $\mathcal{E}_1$ are of known depths, according to Property 1, one may compute the depth of every point along that edge. The same holds for $\mathcal{E}_2$. For any point p in the image, project an edge $\mathcal{E}_p$ that passes through p and intersect $\mathcal{E}_1$ and $\mathcal{E}_2$ at any two distinct points p_1 and p_2 (different from a). Since p_1 and p_2 lie on the two known edges $\mathcal{E}_1$ and $\mathcal{E}_2$, their depths are known. Therefore, following the depth propagation principle (Property 1), the depth of every point along $\mathcal{E}_p$ may be computed, in particular that of the point p (see fig. 7.2). ■

A direct consequence of that property is that the knowledge of the depth at three distinct points in the image is enough to recover the entire scene depth map. Therefore, fixing three scalar parameters (the three depths) is sufficient to retrieve a complete Euclidean reconstruction of the scene (of course restricted to the scannable areas). Notice that we have not yet shown that this is a necessary condition. In other words, there could exist an alternative scanning strategy that requires only 2 or less scalar identifications in order to achieve the same Euclidean reconstruction. Below we show that the condition is also necessary.

Notice that the basic constraint that we used in order to propagate depth information from $\{a, b, c\}$ to p made direct use of the intersecting points p_1 and p_2 of the shadow edges. As the scanning progresses, more edges are projected on the scene, generating more and more intersections. In fact, while approaching the end of the scanning procedure, it is very likely to find a lot more than two intersecting points per edge. Therefore, in presence of noise in the measurements, this direct constructive method may not be the most robust technique to propagate depth information across the image through the edge-web (defined as the entire set of edges). A better algorithm exists in order to make appropriate use of all the edge intersections at once. It will be presented in section 7.2.2.

An edge $\mathcal{E}$ is an *isolated edge* if and only if it does not intersect with at least two other edges on the image. Notice that depth information cannot possibly be propagated to any isolated edge from the rest of the edge-web. In other words, any attempt to compute depth information, or shadow plane coordinates for any

isolated edge is hopeless. Therefore, every isolated edge should be rejected prior to any computation.

7.2.2 3D reconstruction algorithm in dual-space

In this section, we derive the complete algorithm for 3D reconstruction from planar shadows. One may find a summary of it at the end of the section.

Let Π_i be the i^{th} shadow plane generated by the stick ($i = 1, \ldots, N$), with corresponding plane vector $\overline{\omega}_i = [\omega_x^i \;\; \omega_y^i \;\; \omega_z^i]^T$ (in dual space). For all vectors $\overline{\omega}_i$ to be well defined, it is required that none of the planes Π_i contain the camera center O_c. See section 2.2.1. Denote by $\mathcal{E}_i$ the associated shadow edge observed on the image plane. The N vectors $\overline{\omega}_i$ constitute then the main unknowns in the reconstruction problem. Indeed, once those vectors are identified, all edges can be triangulated in space. Therefore, there is apparently a total of $3N$ unknown variables. However, given the scanning scenario, every shadow plane Π_i must contain the light source point S. Therefore, denoting $\overline{X}_S = [X_S \;\; Y_S \;\; Z_S]^T$ the light source coordinate vector in the camera reference frame (known), we have (see sec. 2.2.1):

$$\forall i = 1, \ldots, N, \qquad \langle \overline{\omega}_i, \overline{X}_S \rangle = 1 \tag{7.1}$$

Equivalently, in dual-space, all shadow plane vectors $\overline{\omega}_i$ must lie on the plane $\hat{S}$, dual-image on the light source point S. One may then explicitly use that constraint, and parameterize the vectors $\overline{\omega}_i$ using a two-coordinate vector $\overline{u}_i = [u_x^i \;\; u_y^i]^T$ such that:

$$\overline{\omega}_i = \mathbf{W}\,\overline{u}_i + \overline{\omega}_o = \begin{bmatrix} \overline{\omega}_{s1} & \overline{\omega}_{s2} \end{bmatrix} \overline{u}_i + \overline{\omega}_o \tag{7.2}$$

where $\overline{\omega}_o$, $\overline{\omega}_{s1}$, and $\overline{\omega}_{s2}$ are three vectors defining the parameterization. For example, if $X_S \neq 0$, one may then keep the last two coordinates of $\overline{\omega}_i$ as parameterization: $\overline{u}_i = [\omega_y^i \;\; \omega_z^i]^T$, picking $\overline{\omega}_{s1} = [-Y_S/X_S \;\; 1 \;\; 0]^T$, $\overline{\omega}_{s2} = [-Z_S/X_S \;\; 0 \;\; 1]^T$ and $\overline{\omega}_o = [1/X_S \;\; 0 \;\; 0]^T$. Any other choice of linear parameterization is acceptable (there will always exist one given that $S \neq O_c$). In order to define a valid coordinate change,

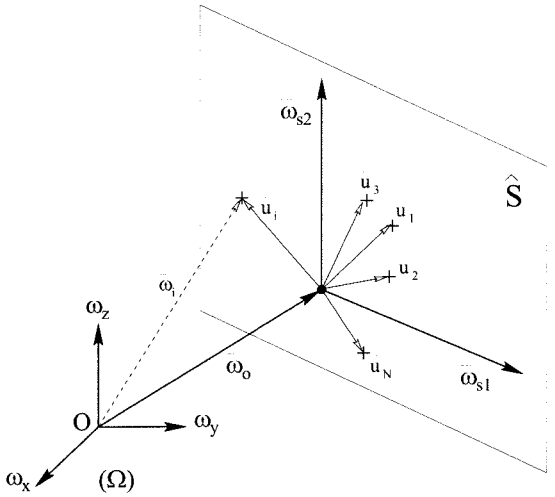


Figure 7.3: Light source constraint: Every shadow plane Π_i contains the light source point S. Therefore, in dual space (Ω), all the shadow plane vectors $\overline{\omega}_i$ ($i = 1, \ldots, N$) must lie on the plane $\hat{S}$, the dual image of the light source S. The reduced parameterization $\overline{u}_i$ makes explicit use of that constraint. It is defined by the three vectors $\overline{\omega}_{s1}$, $\overline{\omega}_{s2}$ and $\overline{\omega}_o$.

the three non-zero vectors $\overline{\omega}_o$, $\overline{\omega}_{s1}$, and $\overline{\omega}_{s2}$ must only satisfy the three conditions (a) $\langle \overline{\omega}_o, \overline{X}_S \rangle = 1$, (b) $\langle \overline{\omega}_{s1}, \overline{X}_S \rangle = \langle \overline{\omega}_{s2}, \overline{X}_S \rangle = 0$, (c) $\overline{\omega}_{s1} \neq \overline{\omega}_{s2}$. In dual-space, $\{\overline{\omega}_{s1}, \overline{\omega}_{s2}\}$ (or $\mathbf{W}$) may be interpreted as a basis vector of the plane $\hat{S}$ and $\overline{\omega}_o$ as one particular point on that plane (see figure 7.3).

After that parameter reduction, the total number of unknown variables clearly reduces to $2N$: two coordinates u_x^i and u_y^i per shadow plane Π_i. Given that reduced plane vector parameterization (called $\overline{u}$-parameterization), let us derive the analytical basis of the global reconstruction algorithm.

As it is described in the previous section, the only elements that lets depth information propagate from edge to edge are the intersecting points between the edges themselves. These points actually provide the only geometrical constraints that may be extracted from the images. Therefore, the first step consists of studying the type of constraint provided by an elementary edge intersection.

Assume that the two edges $\mathcal{E}_n$ and $\mathcal{E}_m$ intersect at the point p_k on the image ($n \neq m$), and let Π_n and Π_m be the two associated shadow planes with coordinate vectors $\overline{\omega}_n$ and $\overline{\omega}_m$. See figure 7.4. Let $\overline{x}_k$ be the homogeneous coordinate vector of

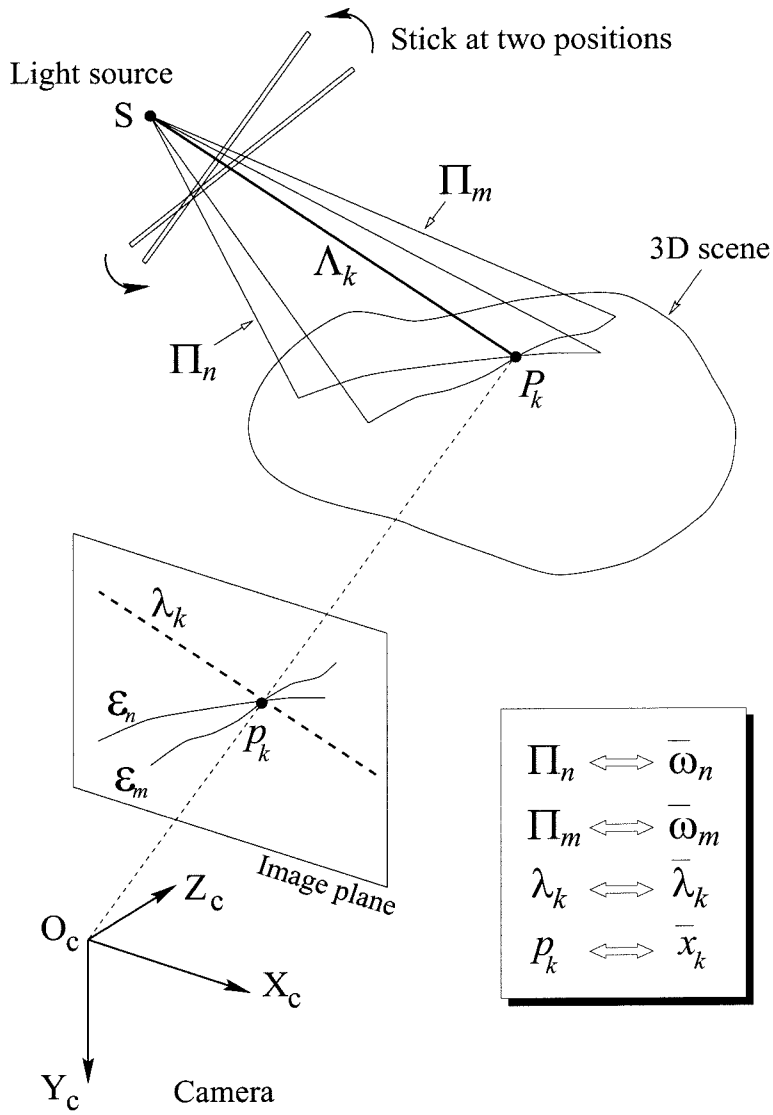


Figure 7.4: Elementary edge intersection: The point p_k lies at the intersection of the two edges $\mathcal{E}_n$ and $\mathcal{E}_m$ on the image plane. What does p_k tell us about the corresponding shadow planes Π_n and Π_m?

p_k on the image plane, and Z_k the depth of the corresponding point P_k in the scene. Then, the two edges $\mathcal{E}_n$ and $\mathcal{E}_m$ intersect in space at P_k if and only if the planes Π_n and Π_m and the scene surface intersect at a unique point in space (P_k). Equivalently, the depth Z_k may be computed by triangulation using either plane Π_n or Π_m. This condition translates into the two constraint equations $Z_k = 1/\langle \overline{\omega}_n, \overline{x}_k \rangle = 1/\langle \overline{\omega}_m, \overline{x}_k \rangle$ (standard triangulation equation in dual-space - see section 2.2.3, eq. 2.33). A very natural way of eliminating the depth variable (Z_k) is to re-write the constraint as follows:

$$\langle \overline{x}_k, \overline{\omega}_n - \overline{\omega}_m \rangle = 0 \tag{7.3}$$

This unique equation captures then all the information that is contained into an elementary edge intersection. There is a very intuitive geometrical interpretation of that equation: Let Λ_k be the line of intersection between the two planes Π_n and Π_m in space, and let λ_k be the perspective projection of that line onto the image plane. Then, the vector $\overline{\lambda}_k = \overline{\omega}_n - \overline{\omega}_m$ is one coordinate vector of the line λ_k (see Proposition 1 in sec. 2.2.2). Therefore, equation 7.3 is merely $\langle \overline{x}_k, \overline{\lambda}_k \rangle = 0$, which is equivalent to enforcing the point p_k to lie on λ_k (see figure 7.4). Equation (7.3) has both advantages of (a) not explicitly involving the depth Z_k (which may be computed afterwards from the shadow plane vectors) and (b) being linear in the plane vectors unknowns $\overline{\omega}_n$ and $\overline{\omega}_m$. The same constraint may also be written as a function of $\overline{u}_n$ and $\overline{u}_m$, the two $\overline{u}$-parameterization vectors of the shadow planes Π_n and Π_m:

$$\langle \overline{y}_k, \overline{u}_n - \overline{u}_m \rangle = 0 \tag{7.4}$$

where $\overline{y}_k = \mathbf{W}^T \overline{x}_k$ (a 2-vector). Notice that this new equation remains linear and homogeneous in that reduced parameter space.

Let N_p be the total number of intersection points p_k ($k = 1, \ldots, N_p$) existing in the *edge-web* (the entire set of edges). Assume that a generic p_k lies at the intersection of the two edges $\mathcal{E}_{n(k)}$ and $\mathcal{E}_{m(k)}$ ($n(k)$ and $m(k)$ are the two different edge indices).

The total set of constraints associated to the N_p intersections may then be collected in the form of N_p linear equations:

$$\forall k = 1, \ldots, N_p, \quad \langle \overline{y}_k, \overline{u}_{n(k)} - \overline{u}_{m(k)} \rangle = 0 \tag{7.5}$$

which may also be written in a matrix form:

$$\mathbf{A}\,\overline{U} = \mathbf{0}_{N_p} \tag{7.6}$$

where $\mathbf{0}_{N_p}$ is a vector of N_p zeros, $\mathbf{A}$ is an $N_p \times 2N$ matrix (function of the $\overline{y}_k$ coordinate vectors only) and $\overline{U}$ is the vector of reduced plane coordinates (of length $2N$): $\overline{U} = [\overline{u}_1^T \quad \overline{u}_2^T \quad \cdots]^T = [u_x^1 \quad u_y^1 \quad u_x^2 \quad u_y^2 \quad \cdots]^T$. The vector $\overline{U}$ will sometimes be denoted $\overline{U} = \{\overline{u}_i\}_{i=1\ldots N}$. According to eq. 7.6, the solution for the shadow plane vectors lies in the null space of the matrix $\mathbf{A}$. It is therefore essential to identify the rank of that matrix or equivalently the dimension of its null space. As a general remark, notice that the basic hope of solving that system comes from the fact that the number of points of intersection grows faster than the number of edges in the image. Essentially, the condition $N_p \geq 2N$ is not too demanding.

Definition 2 - Fully connected edge-web: The edge-web is *fully connected* if and only if it cannot be partitioned into two groups of edges which have less that two (zero or one) points in common. In particular a fully connected edge-web does not contain any isolated edge. Notice that under that condition only, depth information can freely propagate through the entire web following the constructive approach described in section 7.2.1. A *normal scanning scenario* is then defined as a scenario where the edge-web is fully connected and the total number of intersections is larger than $2N$ (this last condition will be relaxed later on).

Theorem 1: In a normal scanning scenario, the rank of the matrix $\mathbf{A}$ is exactly $2N - 3$ (or alternatively, the null space of $\mathbf{A}$ is of dimension 3).

Proof: We presented in section 7.2.1 a constructive method that allows to compute the entire geometry of the scene from the knowledge of the depth at three distinct

points (from propagation of depth information from edge to edge). Therefore, in the case of a normal scanning scenario, the reconstruction problem has at most three free parameters. Consequently the dimension of the null space of $\mathbf{A}$ is at most 3, or equivalently, $\mathbf{A}$ is of rank at least $2N - 3$.

Consider now the dimension 2 linear subspace $\mathcal{S}$ of vectors $\overline{U}$ that have the following form: $\overline{U} = \overline{U}_{\alpha,\beta} = [\alpha \quad \beta \quad \alpha \quad \beta \quad \cdots]^T$ $((\alpha, \beta) \in \mathbb{R}^2)$. It is straightforward to show that for any value of α and β, the vector $\overline{U}_{\alpha,\beta}$ lies in the null space of $\mathbf{A}$: $\mathbf{A}\,\overline{U}_{\alpha,\beta} = \mathbf{0}$. Therefore, $\mathcal{S}$ is included in the null space of $\mathbf{A}$ which is therefore of dimension at least 2. Equivalently, the rank of $\mathbf{A}$ is less or equal than $2N - 2$.

Finally, the null space of $\mathbf{A}$ cannot be restricted to $\mathcal{S}$, otherwise, the only solutions to the problem would reduce to sets of identical shadow planes in space ($\overline{u}_i = [\alpha \quad \beta]^T$, $\forall i = 1, \ldots, N$), which is impossible in practice. Therefore, the dimension of the null-space of $\mathbf{A}$ must be at least three (in order to allow for distinct shadow planes) leading to the rank of the matrix being at most $2N - 3$. Therefore the rank of $\mathbf{A}$ is exactly $2N - 3$. ■

A direct consequence of that theorem is that no matter which strategy one adopts in solving for the set of constraints, there will always be three free parameters to set in order to achieve Euclidean reconstruction. In addition, since the linear system is rank $2N-3$, there needs only a minimum of $2N-3$ intersection points in the edge-web (and not $2N$). It is straightforward to show that this condition is always automatically satisfied if the edge-web is fully connected. Therefore, a normal scanning scenario may be re-defined as a scenario in which the edge-web is fully connected. Notice that figure 7.2 shows the minimum configuration for scanning: $N = 3$ and $N_p = 2N - 3 = 3$. The following corollary is another important consequence of Theorem 1.

Corollary 1: Let $\overline{U}^o = \{\overline{u}_i^o\}$, $(i = 1, \ldots, N)$ be a non-trivial solution for the linear system 7.6 (by non-trivial, we mean a solution vector such that at least two vectors $\overline{u}_i$ and $\overline{u}_j$ are distinct for some $i \neq j$). Then, for every solution vector $\overline{U} = \{\overline{u}_i\}$ to equation 7.6, there exists three scalars α, β and γ such that:

$$\forall i = 1, \ldots, N, \qquad \overline{u}_i = \gamma\, \overline{u}_i^o + \overline{u}_o \tag{7.7}$$

with $\overline{u}_o = [\alpha \quad \beta]^T$. Conversely, for any scalars α, β and γ, the vector $\overline{U} = \{\overline{u}_i\}$ given by equation 7.7 is solution of the linear system (7.6). The vector $\overline{U}^o$ is called a 'seed' solution from which all solutions of the linear system may be identified. Notice that not all these solutions lead to the exact Euclidean reconstruction of the scene, only one of them does (corresponding to one particular set of coefficients α, β and γ). However, one seed vector enables to explicitly identify the whole set of possible solutions, each one of them having a valid 3D scene interpretation agreeing with the set of observed shadows. The following part of the paper will focus on that seed vector identification.

First, since any non-trivial solution vector $\overline{U}^o = \{\overline{u}_i^o\}$ may be used as seed (as long as it is one solution of (7.6)), let us pick the one (called *unitary seed vector*) that satisfies the two extra normalizing conditions: (a) $\sum_{i=1}^{N} \overline{u}_i^o = 0$ (zero mean), (b) $\sum_{i=1}^{N} \|\overline{u}_i^o\|^2 = 1$ (unit norm). Those conditions assure a non trivial solution (all $\overline{u}_i$ cannot be identical). The unitary seed vector $\overline{U}^o$ satisfies the linear equation $\mathbf{B}\,\overline{U}^o = \mathbf{0}_{N_p+2}$, where $\mathbf{B}$ is the following augmented $(N_p + 2) \times 2N$ matrix:

$$\mathbf{B} = \begin{bmatrix} & & & & \mathbf{A} & & \\ 1 & 0 & 1 & 0 & \cdots & 1 & 0 \\ 0 & 1 & 0 & 1 & \cdots & 0 & 1 \end{bmatrix} \tag{7.8}$$

The two last rows of $\mathbf{B}$ enforce the zero-mean constraint, bringing the rank of $\mathbf{B}$ to $2N - 1$ $(= (2N - 3) + 2)$. Therefore, the dimension of null space of $\mathbf{B}$ is one, leading to $\overline{U}^o$ being the unitary eigenvector associated to the unique zero eigenvalue of $\mathbf{B}$. Consequently, a standard a singular value decomposition (SVD) of $\mathbf{B}$ allows to naturally retrieve $\overline{U}^o$. Such a decomposition leads to the following relation:

$$\mathbf{B} = \mathbf{U}\,\mathbf{S}\,\mathbf{V}^T \tag{7.9}$$

where $\mathbf{U}$ and $\mathbf{V} = [\mathbf{V}_1 \quad \mathbf{V}_2 \quad \cdots \quad \mathbf{V}_{2N}]$ are two unitary matrices of respective sizes $(N_p + 2) \times (N_p + 2)$ and $2N \times 2N$, and $\mathbf{S}$ is the $(N_p + 2) \times 2N$ matrix of singular values. The unitary seed vector $\overline{U}^o$ is then the column vector of $\mathbf{V}$ associated to the

zero singular value. Without loss of generality, assume it is the last column vector: $\overline{U}^o = \{\overline{u}_i^o\} = \mathbf{V}_{2N}$. Alternatively, one may retrieve the same $\mathbf{V}$ matrix by applying the same decomposition on the smaller $2N \times 2N$ symmetric matrix $\mathbf{C} = \mathbf{B}^T\mathbf{B}$. Such a matrix substitution is advantageous because the so-defined matrix, $\mathbf{C}$, has a simple block structure:

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_{1,1} & \mathbf{C}_{1,2} & \cdots & \mathbf{C}_{1,N} \\ \mathbf{C}_{2,1} & \mathbf{C}_{2,2} & \cdots & \mathbf{C}_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_{N,1} & \mathbf{C}_{N,2} & \cdots & \mathbf{C}_{N,N} \end{bmatrix} \tag{7.10}$$

where each matrix element $\mathbf{C}_{i,j}$ is of size 2×2. Let us derive a closed-form expression for $\mathbf{C}_{i,j}$. Since two shadow edges can only intersect once (two shadow planes intersect along a line that can only intersect the scene at a single point), then for a given pair of shadow indices (i, j) there exists at most one index k such that $p_k = \mathcal{E}_i \cap \mathcal{E}_j$. This mapping will be denoted $k = k(i, j)$ (notice for example that $k(i, j) = k(j, i)$ and $k(n(k'), m(k')) = k'$). If the two edges do not intersect (such as an edge with itself), or if that intersection is not visible on the image plane, then the point $p_{k(i,j)}$ does not exist. However, we will still refer to it as a phantom point of coordinate vector $\overline{x}_{k(i,j)} = [0 \quad 0 \quad 0]^T$ (leading to $\overline{y}_{k(i,j)} = \mathbf{W}^T\overline{x}_{k(i,j)} = [0 \quad 0]^T$). Adopting that formalism, all the following algebraic equations remain valid even in the case of missing intersections. Given this notation, it may be shown that all matrices $\mathbf{C}_{i,j}$ have the following expressions:

$$\mathbf{C}_{i,j} = I_2 - \overline{y}_{k(i,j)}\,\overline{y}_{k(i,j)}^T \qquad \text{if } i \neq j \tag{7.11}$$

$$\mathbf{C}_{i,i} = I_2 - \sum_{n=1}^{N} \overline{y}_{k(i,n)}\,\overline{y}_{k(i,n)}^T \tag{7.12}$$

where I_2 is the 2×2 identity matrix. Observe that, every off-diagonal matrix element $\mathbf{C}_{i,j}$ $(i \neq j)$ depends only on the intersection point $p_{k(i,j)}$ between edges $\mathcal{E}_i$ and $\mathcal{E}_j$ (equation 7.11). Every diagonal block $\mathbf{C}_{i,i}$ however is function of all the intersection

points of $\mathcal{E}_i$ with the rest of the edge-web (the sum is over all points $p_{k(i,n)}$, for $n = 1, \ldots, N$).

Once the $\mathbf{C}$ matrix is built, $\mathbf{V}$ is retrieved by singular value decomposition. This technique allows then for a direct identification of the unitary seed solution $\overline{U}^o = \{\overline{u}_i^o\}$ leading the set of all possible solutions of (7.6) (following equation 7.7 in Corollary 1). Euclidean reconstruction is thus achieved up to the three parameters α, β and γ.

From that analysis, one may question the optimality of such an estimation scheme in presence of noise in the measurements (intersection coordinates $\overline{x}_k$). Essentially, in a noisy situation, the matrix $\mathbf{B}$ becomes full rank (the smallest singular value is not exactly zero), and then SVD really finds the (non-trivial) unit length vector $\overline{U}^o$ that minimizes the square norm of $\mathbf{A}\,\overline{U}^o$. The question is really about the geometrical meaning of such a cost minimization. Let $\overline{U} = \{\overline{u}_i\}$ be the real set of shadow plane coordinates corresponding to Euclidean reconstruction. Then, according to Corollary 1, there exists a unique set of coefficients α, β and γ such that: $\overline{u}_i = \gamma\,\overline{u}_i^o + \overline{u}_o$ with $\overline{u}_o = [\alpha \quad \beta]^T$, for all $i = 1, \ldots, N$. Those three global scalar parameters upgrade the reconstruction to Euclidean. Then, the full plane coordinate vectors $\overline{\omega}_i$ have the expression $\overline{\omega}_i = \mathbf{W}\,\overline{u}_i + \overline{\omega}_o$ (from equation 7.2). The depth Z_k of a given intersection point p_k may then be computed from either planes $\Pi_{n(k)}$ or $\Pi_{m(k)}$ leading, in an ideal noiseless scenario, to the same estimates. In presence of noise, however, those two quantities may be different. Let us denote them $Z_k^{(1)}$ and $Z_k^{(2)}$:

$$Z_k^{(1)} = \frac{1}{\gamma\,\langle \overline{u}_{n(k)}^o, \overline{y}_k \rangle + \langle \overline{u}_o, \overline{y}_k \rangle + \langle \overline{\omega}_o, \overline{x}_k \rangle}$$

$$Z_k^{(2)} = \frac{1}{\gamma\,\langle \overline{u}_{m(k)}^o, \overline{y}_k \rangle + \langle \overline{u}_o, \overline{y}_k \rangle + \langle \overline{\omega}_o, \overline{x}_k \rangle}$$

Then, the inverse depth difference is a direct function of the seed vector coordinates: $\epsilon_k = 1/Z_k^{(1)} - 1/Z_k^{(2)} = \gamma\,\langle \overline{y}_k, \overline{u}_{n(k)}^o - \overline{u}_{m(k)}^o \rangle$. This may also be written in a matrix form: $\overline{\epsilon} = \gamma\,\mathbf{A}\,\overline{U}^o$, where $\overline{\epsilon} = [\epsilon_1 \quad \epsilon_2 \quad \ldots \quad \epsilon_{N_p}]^T$. Therefore, independently from the global parameters α, β and γ, the least squares solution given by SVD minimizes the

2-norm of $\overline{\epsilon}$, or equivalently:

$$\overline{U}^o = \underset{\|\overline{U}\|=1}{\operatorname{argmin}} \sum_{k=1}^{N_p} \left(1/Z_k^{(1)} - 1/Z_k^{(2)}\right)^2$$

Consequently, this estimation scheme is optimal in the sense of minimizing the sum, over all intersection points, of the square of the inverse depth errors. This cost function may not exactly achieve the goal of reconstructing the "best" possible three-dimensional model of the scene. For example, a more natural cost function would directly involve the depth differences $Z_k^{(2)} - Z_k^{(1)}$ rather than the inverse depth differences. However, the main difficulty about using such a cost function comes from the fact that the quantity $Z_k^{(2)} - Z_k^{(1)}$ remains a function of the three scalars α, β and γ that are not known until three additional geometrical constraints are enforced, such as the depth at three points.

Once the seed solution $\overline{U}^o = \{\overline{u}_i^o\}$ is found (by SVD), one may identify the final "Euclidean" solution $\overline{U} = \{\overline{u}_i\}$ if the depth of (at least) three points in the scene are known. Without loss of generality, assume that these points are p_k for $k = 1, 2, 3$ (with depths Z_k). Those points provide then three linear equations in the unknown coefficient vector $\overline{\alpha} = [\alpha \ \ \beta \ \ \gamma]^T$:

$$\left[\begin{array}{cc} \overline{y}_k^T & \langle \overline{u}_{n(k)}^o, \overline{y}_k \rangle \end{array} \right] \overline{\alpha} = 1/Z_k - \langle \overline{\omega}_o, \overline{x}_k \rangle \tag{7.13}$$

for $k = 1, 2, 3$ resulting into a linear system of three equations and three unknowns. This system may then be solved, yielding the three coefficients α, β and γ, and therefore the final solution vector $\overline{U}$ (through eq. 7.7). Complete Euclidean shape reconstruction is thus achieved. If more points are used as initial depths, the system may be solved in the least squares sense (once again optimal in the inverse depth error sense). Notice that the reference depth points do not have to be intersection points as eq. 7.13 seem to infer. Any three (or more) points in the edge-web may be used.

Finally, the proposed method for 3D reconstruction may be summarized into five

successive steps:

Step 1:	Acquire a set shadow images, extract the shadow edges and compute their intersections.
Step 2:	Reject all isolated edges (and isolated groups of edges) so that the entire edge-web is fully connected (def. 2). Results a set of N edges $\mathcal{E}_i$, and N_p intersection points p_k.
Step 3:	Build the $2N \times 2N$ matrix $\mathbf{C}$ (eq. 7.10, 7.11 and 7.12), and compute the unitary seed vector $\overline{U}^o$ by SVD. Euclidean reconstruction is then achieved up to the three scalars α, β and γ.
Step 4:	Select (at least) three points in the scene with known depths, and solve linearly for the remaining scalars α, β and γ (eq. 7.13).
Step 5:	Compute the list of shadow plane vectors $\overline{\omega}_i$ (eq. 7.7 and 7.2) and triangulate all the points in the edge-web. The resulting set of 3D points may then be triangulated into a surface mesh for visualization purposes (fig. 7.5 and 7.6).

7.2.3 Final discussion

As demonstrated in the previous section, it is necessary to compute the three global parameters α, β and γ in order to achieve Euclidean reconstruction. Shadow edges alone can only provide reconstruction up to three scalar coefficients. The proposed technique requires the knowledge of the depth at at least three points in the scene for computing these coefficients. However, this condition may not always apply. Indeed, other clues may be known about the scene, such as planarity of portions of the scene, angles between different planes, or mutual distances between points in space. Those clues do not constitute direct depth measurements, but may however be used as constraints to upgrade the reconstruction to Euclidean. In those cases, it is clearly useful to be able to keep track of all possible solutions (up to step 3), and then identify among those the one(s) that satisfy the additional constraints (in attempt to find α, β and γ). If these extra geometrical constraints are sufficient, only one solution will be isolated, leading to a unique possible Euclidean interpretation of the whole scene (a new stage 4). The structure of this new reconstruction method allows for that type of modifications. This work is part of future investigations.

Regarding that extension, it is interesting to notice that for example a planarity constraint alone is not sufficient to recover α, β and γ. Indeed, in that case $\gamma = 0$ is

an obvious numerical solution to the problem (for any β and γ values), collapsing all shadow planes into a unique plane in space (which is of course physically impossible). In that case, the entire scene is wrongly reconstructed flat on that single shadow plane. That observation leads us to believe that studying the combination of multiple geometrical constraints for Euclidean upgrade is a good initial path for investigations.

7.3 Experimental results

Figures 7.5 and 7.6 show experimental results obtained from two real scenes. The first one consists of two parallel planes 5cm away from each other, and the second one a small object (a moon) on a plane. In both experiments, first the seed solution vector $\overline{U}^o$ was computed by SVD (following steps 1 to 3 of the method) and then α, β and γ were recovered using the three known depths at the three circled points on the figures. For that purpose, the background plane was calibrated in both cases, but only to recover the depth at the three reference points (the entire background planes were not used for scanning).

There are different ways to assess reconstruction accuracies. The first one consists of looking at the depth errors $Z_k^{(1)} - Z_k^{(2)}$ at the intersection points p_k relative to their absolute depths in the camera reference frame. In both experiments, this error is approximately 3mm (in standard deviation) over an average scene depth of 25cm. This leads to a relative depth error of 1.2%. However, in modeling applications, a more relevant quantity to look at is the reconstructed surface roughness relative to the size of the object of interest. In that case, this error is approximately 3mm over object sizes of 5 to 10cm, or a part in 20. In addition to surface roughness, it is essential to check for possible global distortions in the final reconstructed scene. For that purpose, one may quantify how well a number of intrinsic geometrical properties of the scene are preserved after reconstruction. Planarity of planes, angles between planes, or size of objects are typical samples of such properties. For example, in both experiments, the reconstructed planes deviate from planarity by approximately 4mm (in maximum). This concerns both planes of scene 1 and the background plane of

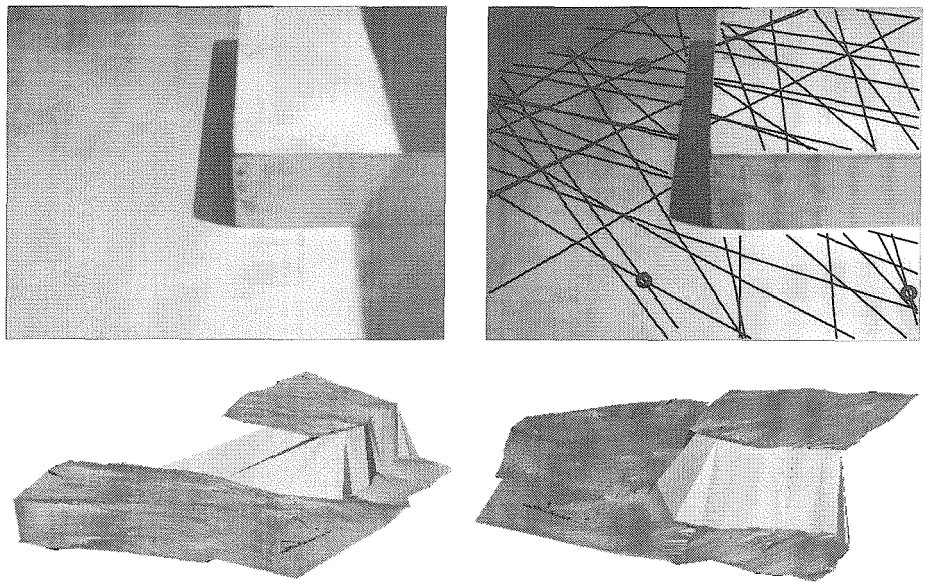


Figure 7.5: Experiment 1 - Two planes scene: Top row: The initial scene with a shadow projected on it and the total set of $N = 26$ shadow edges generating $N_p = 173$ intersection points. Bottom row: Two views of the final 3D reconstruction (in the form of a mesh). The processed images were 320×240.

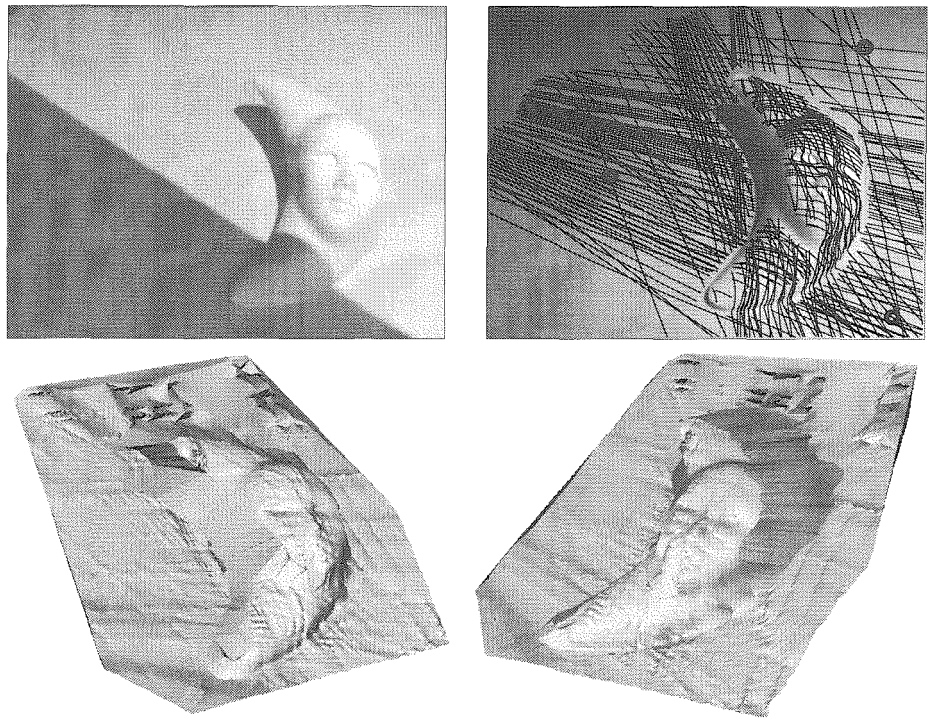


Figure 7.6: Experiment 2 - Luna scene: A total number of $N = 122$ shadow edges are intersecting at $N_p = 3056$ points.

scene 2. The height of the top plane of scene 1 is estimated after reconstruction to 4.6cm±3mm (the error accounts for the surface roughness), its real value being 5cm. Finally, in the first scene reconstruction, parallelism of the two planes is recovered within approximately 3 degrees of error.

One may notice that the reconstruction accuracies achieved on those two scenes are not as good as the ones achieved when using the original shadow scanning technique that we described in the previous chapter 6 (for example, compare figures 7.6 and 6.17). The main reason for that is in this present method, the shadow edge was extracted on the image through spatial processing (based on image gradient) instead of temporal processing as in the method presented in chapter 6. This illustrates the fact that temporal processing is more reliable than spatial processing because it is a lot less sensitive to changes to surface albedo and occlusions. This was originally demonstrated by Curless and Levoy in [84]. Nevertheless, it is not strictly possible to compare reconstruction accuracies of the two shadow scanning methods, given that we used here very few shadow edges for shape estimation (an order of $N = 100$ edges) while in the previous shadow scanning technique (chapter 6), an order of 700 to 1000 images are often necessary to achieve good quality reconstructions.

7.4 Generalization to multiple light sources

In order to maximize the surface coverage of the scanning, it sometimes useful to use multiple light sources. That technique is most useful when the surface of the scene is not convex. Then, a question arises: When using more than one calibrated light source for planar shadow scanning, is there more information contained in the shadow edges that allows to achieve better than Euclidean reconstruction up to three scalar coefficients?

Let us first consider the case where two families of shadow edges are generated by two calibrated light sources. Then, it is interesting to observe that the rank of the matrix $\mathbf{A}$ may be increased to $2N - 2$ (its minimum value is $2N - 3$). Under this condition, Euclidean reconstruction is achievable up to two scalar coefficients only.

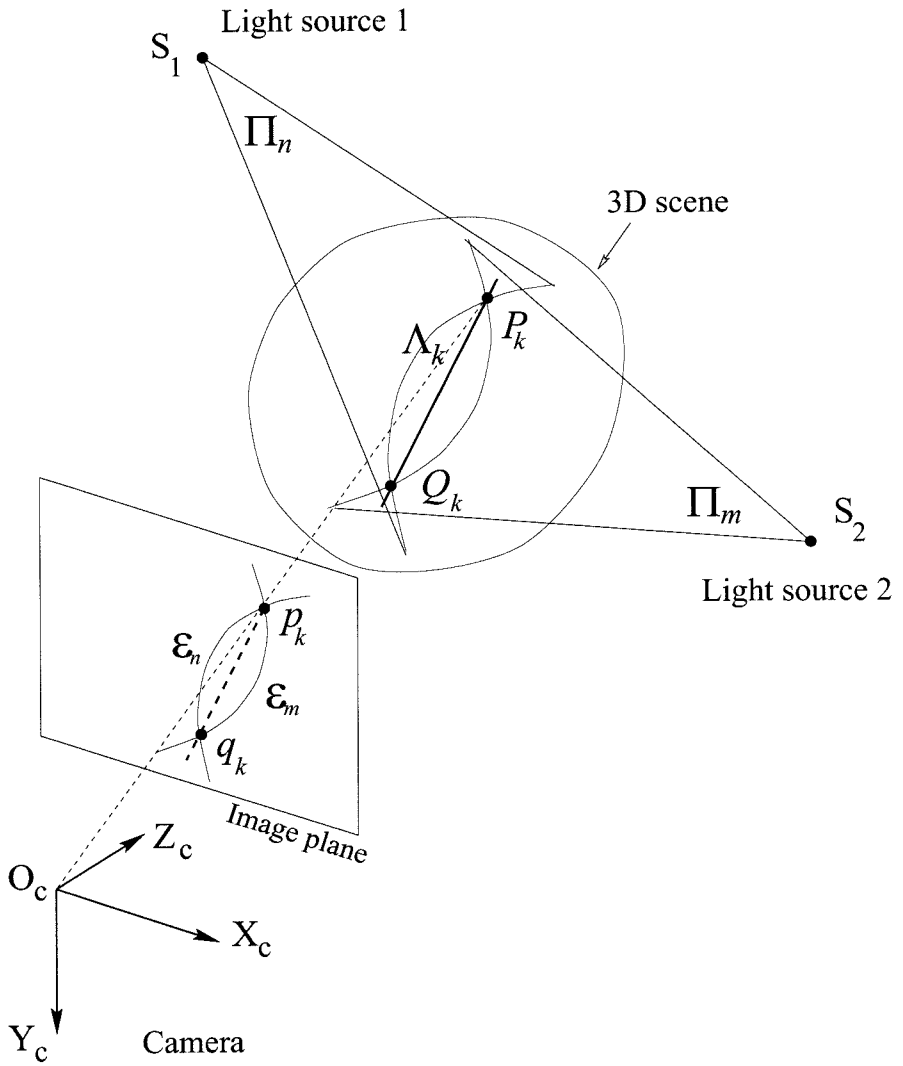


Figure 7.7: Double edge intersection: The two shadow planes Π_n and Π_m are generated by the two light sources S_1 and S_2. The two corresponding shadow edges $\mathcal{E}_n$ and $\mathcal{E}_m$ intersect on the image plane at the two points p_k and q_k. Depth information at p_k and q_k propagates along $\mathcal{E}_n$ and $\mathcal{E}_m$. Euclidean reconstruction is then achieved up to two scalar coefficients.

This situation occurs when at least two shadow planes generated by the two light sources intersect at at least two points in the scene*. Then, depth information at those two points will propagate along the two edges (following Property 1 of section 7.2.1), and then to the rest of the edge web (following Property 2 of section 7.2.1). This principle is illustrated on figure 7.7. On this figure, the two shadow planes Π_n and Π_m are generated by the two light sources S_1 and S_2. The two corresponding shadow edges $\mathcal{E}_n$ and $\mathcal{E}_m$ intersect on the image plane at the two points p_k and q_k. Depth information at p_k and q_k propagates along $\mathcal{E}_n$ and $\mathcal{E}_m$. One question remains: If more pairs of edges intersect at more than one point in the image, can the dimension of the null space of $\mathbf{A}$ be further reduced to 1 or 0? We do not have an answer to that question yet. This would be part of future work. From numerical simulations, it appears that when there exists no pair of shadow edges intersecting at more than one point in the image†, the rank of $\mathbf{A}$ remains to $2N - 3$ (this result appears to generalize to more than 2 light sources). We also keep the proof of that statement as part of future work.

When three calibrated light sources are used for shadow scanning, the matrix $\mathbf{A}$ may become full rank, leading to a direct Euclidean reconstruction of the scene. This situation occurs when three shadow planes generated by the three light sources mutually intersect at at least two points in the image (inducing a total of 6 intersections). Then, the total number of unknowns (2 N = 6) equals the number of scalar constraints (one constraint per intersection point) leading to a unique solution for the shadow plane locations, and therefore for Euclidean reconstruction. Is it a necessary condition for direct Euclidean reconstruction? Future investigations should bring an answer to that question.

*Observe that this situation cannot occur when a single light source is used.

†Notice that this will always happen when the scene is a single plane.

7.5 Conclusions

In this chapter, we have presented a linear closed-form solution for 3D scene geometry recovery from planar shadows only. The method is composed of two fundamental stages. The first one consists of retrieving the scene Euclidean geometry up to three scalar unknowns using only the information contained in the observed shadow edges on the image plane. The solution to that problem reduces to a singular value decomposition of a matrix that is only function of the edge intersection coordinates. In the second stage, the three remaining unknowns are computed making use of the known depths at only three points in the scene. Dual-space geometry provides an appropriate framework for carrying the complete mathematical analysis elegantly and intuitively. As part of future work, we intend to carry out a sensitivity analysis of the method, and study alternate geometrical clues for achieving Euclidean reconstruction (other than direct depth measurements at three points). It is also part of future investigations to extend the analysis of the reconstruction technique to cases where multiple light sources are used for shadow scanning.

In addition, we intend to merge this reconstruction technique with the one presented in the previous chapter for achieving best local surface reconstruction qualities (taking advantage of temporal processing) while dealing with scenes that do not contain any reference surface.

Chapter 8 Conclusion and future work

In this thesis, we have presented several techniques for extracting the three-dimensional shapes of objects for 3D modeling. These methods may be decomposed into passive and active techniques.

Passive techniques rely only on the information contained in the images acquired under natural uncontrolled lighting for reconstructing the 3D structure of the world. In our proposed setup (chapter 4), a single camera is freely moved around the object of interest as it acquires a set of images. A set of salient point features are then tracked on the images (using optical flow techniques) and their locations in space are computed by geometrical triangulation. This set of points in space constitutes then our 3D reconstructed model (for visualization purposes, it is often advantageous to connect them in a surface mesh). In order to perform 3D shape estimation, it is also necessary to compute the overall trajectory of the camera as it is moved in space. Our scheme also includes a method for computing the trajectory of the camera that is sufficiently accurate for 3D shape estimation, even in the challenging case of long image sequences.

There are three main advantages of passive techniques. First, no more than one camera is necessary for shape acquisition (unlike stereo systems that require at least two calibrated cameras). Second, as the camera spans the entire surface of the object, a globally consistent reconstruction is achieved. It is then not necessary to register multiple 3D views together in order to obtain a complete 3D geometrical model. Third, as the camera trajectory is also computed, it does not have to be monitored using specific mechanical hardware (such as a calibrated robot arm). This last feature is a significant ergonomical advantage as the overall size of the system is only limited by the size of the imaging sensor (e.g., CMOS sensor + lens) and that of the computing platform.

The main drawback of passive approaches is that they depend on the presence of

texture in the scene. Indeed, such techniques do not work on textured-less objects where no salient features (patches, points, lines, curves) can be selected and tracked reliably on the images (take the example of a camera approaching a uniform white wall). One option to solve that problem is to physically add texture the scene by pasting landmarks on the surfaces (this is precisely what was done on the walls of the corridor before acquiring the navigation sequence). Of course, such an alteration of the environment is not always possible (it is very unlikely that museums would let someone paste makers on their statues). Therefore, it is sometimes necessary to use alternative techniques to accommodate for the absence of texture in the scene to model. One solution is using active techniques. Active techniques are based on projecting patterns in the scene using an additional device, and infer 3D shape from the way the patterns "deform" on the objects. The artificial texture generated by the active device produces then sufficient salient image features allowing for dense 3D reconstruction. In this thesis, we have proposed three different active scanning techniques. The first method (chapter 5) is directly inspired from traditional structured lighting techniques, where a LCD (Liquid Crystal Display) projector and a camera are used. The novel aspect of our method is in the nature of the projected patterns used: grayscale patterns with sinusoidal brightness profiles*. This choice of projection (together with a novel type of processing) allowed us to compute scene depth at every pixel in the camera image. Experimental results were presented, as well as a characterization of the reconstruction method through a complete error analysis.

The projection device (LCD projector) is by far the most expensive component of the scanning system. The second active lighting method that we presented (chapter 6) provides an alternative scheme for 3D scanning that does not require any other device besides a camera. The main idea behind that technique is in using a combination of a standard light source (such as a desk lamp) and a pencil (or any other object with a straight edge) to cast planar shadows in the scene and infer its 3D geometry from the way the shadow naturally deforms on the objects in the scene. In this method, a reference surface (such as a desk plane) is also necessary. We demonstrated

*Standard techniques use binary patterns consisting of sharp stripes.

the convenience and accuracy of this new scanning technology with a number of experimental results, in indoor as well as outdoor scenarios. In addition, we fully characterized the performance of the method through a complete error analysis.

The last reconstruction method we proposed is an extension of the shadow scanning technique (chapter 7). In that case, we studied the case where no reference plane is present in the scene. In this context, we demonstrated that an entire reference surface is actually not necessary to achieve full Euclidean reconstruction of the scene, and we provided a compact and intuitive numerical algorithm for computing 3D shape from a set of shadow edges only. For that purpose, we made full use of a new mathematical formalism that we also introduced in this thesis, the dual-space formalism (chapter 2).

Regardless of which specific active method is used (standard active lighting approaches, or the ones we described here), the main advantage of active techniques is that dense 3D shape estimation may be achieved, even for textured-less objects. Sometimes, even some level of specularities on the surface may be handled (although entirely specular objects cannot be scanned with standard optical triangulation systems). On the other hand, most scanning techniques only produce partial 3D views of the scene to model (also called range data). Then, in order to retrieve a complete 3D model, it is often necessary to merge several 3D views together into a consistent mesh (this phase is not needed in the first passive technique we presented). That process is very often time consuming because it requires some significant amount of manual intervention (to this day, there exists no fully automatic mesh alignment technique), and it is also very sensitive to calibration errors (even a slight deformation of each individual 3D view due to calibration errors may forbid any alignment procedure to work at all). To this day, more that 80% of the overall modeling time is spent on 3D view registration and global meshing.

It is an interesting research direction to develop a hybrid scanning technology that would merge the advantages of both active and passive worlds. A passive component of the system would first retrieve (through an orbital camera motion for example) a coarse resolution model of the scene. This 3D model would have the advantage

of being globally consistent. Then, individual fine 3D scans of the scene could be acquired using an active lighting system, and then "pasted"[†] onto the initial coarse 3D skeleton. We believe that such an alignment procedure would not suffer as much from global divergence, since a unique 3D structure would be used as reference.

In order to render scenes in a photorealistic way, it is also necessary to incorporate surface properties to the geometrical model. These properties mainly consist of the object surface texture, and the surface reflectance function.

[†]This pasting step would possibly include a deformation to compensate for global distortions due to calibration errors.

Bibliography

[1] L. Spillmann and J.S. Werner, *Visual Perception - The Neurophysiological Foundations*, Academic Press, Inc. Harcourt Brace Jovanovich, Publishers, 1990.

[2] Brian A. Wandell, *Fundations of Vision*, Sinauer Associates, Inc., Sunderland, Massachusetts, 1995.

[3] David H. Hubel, *Eye, Brain and Vision*, Scientific American Library, New York, 1995.

[4] Paul Besl, *Advances in Machine Vision*, chapter 1 - Active optical range imaging sensors, pages 1–63, Springer-Verlag, 1989.

[5] A. Gruss, S. Tada, and T. Kanade, "A VLSI smart sensor for fast range imaging," in *DARPA93*, pages 977–986, 1993.

[6] R. A. Jarvis, "A perspective on range-finding techniques for computer vision," *IEEE Trans. Pattern Analysis Mach. Intell.*, 5:122–139, March 1983.

[7] Marjan Trobina, "Error model of a coded-light range sensor," Technical Report BIWI-TR-164, ETH-Zentrum, 1995.

[8] Y.F. Wang, "Characterizing three-dimensional surface structures from visual images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(1):52–60, 1991.

[9] Reinhard Klette, Karsten Schluns, and Andreas Koschan, *Computer Vision : Three-Dimensional Data from Images*, Springer Verlag, 1998.

[10] O.D. Faugeras, *Three-Dimensional Vision, a Geometric Viewpoint*, MIT Press, 1993.

[11] B. Horn, *Robot vision*, MIT press, 1986.

[12] R.L. Bishop and S.I. Goldberg, *Tensor analysis on manifold*, Dove Publications, 1980.

[13] J.W. Bruce, "Lines, surfaces and duality," Technical report, Dept. of Pure Mathematics, University of Liverpool, 1992.

[14] J.G Semple and G.T. Kneebone, *Algebraic Projective Geometry*, Oxford, England: Clarendon Press, 1952.

[15] Geoffrey Cross and Andrew Zisserman, "Quadric Reconstruction from Dual-Space Geometry," *Proc. 6th Int. Conf. Computer Vision, Bombay, India*, pages 25–31, 1998.

[16] Kalle Astrom and Fredrik Kahl, "Motion Estimation in Image Sequences Using the Deformation of Apparent Contours," *Int. J. of Computer Vision*, 21(2):114–127, 1999.

[17] D. C. Brown, "Calibration of close range cameras," *Proc. 12^{t}h Congress Int. Soc. Photogrammetry, Ottawa, Canada*, 1972.

[18] R.Y. Tsai, "A versatile camera calibration technique for high accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses," *IEEE J. Robotics Automat.*, RA-3(4):323–344, 1987.

[19] R.Y. Tsai, "An efficient and accurate camera calibration technique for 3d machine vision," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Miami Beach, FL*, pages 364–374, 1986.

[20] Y.I. Abdel-Aziz and H.M. Karara, "Direct linear transformation into object space coordinates in close-range photogrammetry," *Proc. ASP Symposium on Close-Range Photogrammetry, Urbana, Illinois, USA*, pages 1–18, 1971.

[21] C. Harris and M.J. Stephens, "A combined corner and edge detector," in *Alvey88*, pages 147–152, 1988.

[22] B. Caprile and V. Torre, "Using vanishing points for camera calibration," *IJCV*, 4(2):127–140, 1990.

[23] W. Chen and B.C. Jiang, "3-d camera calibration using vanishing point concept," *PR*, 24:57–67, 1991.

[24] K. Daniilidis and J. Ernst, "Active intrinsic calibration using vanishing points," *PRL*, 17(11):1179–1189, 1996.

[25] G.-Q. Wei, Z. He, and S.D. Ma, "Camera calibration by vanishing point and cross ratio," *in Proc. IEEE Int. Conf. Acoust. Speech and Signal Processing*, pages 331–340, May 1989.

[26] R.M. Haralick, "Determining camera parameters from the perspective projection of a rectangle," *PR*, 22:223–230, 1989.

[27] R.M. Haralick and L.G. Shapiro, *Computer and Robot Vision*, Addision-Wesley Publishing Company, 1993.

[28] L.L. Wang and W.H. Tsai, "Computing camera parameters using vanishing-line information from a rectangular parallelepiped," *MVA*, 3(3):129–141, 1990.

[29] L.L. Wang and W.H. Tsai, "Camera calibration by vanishing lines for 3-d computer vision," *PAMI*, 13(4):370–376, 1991.

[30] Olivier Faugeras and Luc Robert, "What can two images tell us about a third one?," *Int. J. of Computer Vision*, 18(1):5–19, 1996.

[31] Marc Pollefeys and Luc Van Gool, "A stratified approach to metric self-calibration," *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn.*, pages 407–412, 1997.

[32] Carlo Tomasi and Takeo Kanade, "Detection and tracking of point features," Technical Report CMU-CS-91-132, Carnegie Mellon University, 1991.

[33] B.D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," *Proc. 7th Int. Conf. on Art. Intell.*, 1981.

[34] Jianbo Shi and Carlo Tomasi, "Good features to track," *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn.*, pages 593–600, 1994.

[35] J. Barron, D. Fleet, and S. Beauchemin, "Performance of optical flow techniques," *Int. J. of Computer Vision*, 12(1):43–78, 1994.

[36] H. C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, 293:133–135, 1981.

[37] B.K.P. Horn, "Relative orientation," *Int. J. of Computer Vision*, 4:59–78, 1990.

[38] B.K.P. Horn, "Relative orientation revisited," *J. Opt. Soc. Am. A*, 8(19):1630–1638, 1991.

[39] O. Faugeras and S. Maybank, "Motion from point matches: multiplicity of solutions," *Int. J. of Computer Vision*, 4:225–246, 1990.

[40] Y. Liu, T.S. Huang, and O.D. Faugeras, "Determination of camera location from 2d to 3d line and point correspondences," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):28–37, 1990.

[41] A.D. Jepson and Heeger, "Linear subspace methods for recovering rigid motion," *Spatial Vision in Human and Robots, Cambridge University Press*, 1992.

[42] D.J. Heeger and A.D. Jepson, "Subspace methods for recovering rigid motion i," *Int. J. of Computer Vision*, 7(2):95–117, 1992.

[43] D. Heeger and A. Jepson, "Subspace methods for recovering rigid motion i: algorithm and implementation," RBCV TR-90-35, University of Toronto – CS dept., November 1990, Revised July 1991.

[44] R.I. Hartley, "In defence of the 8-point algorithm," *Proc. 5^{th} Int. Conf. Computer Vision, Boston, USA*, pages 1064–1070, 1995.

[45] Stefano Soatto, *A Geometric Framework for Dynamic Vision*, Ph.D. thesis, California Institute of Technology, May 1996.

[46] S. Soatto, R. Frezza, and P. Perona, "Recursive motion estimation on the essential manifold," in *Proc. 3rd Europ. Conf. Comput. Vision, J.-O. Eklundh (Ed.), LNCS-Series Vol. 800-801, Springer-Verlag*, pages II–61–72, 1994.

[47] S. Soatto, R. Frezza, and P. Perona, "Motion estimation via dynamic vision," *IEEE Trans. Automatic Control*, 41(3):393–414, 1996.

[48] S. Soatto and P. Perona, "Recursive 3d visual motion estimation using subspace constraints," *Int. J. of Computer Vision*, 3(22):235–259, 1997.

[49] S. Soatto and P. Perona, "Reducing "structure from motion": a general framework for dynamic vision. part 1: modeling.," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(9):993–942, 1998.

[50] S. Soatto and P. Perona, "Reducing "structure from motion": a general framework for dynamic vision. part 2: Implementation and experimental assessment.," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(9):943–960, 1998.

[51] J-Y. Bouguet and P. Perona, "Visual navigation using a single camera," in *Proc. 5th Int. Conf. Computer Vision, Boston, USA*, pages 645–652, Cambridge, Mass, 1995.

[52] R.E. Kalman, "A new approach to linear filtering and prediction problems.," *Trans. of the ASME-Journal of basic engineering.*, 35-45, 1960.

[53] R.S. Bucy, "Non-linear filtering theory," *IEEE Trans. A.C. AC-10, 198*, 1965.

[54] R. Hartley, "Lines and points in three views - a unified approach," *Proc. Image Understanding Workshop, Monterey, California*, pages 1009–1016, 1994.

[55] J. Weng, T.S. Huang, and N. Ahuja, "Motion and structure from line correspondences: closed-form solution, uniqueness and optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(3):318–336, 1992.

[56] A. Shashua, "Trilinearity in visual recognition by alignment," *Proc. 3rd Europ. Conf. Comput. Vision, J.-O. Eklundh (Ed.), LNCS-Series Vol. 800-801, Springer-Verlag*, pages 479–484, 1994.

[57] T. Vieville, T.S. Huang, and N. Ahuja, "Motion of points and lines in the uncalibrated case.," Technical Report RR-2054, INRIA, 1993.

[58] T. Papadopoulo and O. Faugeras, "A new characterization of the trifocal tensor," *Proc. 5th Europ. Conf. Comput. Vision, Burkhardt and Neumann (Ed.), LNCS-Series Vol. 1406-1407, Springer-Verlag*, pages 109–123, 1998.

[59] G.P. Stein and A. Shashua, "On degeneracy of linear reconstruction from three views: Linear line complex and applications," in *Proc. 5th Europ. Conf. Comput. Vision, Burkhardt and Neumann (Ed.), LNCS-Series Vol. 1406-1407, Springer-Verlag*, pages 862–878, 1998.

[60] Olivier Faugeras and Bernard Mourrain, "On the geometry and algebra of the point and line correspondence between n images," *Proc. 5th Int. Conf. Computer Vision, Boston, USA*, pages 851–856, 1994.

[61] Shai Avidan and Amnon Shashua, "Threading fundamental matrices," *Proc. 5th Europ. Conf. Comput. Vision, Burkhardt and Neumann (Ed.), LNCS-Series Vol. 1406-1407, Springer-Verlag*, pages 124–140, 1998.

[62] A. Shashua, "Algebraic functions for recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):779–789, 1995.

[63] Bill Triggs, "The geometry of projective reconstruction i: Matching constraints and the joint image.," Technical report, INRIA, 1994.

[64] R. I. Hartley, "Computation of quadrifocal tensor," *Proc. 5th Europ. Conf. Comput. Vision, Burkhardt and Neumann (Ed.), LNCS-Series Vol. 1406-1407, Springer-Verlag*, pages 20–35, 1998.

[65] Anders Heyden, "A common framework for multiple view tensors," *Proc. 5th Europ. Conf. Comput. Vision, Burkhardt and Neumann (Ed.), LNCS-Series Vol. 1406-1407, Springer-Verlag*, pages 3–19, 1998.

[66] F.P. Preparata and M.I. Shamos, *Computational Geometry*, Springer-Verlag, NY, 1988.

[67] G. Sansoni, S. Corini, S. Lazzari, R. Rodella, and F. Docchio, "Three-dimensional imaging based on gray-code projection: characterization of the measuring algorithm and development of a measuring system for industrial applications," *Applied Optics*, 36(19):4463–4472, 1997.

[68] D.C Douglas Hung, "3d scene modelling by sinusoid encoded illumination," *Image and Vision Computing*, 11(5):251–256, June 1993.

[69] T.G. Stahs and F.M. Wahl, "Fast and robust range data acquisition in a low-cost environment," *Proc. SPIE 1395*, pages 496–503, 1990.

[70] W. Krattenthaler, K.J. Mayer, and H.P. Duwe, "3d-surface measurement with coded light approach," *Proc. Oesterr. Arbeitsgem. MustererKennung*, 12:103–114, 1993.

[71] P.J. Besl and N.D. McKay, "A method for registration of 3-d shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[72] Berthold K.P. Horn, *Robot Vision*, MIT Press, 1986.

[73] Jean-Yves Bouguet and Pietro Perona, "3D photography on your desk," *Proc. 6th Int. Conf. Computer Vision, Bombay, India*, pages 43–50, January 1998.

[74] Jean-Yves Bouguet and Pietro Perona, "3D photography on your desk," Technical report, California Institute of Technology, 1997, available at: http://www.vision.caltech.edu/bouguetj/ICCV98.

[75] Jean-Yves Bouguet and Pietro Perona, "3D photography using shadows," *EUSIPCO, Island of Rhodes, Greece*, pages 1273–1276, September 1998.

[76] Jean-Yves Bouguet and Pietro Perona, "3D photography using shadows," *Proc. IEEE International Symposium on Circuits and Systems, Monterey, USA*, June 1998.

[77] Jean-Yves Bouguet and Pietro Perona, "3D photography using shadows in dual space geometry," *Submitted to International Journal of Computer Vision*, 1999.

[78] A.A. Goshtasby, S. Nambala, W.G. deRijk, and S.D. Campbell, "A system for digital reconstruction of gypsum dental casts," *IEEE Transactions on Medical Imaging*, 16(5):664–674, October 1987.

[79] Sylvain Bougnoux, "From projective to euclidean space under any practical situation, a criticism of self-calibration," *Proc. 6^{th} Int. Conf. Computer Vision, Bombay, India*, pages 790–796, January 1998.

[80] Reinhard Koch, Marc Pollefeys, and Luc Van Gool, "Multi viewpoint stereo from uncalibrated video sequence," *Proc. 5^{th} European Conf. Computer Vision, Freiburg, Germany*, pages 55–71, June 1998.

[81] Marc Pollefeys, Reinhard Koch, and Luc Van Gol, "Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters," *Proc. 6^{th} Int. Conf. Computer Vision, Bombay, India*, pages 90–95, January 1998.

[82] D. C. Brown, "Analytical calibration of close range cameras," *Proc. Symp. Close Range Photogrammetry, Melbourne, FL*, 1971.

[83] G.P. Stein, "Accurate internal camera calibration using rotation, with analysis of sources of error," in *Proc. 5th Int. Conf. Computer Vision, Boston, USA*, pages 230–236, 1995.

[84] Brian Curless and Marc Levoy, "Better optical triangulation through spacetime analysis," *Proc. 5th Int. Conf. Computer Vision, Boston, USA*, pages 987–993, 1995.

[85] T. Kanade, A. Gruss, and L. Carley, "A very fast VLSI rangefinder," in *IEEE International Conference on Robotics and Automation*, volume 39, pages 1322–1329, April 1991.

[86] J.F. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[87] Jurgen R. Meyer-Arendt, "Radiometry and photometry: Units and conversion factors," *Applied Optics*, 7(10):2081–2084, 1968.

[88] John W. T. Walsh, *Photometry*, Dover, NY, 1965.

[89] Athanasios Papoulis, *Probability, Random Variables and Stochastic Processes*, Mac Graw Hill, 1991, Third Edition, page 187.

[90] Silvio Savarese, "Scansione tridimensionale con metodi a luce debolmente strutturata," *Tesi di Laurea, Universita degli Studi di Napoli Federico II*, 1998.

[91] H. Gagnon, M. Soucy, R. Bergevin, and D. Laurendeau, "Registration of multiple range views for automatic 3-D model building," *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn.*, pages 581–586, June 1994.

[92] G. Turk and M. Levoy, "Zippered polygon meshes from range images," *In SIGGRAPH '94*, pages 311–318, July 1994.

[93] C.L. Bajaj, F. Bernardini, and G. Xu Xu, "Automatic reconstruction of surfaces and scalar fields from 3D scans," *In SIGGRAPH '95, Los Angeles, CA*, pages 109–118, August 1995.

[94] Brian Curless and Marc Levoy, "A volumetric method for building complex models from range images," *SIGGRAPH96, Computer Graphics Proceedings*, 1996.

[95] Jean-Yves Bouguet, Markus Weber, and Pietro Perona, "What do planar shadows tell us about scene geometry?," *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn.*, 1999, To appear.