Universidade Técnica de Lisboa

Instituto Superior Técnico

Stereo Reconstruction of a Submerged Model Breakwater and Interface Estimation

Ricardo Jorge dos Santos Ferreira
(Licenciado)

Dissertação para a obtenção do Grau de Mestre em
Eng. Electrotécnica e de Computadores

Orientador: Doutor João Paulo Salgado Arriscado Costeira

Júri

Presidente: Doutor João Paulo Salgado Arriscado Costeira

Vogais: Doutor Hélder de Jesus Araújo
Doutor Carlos Jorge Ferreira Silvestre
Doutor Pedro Manuel Quintas Aguiar

Março de 2006

Abstract

The present work is dedicated to the study of refraction effects between two media in stereo reconstructions of a three-dimensional scene. Refraction induces nonlinear effects on the observed image resulting in a highly complex stereo matching process. The proposal is to use a linear, first order Taylor approximation, which maps this problem into a new problem with a conventional solution, feasible around a particular image point. Images are transformed (corrected) before entering any of the known stereo matching algorithms. The final step of converting disparity to world coordinates must also be properly adapted.

An interface estimation algorithm that estimates its shape from stereo image pairs is also presented. It assumes the submerged scenery is known so it works best when a highly textured plane is used. The algorithm consists of a cost function for the interface to pass through a particular point in space. Minimization of this cost function in the presence of smoothness constraints (for example using dynamic programming like algorithms) results in the global optimum surface.

For the two algorithms, results are presented taken both from synthetic images generated by a raytracer and results from real life scenes observing an actual model breakwater.

Keywords: Interface, Reconstruction, Stereo, Calibration, Estimation

Resumo

O trabalho apresentado dedica-se ao estudo de efeitos de refracção entre dois meios em reconstruções stéreo de cenários tri-dimensionais. A refracção provoca efeitos não lineares na imagem observada, dificultando significativamente o processo de emparelhamento. Propõe-se o uso de uma aproximação de Taylor de primeira ordem, linear, que contorna o problema permitindo o uso de soluções convencionais. A solução é válida em torno de um dado ponto. As imagens são previamente transformadas (corrigidas) antes de se aplicar um algoritmo convencional de emparelhamento. O último passo que consiste em converter disparidade em coordenadas no mundo também necessita de ser adaptado.

Apresenta-se também um algoritmo de estimação da interface, que estima a sua forma a partir de pares de imagem stéreo. É assumido que o cenário submerso é conhecido portanto funcionando melhor quando uma superfície plana com uma textura rica é usada. O algoritmo consiste de uma função que atribui um custo para a interface passar num dado ponto do espaço. A minimização desta função de custo introduzindo restrições de suavidade (por exemplo usando algoritmos de programação dinâmica) resulta na superfície global óptima.

Para ambos os algoritmos apresentam-se resultados derivados de imagens sintéticas geradas por computador e de imagens reais que observam um modelo de quebra-mar.

Palavras-Chave: Interface, Reconstrução, Stéreo, Calibração, Estimação

Contents

1 Introduction **1**

1.1 Reconstruction of Submerged Scenes 2

1.2 Interface Estimation 3

1.3 Practical Considerations 4

1.4 Summary of Contributions 4

1.5 Organization of the Thesis 4

2 Preliminary Concepts and Theoretical Framework **5**

2.1 Typographical Conventions 5

2.2 Euclidean Spaces 6

2.2.1 Vector Space Structure for $\mathbb{E}^n$ 7

2.2.2 Charts 7

2.2.3 Tangent Vectors 8

2.2.4 Coordinate Transformations 9

2.3 Camera Model 10

2.4 Projective Space 12

2.5 Stereo System 14

3 Capturing Depth With a Stereo System: Standard Algorithms **17**

3.1 Image Acquisition 17

3.2 Calibration and Image Rectification 19

3.3 Stereo Matching Algorithms 21

3.3.1 Sparse Stereo 21

3.3.2 Dense Stereo 22

3.3.3 Two Dimensional Dense Matching 23

3.4 Reconstruction 26

4 Submerged Scenery Reconstruction **27**

4.1 Snell's Law 27
4.2 First Order Approximation 28
4.2.1 Geometric Interpretation 30
4.2.2 Correction Homography 32
4.3 Reconstruction 33
4.4 Summary of the proposed algorithm 34
4.5 Results 36

5 Interface Estimation **43**
5.1 Problem Formulation 43
5.2 Implementation Considerations 47
5.3 Results 48

6 Conclusion **53**

A Polynomial Regression **55**

B Intersection of Two Straight Lines **59**

List of Figures

1.1 Real breakwater 1
1.2 Model breakwater 2
1.3 Illustration of loss of stereo geometry. 3

2.2.1 Coordinate change 10
2.3.1 Projection and inclusion functions 11
2.4.1 Projective space explanation 13
2.5.1 Disparity chart. 15

3.1.1 Image acquisition hardware 18
3.1.2 Example of a computer generated image. 18
3.2.1 Image rectification results 20
3.3.1 Sparse stereo matching results. 22
3.3.2 Dense stereo matching results 23
3.3.3 Sensitivity of the dense matching algorithm in the presence of rectification errors 24
3.3.4 Cyclic algorithm for dense stereo matching in 2 dimensions 24
3.3.5 Disparity maps obtained using the cyclic algorithm 25
3.3.6 Disparity maps obtained with exhaustive search 26

4.1.1 Snell law in 3 dimensions. 27
4.2.1 First order Snell approximation error. 29
4.2.2 Interpretation of first order approximation of Snell's law 30
4.2.3 First order approximation of Snell's law for various angles. 31
4.3.1 Illustration of the reconstruction error using Snell correction. 33
4.4.1 Extrinsic Snell correction step and camera alignments. 35
4.5.1 Reconstruction visualization. 36
4.5.2 Camera position with respect to the interface. 36
4.5.3 Reconstruction error using first order Snell correction. 38
4.5.4 Reconstruction error using first order Snell correction. 39

4.5.5 Reconstruction error using first order Snell correction. 40
4.5.6 Render setup of a synthesised scenery. 40
4.5.7 Results of the reconstruction of a plane. 41
4.5.8 3D view and left image of a model breakwater partially submerged. 41
4.5.9 3D view and left image of another model breakwater partially submerged. . . . 42

5.1.1 Graphical representation of the possible media transition points. 44
5.1.2 Interface estimation algorithm representation. 45
5.1.3 Interface estimation error function. 46
5.2.1 Two sets of stereo pairs are needed. 47
5.3.1 Synthetic image used for interface estimation. 48
5.3.2 Results obtained using low pass filtering of the input disparity map. 48
5.3.3 Global interface estimation error. 49
5.3.4 Obtained results using polynomial regression of the input disparity map. 50
5.3.5 Interface estimation with images of a real breakwater model. 51

Chapter 1

Introduction

The use of breakwaters (figure 1.1) is of extreme importance in structures that come in direct contact with sea water. Harbors and airports are only a few applications providing almost an uncountable source of examples. Physical modelling is, still today, the main tool for testing and designing these coastal structures. The most important factor that leads to structure degradation and failure is the continuous wave action to which they are subject to. Thus, these structures require periodic labor throughout their usefull life span.

Currently, to test the resistance of a proposed design to wave action, a scale model of the structure is built in a wave tank, such as the one shown in figure 1.2. These models consist of scale reconstructions of actual structures which need to be studied for reliability and durability when subject to adverse conditions. They are then exposed to a sequence of surface waves generated by a wave paddle. One of the parameters that has proved of paramount importance in the forecast of the structure behavior is the profile erosion relative to the initial undamaged profile. Thus, measuring and detecting changes in the structure's envelope is of great importance.

Laser range finders are one obvious and easy way of reconstructing the scene, however, since common lasers do not propagate in the water, the tank has to be emptied every time a

Figure 1.1: Breakwater in Viana do Castelo (Portugal).

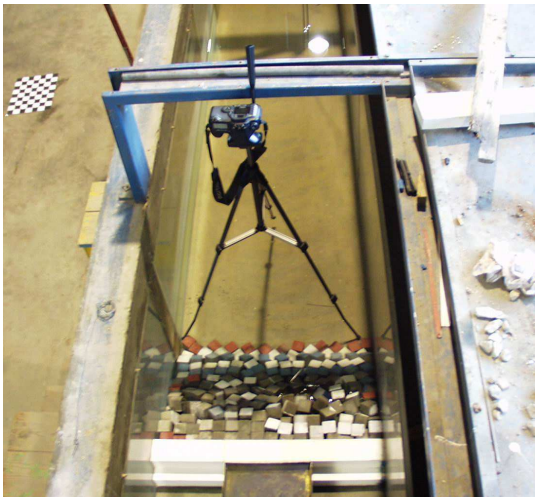


Figure 1.2: Model breakwater in a wave tank at Laboratório Nacional de Engenharia Civil (LNEC) in Portugal. The cameras are positioned above the submerged model as shown.

measurement is taken. This is a quite expensive procedure, both in time and money resources. The proposed solution is to use a stereo mechanism to reconstruct a submerged scene captured from cameras placed outside of the water. This way it's possible to monitor both the emerged and submerged part of the breakwater.

1.1 Reconstruction of Submerged Scenes

The intention of the present work is to develop tools capable of analyzing submerged objects. In particular, to be able to apply stereo reconstruction to images of model breakwaters making it possible to analyze the damage produced by repeated wave action. With this in mind, a camera stereo system is set up above the model (as shown in figure 1.2) and snapshots are taken before and after the experiment allowing it to be reconstructed and analyzed on a computer. The problem that arises in the presence of an interface between two media is that images captured by cameras suffer non-linear light bending effects when transversing the interface (see figure 1.3). The distortion, commonly known as refraction and modelled by Snell's law, forces some of the available stereo geometrical restrictions to be relaxed which would otherwise help in feature matching. This matching process is severely hindered by the lack of the known epipolar constraint. It will be shown that, if the incidence angle is small, the linear part of the Taylor Series expansion, which is equivalent to a modification of the camera's intrinsic parameters, is precise enough for the purposes discussed. In other words current stereo matching algorithms can be used, provided the camera orientation parameters are within a certain range.

Although stereo vision is already a well established field, there are no known works of

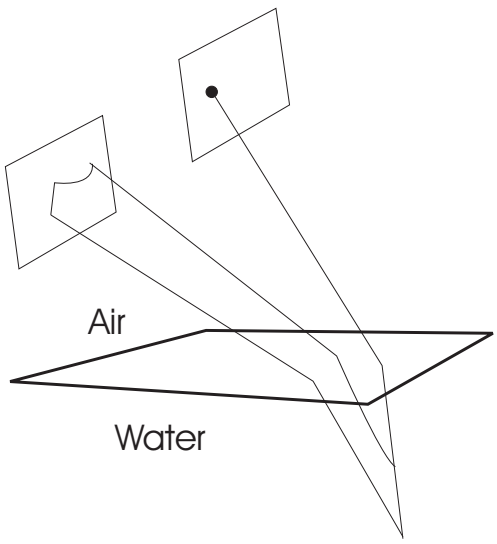


Figure 1.3: Illustration of loss of stereo geometry.

similar nature as most systems are placed underwater, eliminating the refraction issue. Young-Hoo Kwon seems to be one of the few to have approached the problem of submerged sceneries, in particular to study human motricity in swimming athletes. His method, mentioned in [1] and [2], consists of using current calibration algorithms in order to minimize mean square error on a given submerged volume. For this he uses a 3 dimensional grid that needs to be submerged during calibration. The work presented here describes an alternate and independent approach from the one described by Kwon.

An implementation with similar goals but with a much different approach, where the objective is to catalog and compare images taken periodically of south African breakwaters, is described in [3] An operator later registers significant changes by comparing the pictures taken at two different time instances.

1.2 Interface Estimation

Since the distortion introduced by the presence of the interface depends on its position, it is necessary to first develop a means to estimate it. A simple solution consists in calibrating the cameras' extrinsic parameters with a grid floating on the interface. Since scene reconstruction will only be attempted when the interface is in a still, planar configuration, this simple procedure is enough. A more generic solution allowing the estimation of the surface in almost any smooth configuration is also presented, making use of stereo image pairs taken while observing a known submerged scene. No reference in the literature was found that resembles this approach. Trying to solve both problems simultaneously (scenery reconstruction and surface

estimation) is a difficult problem, becoming practically impossible if the interface is not planar.

1.3 Practical Considerations

Another prejudicial effect which can render the acquired images useless unless special attention is given to lighting conditions during image acquisition is reflection. Although the use of polarized filters can help minimize the problem, it can be silently dismissed when dealing with controlled environments, since light sources can usually be submerged.

It is important to keep in mind that although the implementation focuses primarily on air/water interfaces, all results are valid for interfaces between any other media (as long as Snell's law applies). An example that comes to mind is an air/glass interface where it might be of interest to obtain the surface of a lens.

1.4 Summary of Contributions

There are two main contributions in this thesis. The first characterizes the distortion introduced by the the presence of an interface between two media and describes a correction that can be applied to obtained images to minimize it. In particular, it is shown how stereo reconstructions of submerged sceneries can be obtained. The second contribution uses the same distortion to reconstruct the interface's shape from observed pairs of images.

1.5 Organization of the Thesis

This work is structured as follows:

- **Chapter 2** introduces some necessary concepts and the adopted typographical notations.
- **Chapter 3** describes standard algorithms necessary for stereo reconstructions in general.
- **Chapter 4** adapts the algorithms described in chapter 3 for use when a planar interface is placed between the cameras and the scenery to be reconstructed.
- **Chapter 5** indicates how the interface's position can be estimated using stereo image pairs and a known correspondence with the submerged scenery.

Chapter 2

Preliminary Concepts and Theoretical Framework

The intention of this chapter is to introduce the notation and conventions adopted in the work that follows. Typographical conventions are presented first, followed by an explanation of what are considered necessary concepts. Although the convention might seem a little odd at first, it is the author's belief that it eases the description of the algorithms, allowing for details such as coordinate changes to be ignored until actual implementation. Mathematicians and physicists have for a long time used these coordinate free representations with great success. This chapter is included only as an introduction to the subject and is by no means an exhaustive approach of the matter. For an in-depth description see any of [4], [5], [6] or [7].

2.1 Typographical Conventions

$\mathbb{E}^n$	n-dimensional Euclidean space.
$\mathbb{R}^n$	The space of real n-tuples.
$\mathbb{P}^n$	n-dimensional projective space.
a	A real number (belongs to $\mathbb{R}$).
$\mathfrak{C}$	A coordinate chart.
$\mathbf{p}$	A point in $\mathbb{E}^n$ (or another manifold if indicated).
$\mathbf{v_p}$	A tangent vector at $\mathbf{p} \in \mathbb{E}^n$.
$\mathbf{T_p}\mathbb{E}^n$	The set of tangent vectors at $\mathbf{p} \in \mathbb{E}^n$.
${}^{\mathfrak{C}}p^i$	The ith coordinate of the point (or vector) $\mathbf{p}$ in the chart $\mathfrak{C}$.
${}^{\mathfrak{C}}\mathbf{p} = (a, b, c)$	(a, b, c) are the coordinates of $\mathbf{p}$ in $\mathfrak{C}$.
$\langle \cdot, \cdot \rangle$	Usual inner product.

$\mathbf{v}_1 \times \mathbf{v}_2$	Cross product of two vectors ($\mathbf{v}_1$,$\mathbf{v}_2 \in \mathbf{T}_\mathbf{p}\mathbb{E}^3$) at $\mathbf{p} \in \mathbb{E}^3$.
$\lvert\cdot\rvert$	Absolute value or matrix determinant.
$\lVert\cdot\rVert$	Induced norm of a vector $\left(\sqrt{\langle\cdot,\cdot\rangle}\right)$.
$\hat{\mathbf{v}}_\mathbf{p}$	Unit normed vector at $\mathbf{p} \in \mathbb{E}^n$, that is $\langle\hat{\mathbf{v}},\hat{\mathbf{v}}\rangle = 1$.
$\bar{\mathbf{p}}$	A point in $\mathbb{P}^n$.
f	A function.
$\mathscr{P}^{mn}$	The set of bivariate polynomials of order mn.
$\mathbf{M}$	A matrix.
$\mathcal{P}_P(i,j,K)$	Set of rank K partial permutation matrices, of size $i \times j$
$\sim$	Equivalent to (same equivalence class)
$\propto$	Proportional to
$\approx$	Approximately equal to
$\equiv$	Equivalent to
$\cong$	Isomorphic to

2.2 Euclidean Spaces

This document will focus primarily on two Euclidean spaces, namely $\mathbb{E}^2$ and $\mathbb{E}^3$. The second is where the scenery exists, and the first will contain a given projection of the scenery on a plane. It is important to realize that a point in $\mathbb{E}^n$ is not an n-tuple of coordinates, although it can be represented as such given a chart. If $\mathbf{p} \in \mathbb{E}^n$ and a one-to-one mapping $\mathfrak{C} : W \subset \mathbb{E}^n \longrightarrow U \subset \mathbb{R}^n$ is given, where W and U are open subsets, then ${}^{\mathfrak{C}}\mathbf{p} \equiv \mathfrak{C}(\mathbf{p})$ is a coordinate representation for $\mathbf{p}$. Note that if $\mathfrak{D}$ is another such one-to-one mapping, ${}^{\mathfrak{D}}\mathbf{p}$ will also be a coordinate representation for the same $\mathbf{p}$. Under certain conditions guaranteeing continuity and differentiability these one-to-one mappings are called charts and will be further discussed later.

Although this document deals with only one copy of $\mathbb{E}^3$ (the ambient space, where the scenery lives), there are multiple copies of $\mathbb{E}^2$ since multiple projections (images) are considered. So that no ambiguity exists as to which of these spaces (images) is meant, different spaces are given different indexes, for example $\mathbb{E}^2_\mathfrak{C}$ and $\mathbb{E}^2_\mathfrak{D}$. The typographic convention of the subscripts used will be made clearer when discussing the camera model.

It is assumed that a unit distance in the ambient space $\mathbb{E}^3$ is chosen beforehand and that orthogonality is also agreed upon.

2.2.1 Vector Space Structure for $\mathbb{E}^n$

$\mathbb{E}^n$ may be identified with a vector space once an action and an origin are chosen, where as expected multiplication by a scalar is identified with scaling centered at the origin and addition is identified as translation. For this construction let V^n be an n-dimensional vector space. Since a vector space defines an abelian group under addition, define an action of this group that is transitive and free[1] on $\mathbb{E}^n$:

$$\begin{aligned} \oplus : V^n \times \mathbb{E}^n &\longrightarrow \mathbb{E}^n \\ (v, \mathbf{p}) &\longmapsto \mathbf{p} \oplus v \end{aligned}$$

such that if $\mathbf{p} \in \mathbb{E}^n$ and $u, v \in V^n$ then

$$(\mathbf{p} \oplus v) \oplus w = \mathbf{p} \oplus (v + w)$$

Once a point $\mathbf{p}_0$ is fixed as the origin of $\mathbb{E}^n$, any point $\mathbf{p}$ is identified with a vector $v \in V^n$ such that

$$\mathbf{p} = \mathbf{p}_0 \oplus v$$

Transitivity guarantees that all elements of $\mathbb{E}^n$ are identified in this manner, and freeness guarantees a unique identification. Note that the choice of an action is not unique, nor the choice of an origin. These vector space constructions will be given special importance when choosing charts.

The symbol $\oplus$ will henceforth be silently substituted by the preferred symbol $+$.

2.2.2 Charts

Charts are defined as being continuous bijective maps with continuous inverse from an open set of $\mathbb{E}^n$ to an open set of $\mathbb{R}^n$, and they are given the important role of assigning coordinates to points. It is also required that all charts be granted certain smoothness properties, in particular that the coordinate change functions that will be described in section 2.2.4 must be of class C^∞. Although it is possible to choose many different charts, some are of special interest since they simplify the coordinate representation for $\mathbb{E}^n$. In particular if $\mathbb{E}^n$ is given a vector space structure as described in section 2.2.1 and the choice of action of V^n on $\mathbb{E}^n$ is such that the usual dot product on V^n agrees with the notion of orthogonality on $\mathbb{E}^n$ and the choice of unit

[1] Given an action of a group G on a set E it is said to be transitive if $\forall p_1, p_2 \in E$ there's an element $g \in G$ such that $g \cdot p_1 = p_2$. In other words if any element from the set can be taken to any other by the action of an element in G. It is said to be free if for any $p \in E$ the only element of G that fixes it is the identity: $g \cdot p = p \Longrightarrow g = e$.

length, the identification of E^n with V^n induces an isometry which can be used as a chart. Such charts will be referred to as cartesian (or orthonormal) charts and are the prefered choice when doing computations.

In a cartesian chart $\mathfrak{C}$ a point $\mathbf{p} \in \mathbb{E}^2$ will have coordinates x and y (referring to ${}^{\mathfrak{C}}p^1$ and ${}^{\mathfrak{C}}p^2$ respectively) as long as there's no ambiguity as to which referential is meant. If the point belongs to $\mathbb{E}^3$ it will also have the z coordinate.

Throughout this document there'll be 4 particular charts for $\mathbb{E}^3$ which are to be kept in mind. These are:

- $\mathfrak{W}$ - World cartesian chart. Normally this chart is chosen when camera calibration is performed. Camera calibration also guarantees that this is a cartesian chart (by construction). Although any point in $\mathbb{E}^3$ can be chosen as the origin and many vector space structures can be assigned, some computations will be easier if a particular one is chosen. For example computations can be greatly simplified when considering a planar interface by having it described by the plane equation $z = 0$ on a chart.

- $\mathfrak{L}$ and $\mathfrak{R}$ - Cartesian charts describing the left and right camera position in space. Since calibrated stereo is considered, each of these charts describe a projection center and an image plane. This will be further discussed in section 2.3.

- $\mathfrak{D}$ - Disparity chart. This is not a cartesian chart but it is important since it arises naturally on a stereo setup. It will be further described in section 2.5.

And two charts should always be present for each copy of $\mathbb{E}^2$ associated to each camera projection as will be described in detail in section 2.3:

- $\mathfrak{p}$ - Projection chart that appears naturally considering that $\mathbb{E}^2$ comes from a projection of $\mathbb{E}^3$.

- $\mathfrak{i}$ - This is where physical image pixels are measured, called the image chart. It differs from the former by the camera's intrinsic parameters.

2.2.3 Tangent Vectors

A vector at a point $\mathbf{p} \in \mathbb{E}^n$ is commonly viewed as an oriented line segment based at $\mathbf{p}$. This intuitive description is discarded in favor of a more general definition describing tangent vectors as derivatives of curves in space. Let $c :]-\epsilon, \epsilon[\longrightarrow \mathbb{E}^n$ be any smooth curve[2] in space such that

[2] A curve $c(t)$ is smooth if given any coordinate chart $\mathfrak{C}$, $\mathfrak{C}(c(t))$ is smooth.

$c(0) = \mathbf{p}$. A vector at $\mathbf{p}$ is defined as

$$\mathbf{v}_{\mathbf{p}} = \left.\frac{d}{dt}\right|_{t=0} c(t)$$

Note that this is only a notation for a derivative done in any coordinate chart. The tangent space at $\mathbf{p}$, denoted $\mathbf{T}_{\mathbf{p}}\mathbb{E}^n$, is the set of all vectors constructed from curves such that $c(0) = \mathbf{p}$. This space is actually a vector space at each point. Note though that addition of vectors at different points in space is not defined by this construction. For details, in order of least to most mathematically inclined, see any of [4], [5], [6] or [7].

It is interesting to note how to recover the notion of tangent vectors as oriented line segments from this definition. Choosing any vector space structure for $\mathbb{E}^n$ (as mentioned in the section 2.2.1), consider the family of functions parameterized by $\mathbf{p}_1, \mathbf{p}_2 \in \mathbb{E}^n$

$$f_{\mathbf{p}_1,\mathbf{p}_2}(t) = \mathbf{p}_1 + t\left(\mathbf{p}_2 - \mathbf{p}_1\right)$$

then its tangent vector at $\mathbf{p}_1$ is given by

$$\left.\frac{d}{dt}\right|_{t=0} f_{\mathbf{p}_1,\mathbf{p}_2}(t) = \mathbf{p}_2 - \mathbf{p}_1$$

So tangent vectors when $\mathbb{E}^n$ is given a vector space structure are nothing else than the usual interpretation given to them

$$\mathbf{v}_{\mathbf{p}_1} = \mathbf{p}_2 - \mathbf{p}_1$$

The seemingly superfluous definition here presented is needed to guarantee consistency when dealing with coordinate changes once charts are defined in the next section.

Note that the same action described in the last section is used to sum vectors to points anytime this is needed since vectors on a vector space are naturally isomorphic to the vector space itself. This is used for example to parametrically define straight lines.

2.2.4 Coordinate Transformations

Since a point can be described in different charts, there will be functions which change its coordinate representation. So, if $\mathfrak{R}$ and $\mathfrak{L}$ are two charts for a given point $\mathbf{p} \in \mathbb{E}^n$, define ${}^{\mathfrak{L}}_{\mathfrak{R}}\mathcal{E} : U \subset \mathbb{R}^n \to V \subset \mathbb{R}^n$ as ${}^{\mathfrak{L}}_{\mathfrak{R}}\mathcal{E} = \mathfrak{L} \circ \mathfrak{R}^{-1}$, such that

$${}^{\mathfrak{L}}\mathbf{p} = {}^{\mathfrak{L}}_{\mathfrak{R}}\mathcal{E}\left({}^{\mathfrak{R}}\mathbf{p}\right)$$

Thus ${}^{\mathfrak{L}}_{\mathfrak{R}}\mathcal{E}$ is the coordinate change from $\mathfrak{R}$ to $\mathfrak{L}$ (see figure 2.2.1 for a representation). It is

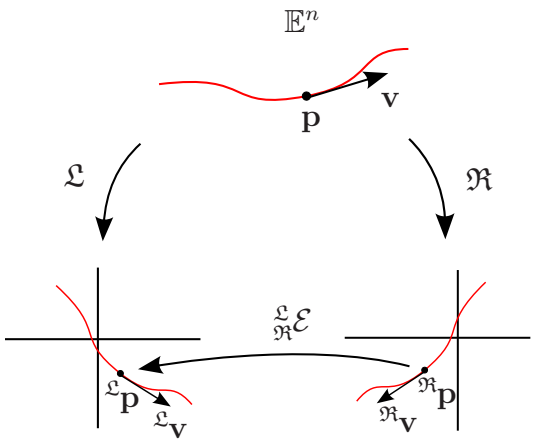


Figure 2.2.1: Representation of a coordinate change between two charts.

interesting to note that any coordinate change between two cartesian charts (thus isometries) may be described as an element of the Euclidean group $E(n)$ (see for example [5]).

The coordinate transformation of vectors is not so straight forward to describe since they must be thought of as tangent vectors to curves. Given a coordinate representation for a vector $\mathbf{v_p}$, let $c :]-\epsilon, \epsilon[\longrightarrow \mathbb{R}^3$ such that $c(0) = {}^{\mathfrak{R}}\mathbf{p}$ and $\frac{d}{dt}\big|_{t=0} c(t) = {}^{\mathfrak{R}}\mathbf{v_p}$. Notice that this is a coordinate parameterization for the curve. Differentiating this curve in the new coordinates results in the new representation of the tangent vector:

$$
\begin{aligned}
{}^{\mathfrak{L}}\mathbf{v_p} &= \frac{d}{dt}\bigg|_{t=0} \left({}^{\mathfrak{L}}_{\mathfrak{R}}\mathcal{E} \circ c(t)\right) \\
&= \sum_i \frac{\partial\, {}^{\mathfrak{L}}_{\mathfrak{R}}\mathcal{E}}{\partial_i}\bigg|_{\mathbf{p}} \frac{d}{dt}\bigg|_{t=0} c^i(t) \\
&= {}^{\mathfrak{L}}_{\mathfrak{R}}\mathcal{E}_* \, {}^{\mathfrak{R}}\mathbf{v_p}
\end{aligned}
$$

where ${}^{\mathfrak{L}}_{\mathfrak{R}}\mathcal{E}_*$ is the linear application represented as the Jacobian matrix of ${}^{\mathfrak{L}}_{\mathfrak{R}}\mathcal{E}$.

2.3 Camera Model

The cameras used obey a projection model characterized by a projection center ($\mathbf{p}_c$) and an image plane at a unit distance from $\mathbf{p}_c$. The camera projects points in the ambient space ($\mathbb{E}^3$) on the image plane (identified with $\mathbb{E}^2$) through the projection center. Given $\mathbf{p} \in \mathbb{E}^3$ and an orthonormal chart $\mathfrak{C}$ centered on the chosen projection center $\mathbf{p}_c$ with the image plane described by the equation $z = 1$ in this chart, then the projection function (which will be denoted by $\mathcal{P}_{\mathfrak{C}}$) written in coordinates is given by

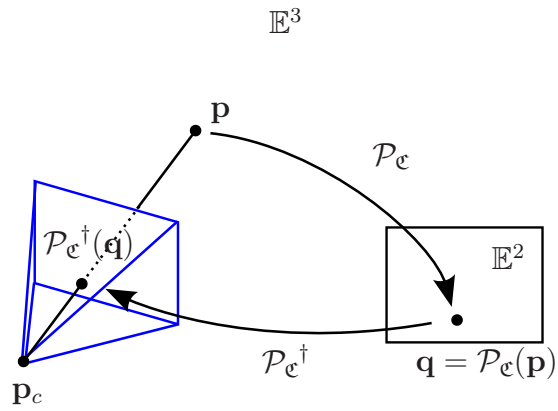


Figure 2.3.1: Representation of the projection function $\mathcal{P}_{\mathfrak{C}}$ and its pseudo-inverse $\mathcal{P}_{\mathfrak{C}}{}^{\dagger}$ which includes the projected point back on the image plane.

$$\begin{aligned}\mathcal{P}_{\mathfrak{C}} : \mathfrak{C}(\mathbb{E}^3) &\longrightarrow \mathfrak{p}(\mathbb{E}^2_{\mathfrak{C}}) \\ {}^{\mathfrak{C}}\mathbf{p} &\longmapsto \left(\frac{{}^{\mathfrak{C}}p^1}{{}^{\mathfrak{C}}p^3}, \frac{{}^{\mathfrak{C}}p^2}{{}^{\mathfrak{C}}p^3}\right)\end{aligned}$$

where $\mathfrak{p}$ is the chart mentioned in 2.2.2. Although this is the natural chart arising from the projection model, the camera's physical construction originates another chart ($\mathfrak{i}$), called the image chart, where actual pixels are measured. The coordinate change function between these charts is given by

$$\begin{aligned}{}^{\mathfrak{i}}_{\mathfrak{p}}\mathcal{E} : \mathbb{R}^2 &\longrightarrow \mathbb{R}^2 \\ {}^{\mathfrak{p}}\mathbf{q} &\longmapsto \left(f_x\ {}^{\mathfrak{p}}q^1 + c_x,\ f_y\ {}^{\mathfrak{p}}q^2 + c_y\right)\end{aligned}$$

where f_x, f_y, c_x e c_y are the camera's intrinsic parameters and are described in [8]. It is assumed that the image acquisition hardware does not introduce distortions. Since these can usually be corrected beforehand [10] [11] there is no loss in generality overlooking them here.

It is also usefull to consider a pseudo-inverse for the projection function (written as $\mathcal{P}_{\mathfrak{C}}{}^{\dagger}$), that includes a projected point $\mathbf{q} \in \mathbb{E}^2_{\mathfrak{C}}$ on the ambient space as $\mathbf{p} \in \mathbb{E}^3$ (inclusion of the image plane on the ambient space). In coordinates this operation is written as

$$\begin{aligned}\mathcal{P}_{\mathfrak{C}}{}^{\dagger} : \mathfrak{p}(\mathbb{E}^2_{\mathfrak{C}}) &\longrightarrow \mathfrak{C}(\mathbb{E}^3) \\ {}^{\mathfrak{p}}\mathbf{q} &\longmapsto \left({}^{\mathfrak{p}}q^1, {}^{\mathfrak{p}}q^2, 1\right)\end{aligned}$$

Note that these two functions are not inverses since $\mathcal{P}_{\mathfrak{C}}{}^{\dagger} \circ \mathcal{P}_{\mathfrak{C}} \neq Id$, although $\mathcal{P}_{\mathfrak{C}} \circ \mathcal{P}_{\mathfrak{C}}{}^{\dagger} = Id$. See figure 2.3.1 for a representation.

2.4 Projective Space

An alternate natural representation for describing a projection is known as the projective space, denoted $\mathbb{P}^n$. It is defined as the quotient space

$$\mathbb{P}^n \cong \frac{\mathbb{R}^{n+1} - \{0\}}{\sim}$$

where $\sim$ denotes the equivalence relation

$$a \sim b \Longleftrightarrow \exists \lambda \in \mathbb{R} - \{0\} : a = \lambda b$$

This construction characterizes $\mathbb{P}^2$ as the space of straight lines through the origin in $\mathbb{R}^3$. To denote a point in $\mathbb{P}^2$ (each one representing a straight line) a point $\mathbf{p} = (x, y, w) \in \mathbb{R}^3 - \{0\}$ is chosen as a representative of the equivalence class and written as

$$\bar{\mathbf{p}} = [x : y : w]$$

which represents a point in $\mathbb{P}^2$ with the interpretation of a line through the origin containing $\mathbf{p} \in \mathbb{R}^3$. For further insight see [8] and [9].

Remembering that given an orthonormal chart $\mathfrak{C}$ for $\mathbb{E}^3$ it can be used to project a scenery point $\mathbf{p} \in \mathbb{R}^3$ on the plane $z = 1$, this projected point can also be thought as the intersection of the line that passes through the origin and $\mathbf{p}$ with the plane $z = 1$. It is this interpretation that ties $\mathbb{P}^2$ with the images $\mathbb{E}^2$. Given an image point $\mathbf{q} \in \mathbb{E}^2$ and its representation on the projection chart ${}^{\mathfrak{p}}\mathbf{q} = (x, y) \in \mathbb{R}^2$ it can be naturally embedded in $\mathbb{P}^2$ as $\bar{\mathbf{q}} = [x : y : 1]$ (this operation will be refered to as inclusion $\iota : \mathbb{R}^n \to \mathbb{P}^n$). The inverse operation mapping $\bar{\mathbf{q}} = [x : y : w]$ in ${}^{\mathfrak{p}}\mathbf{q} = (\frac{x}{w}, \frac{y}{w})$ is also possible as long as $w \neq 0$. The similarities between these operations and the camera projections described in section 2.3 should be obvious and are evidenced next. If $\mathbf{p} \in \mathbb{E}^3$ and $\mathbf{q} \in \mathbb{E}^2_{\mathfrak{C}}$, then

$$[{}^{\mathfrak{p}}q^x : {}^{\mathfrak{p}}q^y : 1] \sim \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\bar{\mathcal{P}}} \cdot [{}^{\mathfrak{C}}p^x : {}^{\mathfrak{C}}p^y : {}^{\mathfrak{C}}p^z : 1]$$

$$[{}^{\mathfrak{C}}p^x : {}^{\mathfrak{C}}p^y : {}^{\mathfrak{C}}p^z : 1] \sim \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}}_{\bar{\mathcal{P}}^\dagger} \cdot [{}^{\mathfrak{p}}q^x : {}^{\mathfrak{p}}q^y : 1]$$

$$
\begin{tikzcd}
\mathbb{R}^{n+1} \arrow[r, bend left, "\hat{f}"'] \arrow[d, bend right, "\pi"'] & \mathbb{R}^{m+1} \arrow[d, bend left, "\pi"] \\
\mathbb{P}^n \arrow[r, dotted, "\bar{f}"'] & \mathbb{P}^m \arrow[d, bend left, "p"] \\
\mathbb{R}^n \arrow[u, bend left, "\iota"] \arrow[r, bend right, "f"'] & \mathbb{R}^m
\end{tikzcd}
$$

Figure 2.4.1: Projective space explanation. Note that when a point in $\mathbb{R}^n$ is embedded in $\mathbb{P}^n$ it is represented as an element of a fibre in $\mathbb{R}^{n+1}$. All computations are performed on this representation.

where matrix multiplication is done as described in the next paragraph. These maps in projective space are important in image processing applications since many common operations can be described linearly. This also allows for multiple operations to be concatenated in a single operation through matrix multiplication.

Figure 2.4.1 provides a representation of what happens when the projective space is used to apply a map $f : \mathbb{R}^n \to \mathbb{R}^m$ to points. First a point $\mathbf{p} \in \mathbb{R}^n$ is included in the projective space as $\bar{\mathbf{p}} \in \mathbb{P}^n$ through the function ι. Since $\mathbb{P}^n$ is a quotient space, its elements can be represented by choosing an element of the fiber in $\mathbb{R}^{n+1}$. Noting that for this map to be well defined it must map fibers of this equivalence class to fibers, inducing a function $\bar{f} : \mathbb{P}^n \to \mathbb{P}^n$. Then the point can once again be projected to $\mathbb{R}^n$ through a function $p : \mathbb{P}^n \to \mathbb{R}^n$. What happens is that $f = p \circ \bar{f} \circ \iota$.

This representation also provides an elegant description of image lines (note that these are lines on the image plane and not the straight lines in $\mathbb{R}^3$ through the origin discussed above) by considering the line equation

$$
ax + by + c = 0 \Longleftrightarrow \begin{bmatrix} a & b & c \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0
$$

Here a line is represented as a triplet (a, b, c) that also obeys the equivalence relation defined above so it can also be described as a point in $\mathbb{P}^2$ (this abstraction of interchanging the role of points and lines is known as duality). A point $\bar{\mathbf{p}} = [x : y : w] \in \mathbb{P}^2$ belongs to a line $\bar{\mathbf{k}} = [a : b : c] \in \mathbb{P}^2$ if and only if $ax + by + cw = 0$. For obvious reasons, this operation shall be denoted as $\bar{\mathbf{k}}^T \bar{\mathbf{p}} = ax + by + cw$.

Defining an homography f (also known as a projective transformation) as a one to one mapping between two images that:

- Maps collinear image points to collinear image points,
- Maps concurrent lines to concurrent lines,
- Preserves incidence.

It can be proved that every homography can be described as a linear mapping of homogenous coordinates as a $(n+1) \times (n+1)$ non-singular matrix $\mathbf{A}$ (the converse is easilly checked as well). This matrix is unique up to a scale factor.

Although the duallity of points and lines allows for a unique representation of the two entities, these are intrinsically different objects. This reflects, for example, in the way they are transformed. Given a point $\bar{\mathbf{p}} \in \mathbb{P}^2$ and a line $\bar{\mathbf{k}} \in \mathbb{P}^2$ these are mapped through an homography f described by matrix $\mathbf{A}$ as:

$$\bar{\mathbf{p}}' = \mathbf{A} \cdot \bar{\mathbf{p}}$$
$$\bar{\mathbf{k}}' = \mathbf{A}^{-T} \cdot \bar{\mathbf{k}}$$

where the dot represents matrix multiplication on the left, treating $\bar{\mathbf{p}} = [x : y : w]$ as a column vector $\begin{bmatrix} x & y & w \end{bmatrix}^T$.

Note that in the above discussion all interest has been given to $\mathbb{P}^2$. It is important to realize that the duallity that exists between image points and lines in $\mathbb{P}^2$ also exists between points and planes in $\mathbb{P}^3$ with the same transformation rule through a non-singular 4×4 matrix. A line in $\mathbb{P}^3$ is not as easilly described, but given two points $\mathbf{p}_1$ and $\mathbf{p}_2$ a straight line that passes through these two points can be described in the projective space parametrically as the set of projective points

$$L \sim \{\bar{\mathbf{p}}_1 + \lambda \bar{\mathbf{p}}_2 : \lambda \in \mathbb{R}\}$$

2.5 Stereo System

The considered stereo system consists in the simultaneous acquisition of 2 images (each one as described in 2.3) using two different projections associated with the charts $\mathfrak{L}$ and $\mathfrak{R}$, each centered at a different point. Thus, a point $\mathbf{p} \in \mathbb{E}^3$ shall be projected on 2 planes, through 2 different projection centers, each seen as a copy of $\mathbb{E}^2$. The fact that both images are observing the same scenery through this particular sensor originates the well known epipolar constraint (see for example [8]).

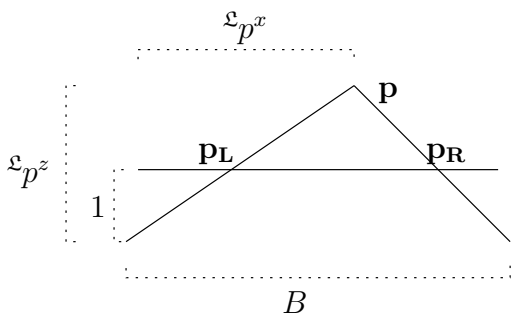


Figure 2.5.1: Explanation of the disparity coordinate chart.

Other than the already mentioned cartesian charts to describe points in $\mathbb{E}^3$, there is also another chart that arises naturally when $\mathfrak{L}$ and $\mathfrak{R}$ are considered to differ only by a horizontal translation (previous image stereo rectification relaxes this restriction so it can be used on real images). Suppose a point $\mathbf{p} \in \mathbb{E}^3$ is observed by the two cameras under these assumptions, resulting in the projections $\mathbf{p}_L \in \mathbb{E}^2_{\mathfrak{L}}$ and $\mathbf{p}_R \in \mathbb{E}^2_{\mathfrak{R}}$. Since these charts differ only by a horizontal translation ${}^{\mathfrak{p}}p^y_R = {}^{\mathfrak{p}}p^y_L$ (this is a special case of the known epipolar constraint). The x coordinate though differs on the two projections. This difference, known as disparity, can then be used in triangulation to solve for $\mathbf{p}$. This discussion hints at the possibility of using $({}^{\mathfrak{p}}p^x_L, {}^{\mathfrak{p}}p^y_L, {}^{\mathfrak{p}}p^x_R - {}^{\mathfrak{p}}p^x_L)$ as a coordinate chart for $\mathbb{E}^3$. Since the $\mathfrak{L}$ and $\mathfrak{R}$ differ only by a horizontal translation (see figure 2.5.1) the following relations hold in the $\mathfrak{p}$ chart:

$$\frac{^{\mathfrak{L}}p^x}{^{\mathfrak{L}}p^z} = \frac{^{\mathfrak{p}}p^x_L}{1}$$
$$\frac{^{\mathfrak{L}}p^x - B}{^{\mathfrak{L}}p^z} = \frac{^{\mathfrak{p}}p^x_R}{1}$$

where a similar system can be written for the second coordinate. Thus a possible coordinate change is

$$^{\mathfrak{D}'}p^1 = {}^{\mathfrak{p}}p^x_L = \frac{^{\mathfrak{L}}p^x}{^{\mathfrak{L}}p^z}$$
$$^{\mathfrak{D}'}p^2 = {}^{\mathfrak{p}}p^y_L = \frac{^{\mathfrak{L}}p^y}{^{\mathfrak{L}}p^z}$$
$$^{\mathfrak{D}'}p^3 = {}^{\mathfrak{p}}p^x_R - {}^{\mathfrak{p}}p^x_L = \frac{-B}{^{\mathfrak{L}}p^z}$$

This can be written as an homography as

$$[{}^{\mathfrak{D}'}p^1 : {}^{\mathfrak{D}'}p^2 : {}^{\mathfrak{D}'}p^3 : 1] \sim \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -B \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot [{}^{\mathfrak{L}}p^x : {}^{\mathfrak{L}}p^y : {}^{\mathfrak{L}}p^z : 1]$$

For most purposes in which this chart is to be used, it is more convenient to apply these maps directly to points on the image using chart $\mathfrak{i}$ instead. This is what will actually be called chart $\mathfrak{D}$, where $({}^{\mathfrak{D}}p^1, {}^{\mathfrak{D}}p^2, {}^{\mathfrak{D}}p^3) = ({}^{\mathfrak{i}}p^x_L, {}^{\mathfrak{i}}p^y_L, {}^{\mathfrak{i}}p^x_R - {}^{\mathfrak{i}}p^x_L)$. Then, ommiting details,

$$[{}^{\mathfrak{D}}p^1 : {}^{\mathfrak{D}}p^2 : {}^{\mathfrak{D}}p^3 : 1] \sim \begin{bmatrix} f_x & 0 & c^x_l & 0 \\ 0 & f_y & c^y_l & 0 \\ 0 & 0 & c^x_r - c^x_l & -Bf \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot [{}^{\mathfrak{L}}p^x : {}^{\mathfrak{L}}p^y : {}^{\mathfrak{L}}p^z : 1] \tag{2.5.1}$$

where f is the camera focal distance, (c^x_l, c^y_l) is the left camera's principal point and (c^x_r, c^y_r) is the right camera's principal point. Note that a previous image rectification process is needed to guarantee that the left and right focal distances are the same and that $c^y_L = c^y_R$.

Please note that this chart is not global (a point $\mathbf{p} \in \mathbb{E}^3$ with ${}^{\mathfrak{L}}p^z = 0$ is not representable and a disparity of 0 represents a point at infinity) and is not cartesian (implying, for example, that the dot product and cross product are not the usual ones). As such, unless certain care is taken its use is only recommended as an intermediate step.

Chapter 3

Capturing Depth With a Stereo System: Standard Algorithms

This chapter describes the necessary algorithms to perform a standard stereo reconstruction (in the absence of an interface). The process consists of several steps, mentioned here and described in the next pages:

- Image acquisition consists of acquiring the stereo pair to a computer representable form.
- Image rectification eliminates distortion introduced by the image acquisition hardware and treats the image so that epipolar lines are horizontal and on the same scanline on both cameras.
- Matching of features of both images by fotometric and/or geometric constraints.
- Reconstruction, where the matched features are triangulated to infer depth.

Although not included in the previous list, image rectification requires a one-time camera calibration step which completely describes the camera geometry.

3.1 Image Acquisition

The first step in any stereo reconstruction process is image acquisition. Since calibrated stereo is used, a means to fix two image acquisition devices in space is needed. This can be accomplished in different ways, the most common being a rigid bar on which two cameras are screwed tight. An alternative is to use a beam splitter enabling a single camera to acquire both images. The only drawback of the later approach is that only half the resolution of the camera is available. Figure 3.1.1 illustrates both approaches.

Figure 3.1.1: Example of image aquisition hardware. On the left an example of two cameras mounted on a horizontal bar, on the right an example of a beam splitter to be mounted on a single camera.

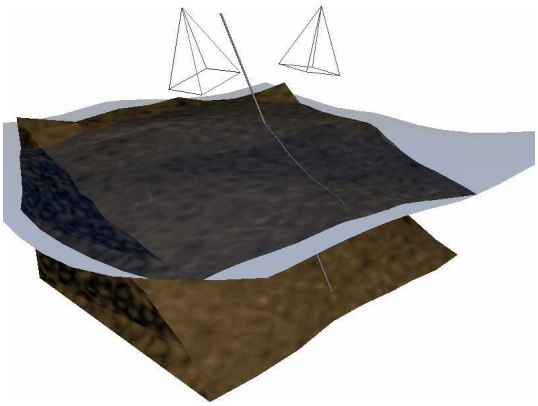


Figure 3.1.2: Example of a computer generated image of a submerged scenery, illustrating the distortion introduced by refraction. Notice how the inserted rod seems to bend once it penetrates the interface. Images such as these can be generated of arbitrary exactly known scenery so error measures can be taken.

Reference is made to a program used to render synthetic images from a generated scenery of which all parameters are known. These images are usefull since they allow for measurement of reconstruction errors not possible with real images since exact position is usually unknown. The chosen program was `POV-Ray` since it models refraction correctly and is one of the oldest of its kind still in use today (which means it has been extensively tested). Its free availability also played its part in the decision process. Unfortunately it suffers from a relatively steep learning curve. To the rescue come third party graphical interfaces which ease the user through the process of creating a scene. For an example of a rendered image, see figure 3.1.2.

3.2 Calibration and Image Rectification

Camera calibration plays a crucial role in stereo systems. It not only simplifies the matching process by infering the geometry between cameras, but it also fixes the metric of the world. For this task, Jean-Yves Bouguet's *Camera Calibration Toolbox for Matlab* is used. The toolbox is freely available and allows for intrinsic and extrinsic camera calibration using a calibration rig similar to a chess board. The work is based on Zhang [10] and Heikkila [11]. Since the camera's position relative to the calibration rig is also obtained, a chart with the interface at $z = 0$ is easily calibrated by acquiring a pair of images with the rig floating on the interface. Although the toolbox also performs standard stereo image rectification, an alternate implementation was developed, with much faster performance and with the additional Snell correction builtin (which will be described in chapter 4). An in depth description of the conventional calibration procedure can be found in [9].

Standard image rectification without Snell rectification which will be described in the appropriate chapter, is implemented in 3 steps:

1. Each pixel commonly described in the image chart ($\mathfrak{i}_1$) is first converted to the natural projection chart through ${}^{\mathfrak{p}}_{\mathfrak{i}_1}\mathcal{E}$.

2. Once on this plane it is possible to compensate non-linear distortions introduced by the camera acquisition hardware. This distortion is mainly radial in nature, characterized through the even powers of a polynomial in $r = \sqrt{({}^{\mathfrak{p}}p^1)^2 + ({}^{\mathfrak{p}}p^2)^2}$. Tangential distortion is also usually corrected.

3. If an extrinsic correction is necessary (a change in the desired projection plane with the projection center fixed) it is possible to do so at this point. These are usually implemented as an homography between two projective spaces and are usually applied in stereo setups to make epipolar lines horizontal.

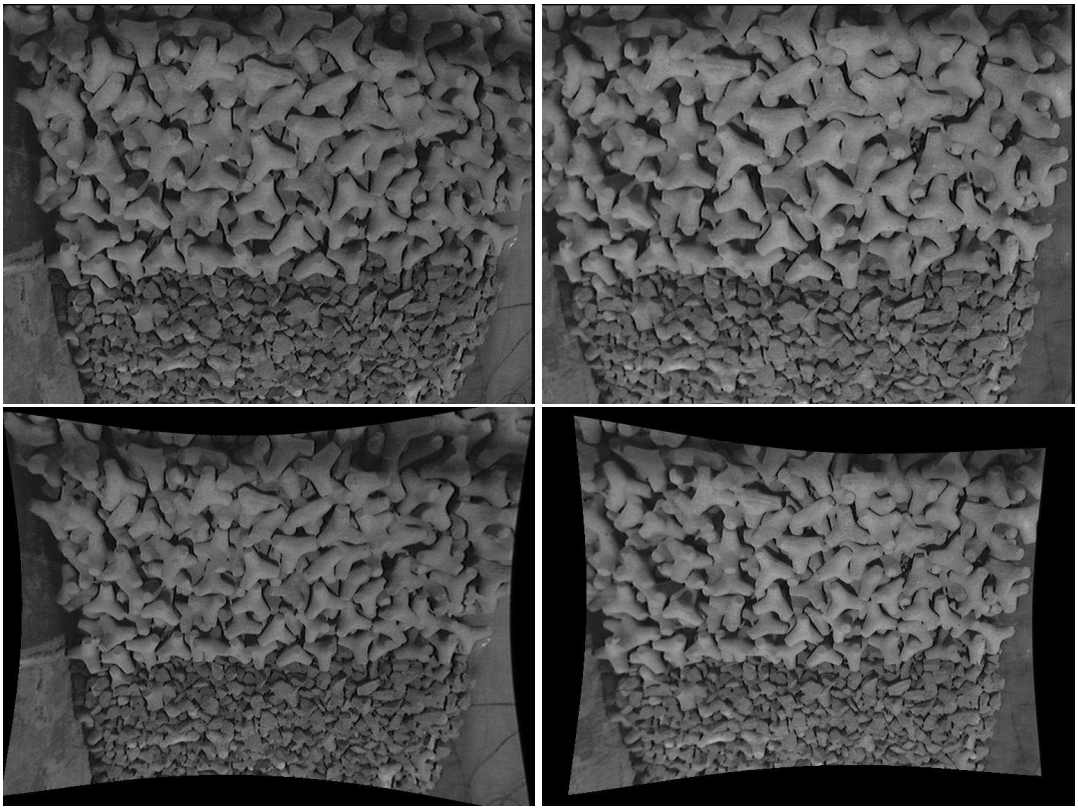


Figure 3.2.1: Image rectification results. On top the left and right original images are presented and on the bottom the corresponding rectified images with horizontal epipolar lines. Notice the high radial distortion that was corrected.

4. It is then possible to choose desired intrinsic parameters for the new "desired camera" and change back to an image chart using ${}^{\mathfrak{i}_2}_{\mathfrak{p}}\mathcal{E}$. These new intrinsic parameters are usually chosen so as to minimize information loss contained in the image.

Since what is commonly needed is for every pixel in the rectified image to have a brightness value set, the whole rectification procedure is usually run backwards. So for every pixel in the desired rectified image the steps described above are run in the inverse order, applying the inverse operation in each step. This results on the color value being set by the correct pixel on the original image. Note that under normal operating conditions every step is invertible (an exception occurs in the third step, where it is possible for the whole image to collapse on a line for certain extrinsic transformations not usually encountered). An example of the results obtained is presented in figure 3.2.1.

3.3 Stereo Matching Algorithms

Two distinct algorithms were implemented to solve the correspondence problem. The problem consists in assigning a correspondence of features on the right image with features on the left image. Two categories of such algorithms exist, based on what is considered a feature that needs matching. If a correspondence is attempted for every pixel on one of the images then it is a dense correspondence algorithm. If, on the other hand, correspondence is only attempted on previously detected features (such as corners or lines present in the image) then it is a sparse correspondence algorithm. An implementation of each was tried, but only the dense correspondence algorithm proved usefull.

3.3.1 Sparse Stereo

Although of limited use for these particular problems (a dense stereo algorithm is needed for interface estimation), the sparse correspondence algorithm described in [12] was implemented. It uses correlation (or any other cost function) between features to find the permutation matrix that maximizes the global gain.

If the intensity values (or, more appropriately the zero mean normalized intensity) of $N \times N$ windows centered at some detected feature locations on the left and right images are stacked on the lines of two matrices $\mathbf{F}_L$ and $\mathbf{F}_R$ respectively, the correlation of all these features is found by computing $\mathbf{C} = \mathbf{F}_L \mathbf{F}_R^T$. The correspondence problem then resumes itself to finding the partial permutation matrix $\mathbf{P}$ that solves

$$\begin{aligned} \mathbf{P}^* = \arg\max_{\mathbf{P}} \quad & \text{trace}(\mathbf{P}\mathbf{F}_L\mathbf{F}_R^T) \\ s.t. \quad & \mathbf{P} \in \mathcal{P}_P(p_L, p_R, K) \end{aligned} \tag{3.3.1}$$

where $\mathcal{P}_P(p_L, p_R, K)$ denotes the set of partial permutation matrices of size $p_L \times p_R$ (p_L and p_R are the number of features on each image) with K correspondences. A partial permutation matrix is a permutation matrix that allows for some of its columns or lines to be zero. To avoid many false matches, it is imposed that the $\mathbf{P}$ matrix has to have rank K so that only the K strongest matches are allowed. For example K might be $\min(p_L, p_R)/2$.

Other constraints can be added through the use of a support matrix $\mathbf{S}$ that indicates which matches are valid. This way it is possible to reject correspondences which are known from the start not to be feasible due to, for example, epipolar constraint or minimum/maximum allowed disparity. This reduces the search space considerably, increasing the algorithm's performance and also prevents possible false matches that could otherwise occur.

The chosen features for this problem are corners, using the well known Harris corner detector [14]. Its choice was based on the structure of the pretended scenery (a pile of rocks with

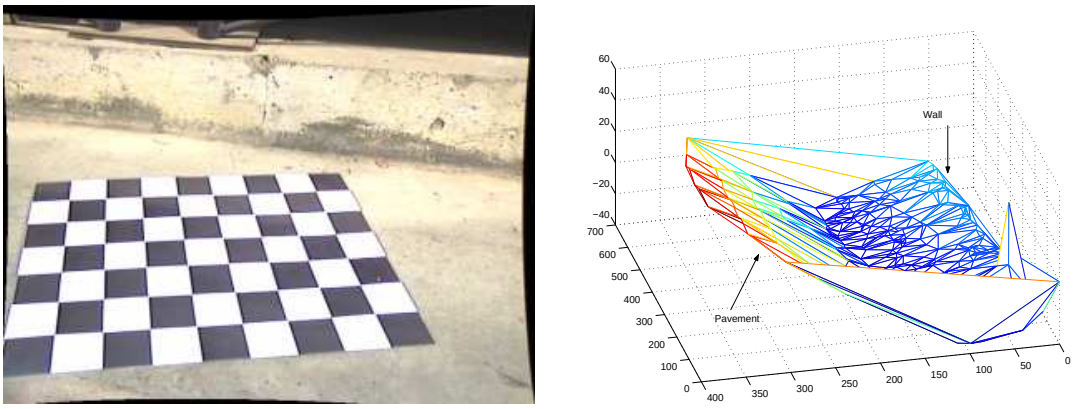


Figure 3.3.1: Sparse stereo matching results. Left: the previously rectified left image of a stereo pair. Right: Computer reconstruction of the observed scene where a triangulation algorithm was applied. The sparse stereo algorithm described in [12] was used. The units on the axis are pixels (disparity space).

sharp corners).

Problem 3.3.1 is solved using the well known simplex method for linear optimization problems and an implementation by Michel Berkelaar (`lp_solve`) was used. Figure 3.3.1 provides an example of the results obtained.

3.3.2 Dense Stereo

Since the surface estimation algorithm requires dense stereo maps to be available, Sun's [13] algorithm was used. It consists of two dynamic programming steps in order to find the maximum surface $\mathcal{S}$, on a 3 dimensional space, that minimizes $\sum_{(x,y,d)\in\mathcal{S}} C(x,y,d)$, where $C(\cdot)$ defines a distance measure, for example the symmetric value of the normalized cross correlation of a window centered at (x,y) on the first image with a window of the same size centered at $(x+d,y)$ on the second. Although the complexity of the algorithm is $O(MND)$ where (M,N) is the image size and D is the maximum allowed disparity, the use of sub-regions with multi-resolution techniques and the fact that it was implemented in C (with a Matlab interface) makes the algorithm efficient in terms of speed, taking a few seconds to run on video frames (see [13] for details). An example of the output is presented in figure 3.3.2. Although it is not evident in this case, due to the use of correlation the algorithm does not fare well in regions without clearly defined features and in the presence of occlusion.

Even though the algorithm requires the images to have been perfectly rectified (which in the presence of an interface is not guaranteed), practice shows that acceptable results are still obtained in the presence of slight deviations (1 or 2 pixels) as shown in figure 3.3.3. Obviously,

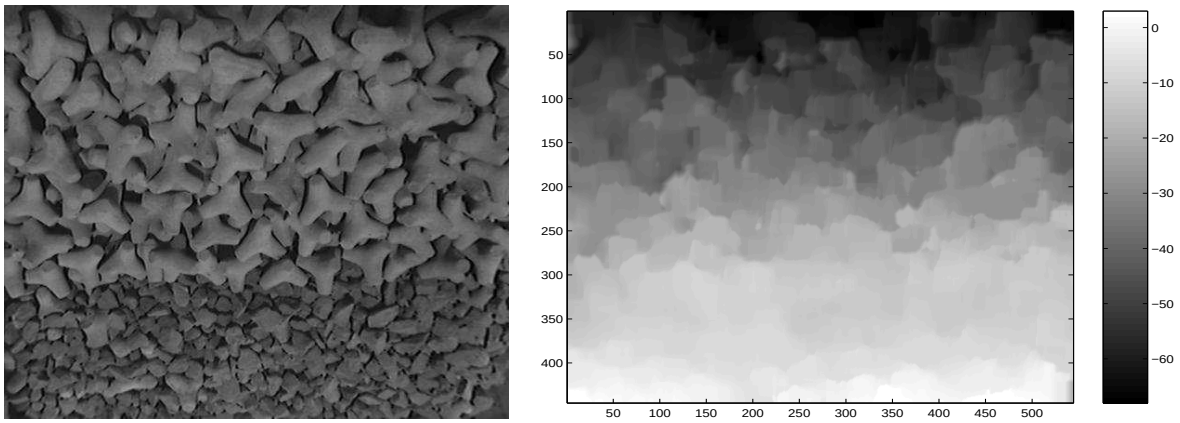


Figure 3.3.2: Dense stereo matching results. Left: previously rectified left image of a stereo pair. Right: dense disparity map obtained by the algorithm described in [13] (The scale on the right allows a conversion of the grayscale levels to disparity numerical values.

with the intent of precision, exact matching is of utmost importance.

Another dense matching algorithm, described by Kolmogorov in [15], was tested but it turned out to be significantly slower without any significantly visible improvements. Although its implementation allows for two dimensional matching (allowing two dimensional disparity maps to be obtained) as necessary for interface estimation, its use is not practical since it takes many hours to run (an attempt was aborted after a few hours).

3.3.3 Two Dimensional Dense Matching

Given the distortion introduced by the interface, it is necessary to obtain disparity maps not only along the expected epipolar direction, but also on the surrounding area. The presented methods are too simple for real-world applications. Their usefullness lies only in providing the necessary disparity maps to test the algorithms developed. No interest is given to their robustness or performance on other images besides the ones presented.

A first idea is to iterate the standard algorithm in a cyclic manner on the two dimensions. The disparity maps indicate which pixel on the right image best matches an given pixel on the left image. Once a map has been obtained an approximation of the left image can be constructed using the color information of the right image using the pull-back of the disparity function on the pixels of the right image. Let L and R denote the matrices containing the left and right intensity images of a stereo pair. If D represents a disparity map the notation D^*R will be used to represent the pull-back of the image R through the disparity map D.

The used algorithm works as follows (figure 3.3.4 represents the steps)

1. Two images are provided and the algorithm is run along the epipolar direction obtaining a

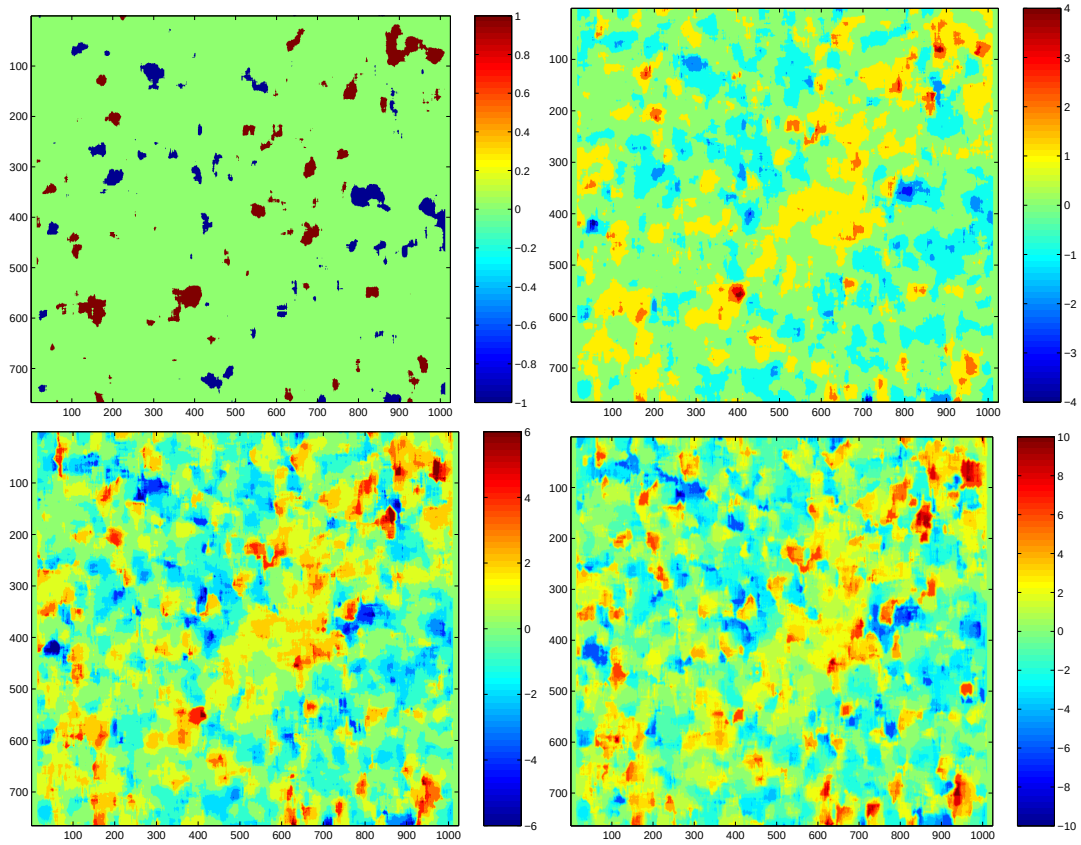


Figure 3.3.3: Sensitivity of the dense matching algorithm in the presence of rectification errors. The images present the disparity map obtained when matching an image to itself translated by n pixels in the vertical direction. Top right: 1 pixel; top left: 2 pixels; bottom left: 3 pixels; bottom right: 4 pixels.

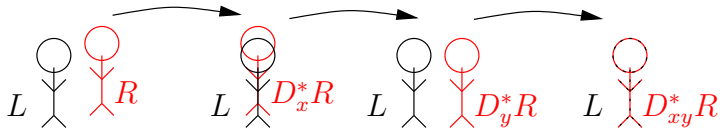


Figure 3.3.4: Illustration of the cyclic algorithm for dense stereo matching in 2 dimensions. First an estimate of the disparity along the principal direction is obtained, being used to obtain the disparity map along the other direction (this one must be close to 0 for the first map to have meaning). This new map is then used to recalculate the disparity along the principal direction.

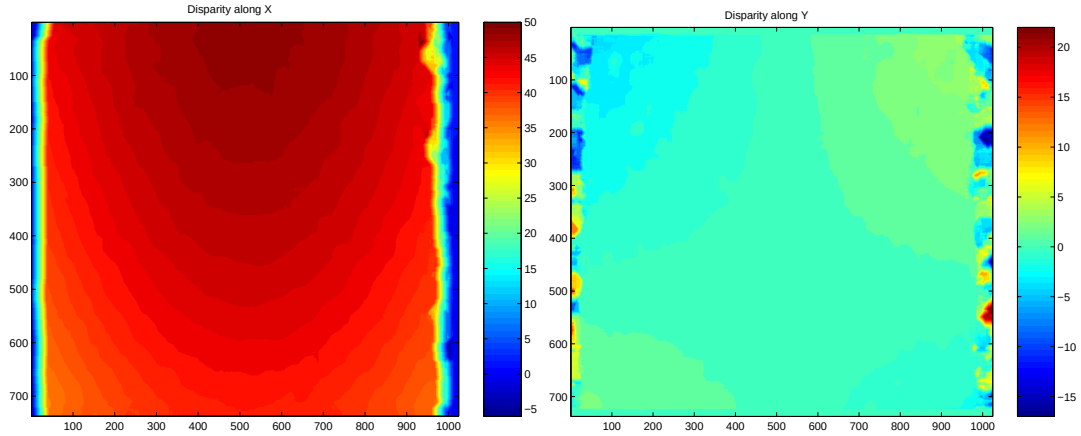


Figure 3.3.5: Disparity maps obtained using the cyclic algorithm applied to a submerged plane at a depth of $1.5m$. The cameras were at about $1.3m$ above the interface. As indicated, on the left the disparity map along the principal direction is shown, on the right the disparity map along the other direction.

disparity map D_x. It's assumed that the disparity along the other direction is sufficiently small so the algorithm can still lock onto the desired disparity.

2. The right image is pulled back and the algorithm is run with the resulting image (hopefully already aligned along the epipolar direction) and the original left image along the direction orthogonal to the epipolar. This results in a D_y disparity map.

3. The previous steps can be iterated (now using $D_y^* R$ as a starting image). The final D_x and D_y disparity maps are the output of the algorithm. Figure 3.3.5 presents the results obtained when using this technique on a submerged plane.

To estimate the interface it will also be necessary to match images where both disparities assume high values. The previous algorithm will not work in these situations. Brute force is applied in these cases, searching on a region of the right image something resembling a given feature on the left. If $C(u, v, du, dv)$ denotes a cost measure (the symmetric of the zero mean normalized cross correlation for example) of matching a window of a given size on the left image centered at (u, v) with a window of the same size on the right image centered at $(u+du, v+dv)$, the disparity maps without any smoothness constraints are obtained by solving

$$\begin{aligned} (du^*, dv^*)_{uv} = \arg \max_{(du,dv)} \quad & C(u, v, du, dv, n) \\ s.t. \quad & du \in I_u \\ & dv \in I_v \end{aligned} \tag{3.3.2}$$

where I_u e I_v are the sets of admissible values for du e dv respectively. For an image of

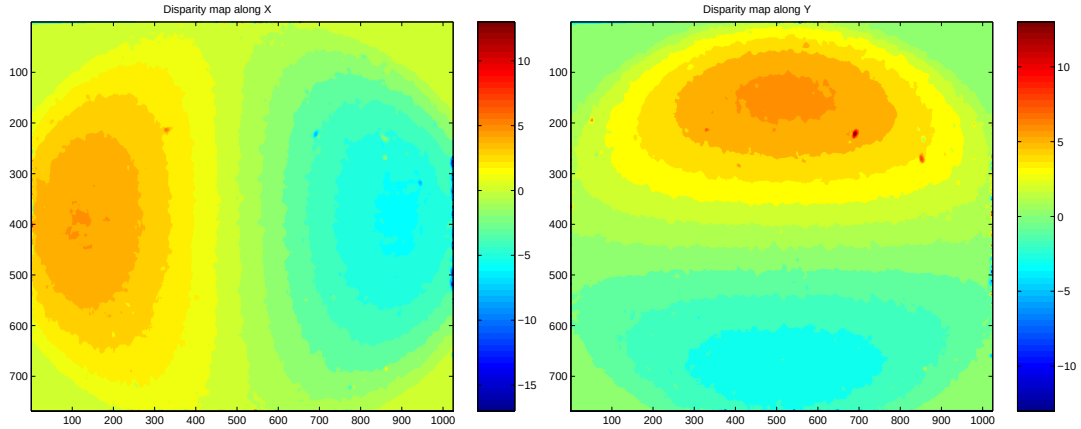


Figure 3.3.6: Disparity maps obtained with exhaustive search applied to a water bubble with $1dm$ of thickness in the middle. The bubble is on a plane at a distance of about $2.8m$ from the cameras. The disparity map along the principal direction is shown, on the right the map along the other direction.

dimension MxN, MN distinct optimization problems need to be solved, each with complexity proportional in both the number of admissible values in I_u and in I_v. To aid the search multi-resolution techniques are implemented, starting with scaled versions of the images and propagating the results (and possible errors) through the scale pyramid up to the actual sized images.

An example of the results obtained is provided in figure 3.3.6.

3.4 Reconstruction

Once a disparity map has been obtained, it is a description of the scenery on the $\mathfrak{D}$ chart. All that needs to be done is to convert it to a more suitable coordinate chart, such as $\mathfrak{W}$. The easiest way to accomplish this is through the projective transformation described by equation 2.5.1. This step will also need a correction when in the presence of an interface, but this discussion will be omitted in this chapter.

Chapter 4

Submerged Scenery Reconstruction

This chapter focuses on the reconstruction of submerged scenes (in the presence of an interface between the sensor and the scenery) using stereo image pairs. In these conditions, the non-linearity characterized by Snell's law distorts the acquired images breaking the geometric constraints usually exploited for reconstruction. In particular, the epipolar constraint is no longer valid, greatly hampering the feasibility of feature matching. The objective will be to study the nature of the distortion and reduce it (it is not possible to remove it completely) so that normal stereo matching algorithms can be used under certain conditions. It is assumed that the interface is planar (in a static configuration) and its location is known on a cartesian chart.

4.1 Snell's Law

Let $\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2 \in \mathbf{T_p}\mathbb{E}^3$ be two vectors (incident and refracted) with unit norm at a point $\mathbf{p} \in \mathbb{E}^3$ on the interface and let $\hat{\mathbf{u}} \in \mathbf{T_p}\mathbb{E}^3$ be a unit norm vector at the same point, orthogonal to the interface's surface (figure 4.1.1 illustrates these). Snell's law [16] [17] [18] relates these three vectors through the equation

$$k_1(\hat{\mathbf{v}}_1 \times \hat{\mathbf{u}}) = k_2(\hat{\mathbf{v}}_2 \times \hat{\mathbf{u}})$$

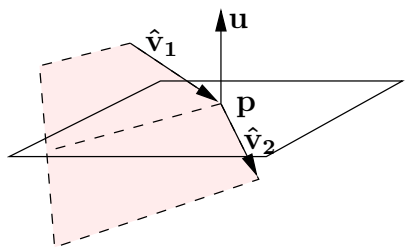


Figure 4.1.1: Snell law in 3 dimensions.

where $k_1, k_2 \in \mathbb{R}$ are each media's refractive index. Note that the cross product is an intrinsic operation so it does not matter on which coordinate chart it is performed (as long as it is correctly described in it). For the equation to be valid, $\hat{\mathbf{u}}$ does not necessarily have to have unit norm and the norms of $\hat{\mathbf{v}}_1$ and $\hat{\mathbf{v}}_2$ only have to be equal, not necessarily 1. So the former can be relaxed and written as the system

$$(k_1\mathbf{v}_1 - k_2\mathbf{v}_2) \times \mathbf{u} = 0$$
$$\|\mathbf{v}_1\| = \|\mathbf{v}_2\|$$

The first equation clearly states that $k_1\mathbf{v}_1 - k_2\mathbf{v}_2$ has to be collinear with $\mathbf{u}$, so there'll be a $\gamma \in \mathbb{R}$ such that:

$$k_1\mathbf{v}_1 - k_2\mathbf{v}_2 = \gamma\mathbf{u}$$

Since the interface is assumed to be planar and known, there is a cartesian chart $\mathfrak{W}$ where ${}^{\mathfrak{W}}\mathbf{u} = (0, 0, 1)$. In this chart the following holds:

$${}^{\mathfrak{W}}v_2^x = \frac{k_1}{k_2}{}^{\mathfrak{W}}v_1^x \tag{4.1.1}$$

$${}^{\mathfrak{W}}v_2^y = \frac{k_1}{k_2}{}^{\mathfrak{W}}v_1^y \tag{4.1.2}$$

Since $\|\mathbf{v}_1\| = \|\mathbf{v}_2\|$, it follows that

$${}^{\mathfrak{W}}v_2^z = \sqrt{\left(1 - \left(\frac{k_1}{k_2}\right)^2\right)\left(({}^{\mathfrak{W}}v_1^x)^2 + ({}^{\mathfrak{W}}v_1^y)^2\right) + ({}^{\mathfrak{W}}v_1^z)^2} \tag{4.1.3}$$

Henceforth k_1/k_2 will be denoted as k.

4.2 First Order Approximation

Due to the complexity of the former expression, the exact result is approximated by its expansion in Taylor series, of which only the terms up to first order are retained. So the approximation around a point $\mathbf{a}$ is given by the expression

$$f(\mathbf{p}) \approx f(\mathbf{a}) + df(\mathbf{a}) \cdot (\mathbf{p} - \mathbf{a}), \quad \forall f : \mathbb{R}^N \to \mathbb{R} \text{ analytic}$$

where $df(\mathbf{a})$ is the linear application described by the Jacobian of f (written in a coordinate chart) at the point $\mathbf{a}$. The obvious choice is to linearize around the vertical direction, so this is

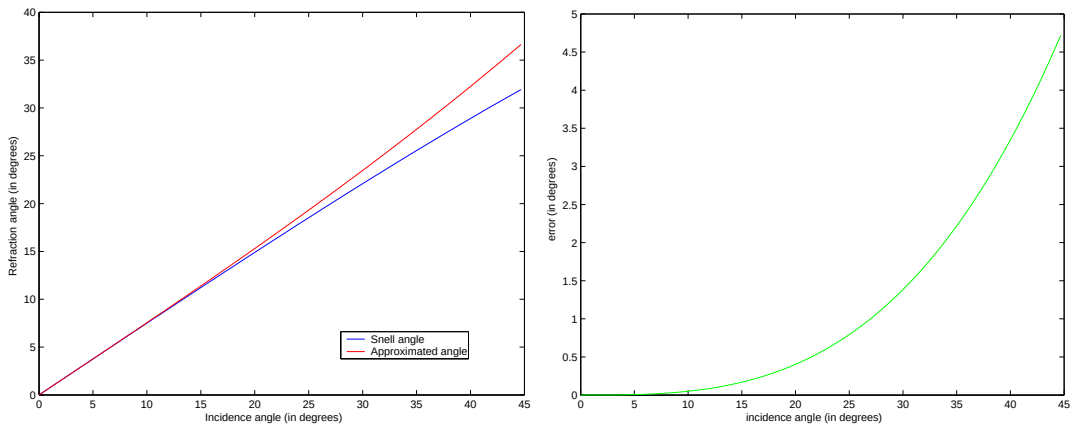


Figure 4.2.1: Comparison between the refracted angle when using a first order approximation and the angle given by Snell's law. The interface considered is air/water, with refraction index $k = 1/1.33$. All scales are in meters.

what will be done. Linearizing ${}^{\mathfrak{W}}v_2^z$ around the vector ${}^{\mathfrak{W}}\mathbf{v}_a = (0, 0, -1)$. Thus, dropping the chart notation in favor of easier reading (all coordinate operations refer to the $\mathfrak{W}$ chart):

$$
\begin{aligned}
dv_2^z(\mathbf{v})|_{\mathbf{v}=\mathbf{v}_a} &= \left.\begin{bmatrix} \frac{(k^2-1)v^x}{\sqrt{(1-k^2)\left((v^x)^2+(v^y)^2\right)+(v^z)^2}} \\ \frac{(k^2-1)v^y}{\sqrt{(1-k^2)\left((v^x)^2+(v^y)^2\right)+(v^z)^2}} \\ \frac{-v^z}{\sqrt{(1-k^2)\left((v^x)^2+(v^y)^2\right)+(v^z)^2}} \end{bmatrix}^T\right|_{\mathbf{v}=\mathbf{v}_a} \\
&= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}
\end{aligned}
$$

So the first order Taylor series approximation is

$$
\begin{aligned}
v_2^z(\mathbf{v}) &\approx v_2^z(\mathbf{v}_a) + df(\mathbf{v}_a) \cdot (\mathbf{v} - \mathbf{v}_a) \\
&= -1 + \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \cdot \left(\begin{bmatrix} v^x \\ v^y \\ v^z \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \right) = v^z
\end{aligned}
$$

This results in

$$
\mathbf{v_2} \approx \begin{bmatrix} kv_1^x \\ kv_1^y \\ v_1^z \end{bmatrix} = \begin{bmatrix} k & 0 & 0 \\ 0 & k & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \mathbf{v_1}
$$

The approximation error is presented in figure 4.2.1. It is seen to be less than half a degree for incidence angles up to 20 degrees. In practice, for stereo reconstruction, this approximation works well for incidence angles of up to about 15 degrees.

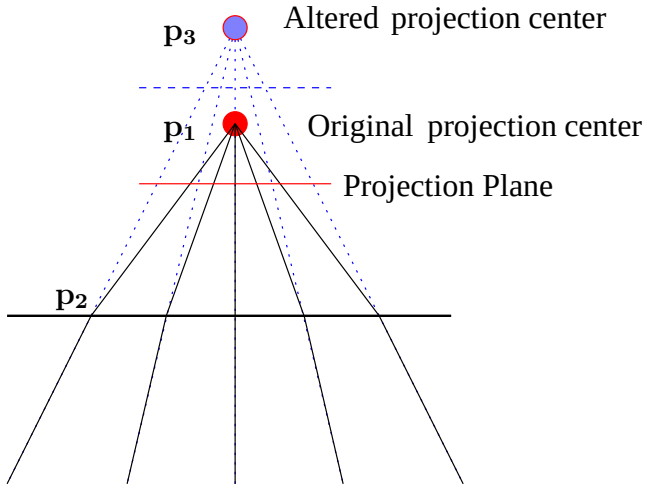


Figure 4.2.2: This illustrates the interpretation of the first order approximation of Snell's law. The path of various light beams are drawn and bended by the interface. If the bended light rays are extendend back to the original media, they all converge at a point allowing for a virtual camera to be placed at that location.

4.2.1 Geometric Interpretation

Let's assume an orthonormal chart $\mathfrak{W}$ calibrated so that the interface plane satisfies the equation $z = 0$. All computations here are done in the $\mathfrak{W}$ chart, but to simplify notation the chart will be dropped from the notation. There's a camera observing the scenery with projection center at a point $\mathbf{p}_1 \in \mathbb{E}^3$.

Following a beam of light that leaves $\mathbf{p}_1$ in a given direction $\mathbf{v}_1$, it will hit the interface at $\mathbf{p}_2$, which in $\mathfrak{W}$ coordinates may be written as

$$\mathbf{p}_2 = \mathbf{p}_1 - \frac{p_1^z}{v_1^z}\mathbf{v}_1$$

Accordingly to the approximation mentioned previously, there the beam of light shall be refracted and change its direction to $\mathbf{v}_2 = (kv_1^x, kv_1^y, v_1^z)$, departing from $\mathbf{p}_2$. Intersecting this straight line with the straight line $l = \{(x, y, z) \in \mathbb{R}^3 : x = p_1^x, y = p_1^y\}$

$$\begin{aligned} &\mathbf{p}_2 + t\mathbf{v}_2 = (p_1^x, p_1^y, \cdot) \\ \Longleftrightarrow &t = \frac{p_1^z}{v_1^z k} \end{aligned}$$

a new point $\mathbf{p}_3$ is obtained:

$$\mathbf{p}_3 = \mathbf{p}_2 + \frac{p_1^z}{v_1^z k}\mathbf{v}_2 = (p_1^x, p_1^y, \frac{p_1^z}{k}) \tag{4.2.1}$$

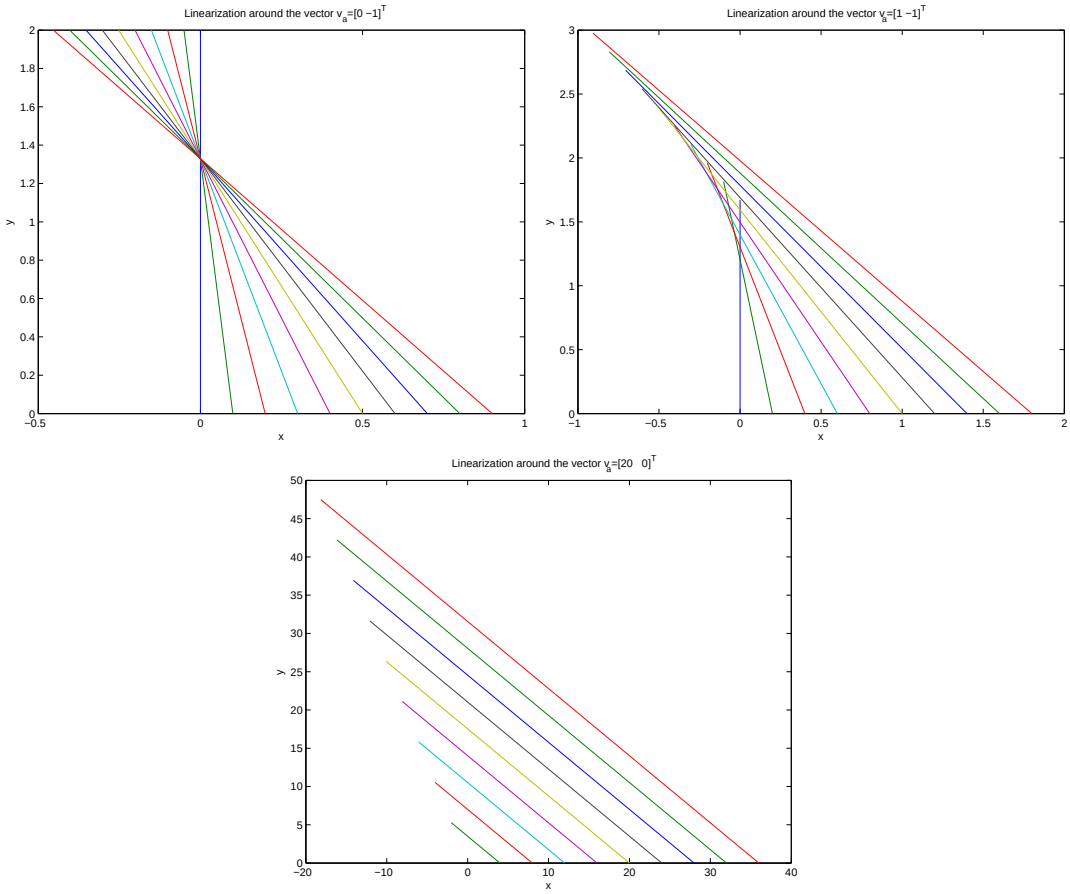


Figure 4.2.3: First order approximation of Snell's law around an angle perpendicular to an interface (on the left) placed at $z = 0m$ and around an angle of $\pi/4$ (on the right). Below a linearization around an angle close to $\pi/2$. The camera was placed at $1m$ above the interface.

This shows that $\mathbf{p}_3$ is independent of the initial direction $\mathbf{v}_1$ and is illustrated in figure 4.2.2. It means that the interface's distortion can be compensated considering a virtual projection center at $\mathbf{p}_3$ (as long as the first order Taylor approximation is considered valid). The camera's orientation does not need to be altered, although it normally is in the image rectification step when considering stereo pairs of images so as to make the epipolar lines horizontal.

Note that the first order approximation was done around an angle perpendicular to the interface. Figure 4.2.3 illustrates that although this allows to think of a virtual projective camera with a different projection center, the same is not possible when linearizing around other angles. There is a notable exception that occurs when linearizing around an angle of $\pi/2$. In this case it is possible to think not of a projective virtual camera as before, but as an orthographic virtual camera. Unfortunatly, due to physical phenomena of refractive atenuation and the increase of reflection when the incidence angle is high, this solution has no practical interest.

4.2.2 Correction Homography

Since the refracted rays and the approximated rays coincide at the interface plane, image correction consists in projecting the image points onto the interface plane, calculating the new camera parameters and re-projecting on the desired virtual camera. Suppose a point $\bar{\mathbf{p}}_0 \in \mathbb{P}^3$ is to be projected on a plane $\bar{\mathbf{k}}$ using $\bar{\mathbf{p}}_1 \in \mathbb{P}^3$ as a projection center. Consider the straight line through these two points

$$r = \{\bar{\mathbf{p}}_0 + \lambda \bar{\mathbf{p}}_1 : \ \lambda \in \mathbb{R}\}$$

and its intersection with the plane $\bar{\mathbf{k}} \in \mathbb{P}^3$ by finding λ_* that satisfies:

$$\bar{\mathbf{k}}^T \left(\bar{\mathbf{p}}_0 + \lambda_* \bar{\mathbf{p}}_1\right) = 0$$
$$\Leftrightarrow \lambda_* = -\frac{\bar{\mathbf{k}}^T \bar{\mathbf{p}}_0}{\bar{\mathbf{k}}^T \bar{\mathbf{p}}_1}$$

The projected point $\bar{\mathbf{p}}_2 \in \mathbb{P}^3$ will then be

$$\bar{\mathbf{p}}_2 \sim \bar{\mathbf{p}}_0 - \frac{\bar{\mathbf{k}}^T \bar{\mathbf{p}}_0}{\bar{\mathbf{k}}^T \bar{\mathbf{p}}_1} \bar{\mathbf{p}}_1$$

Since $\bar{\mathbf{k}}^T \bar{\mathbf{p}}_1 \neq 0$ (the projection center does not belong to the plane), multiply the former by this value using the equivalence relation, resulting in

$$\bar{\mathbf{p}}_2 \sim (\bar{\mathbf{k}}^T \bar{\mathbf{p}}_1)\bar{\mathbf{p}}_0 - \bar{\mathbf{p}}_1(\bar{\mathbf{k}}^T \bar{\mathbf{p}}_0)$$
$$\bar{\mathbf{p}}_2 \sim \underbrace{\left(\bar{\mathbf{k}}^T \bar{\mathbf{p}}_1 \mathbf{I} - \bar{\mathbf{p}}_1 \bar{\mathbf{k}}^T\right)}_{\mathbf{M}(\bar{\mathbf{p}}_1, \bar{\mathbf{k}})} \bar{\mathbf{p}}_0$$

where $\mathbf{I}$ denotes the identity matrix and $\mathbf{M}(\bar{\mathbf{p}}_1, \bar{\mathbf{k}})$ is the projection matrix parameterized by the projection center and the projection plane. In particular, if the plane $z = 0$ is considered ($\bar{\mathbf{k}} = [0 : 0 : 1 : 0]$), the matrix takes the form

$$\bar{\mathbf{p}}_2 \sim \underbrace{\begin{bmatrix} p_1^z & 0 & -p_1^x & 0 \\ 0 & p_1^z & -p_1^y & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & p_1^z \end{bmatrix}}_{\mathbf{M}_k(\bar{\mathbf{p}}_1)} \bar{\mathbf{p}}_0 \tag{4.2.2}$$

So it is possible to correct the images (described in the $\mathfrak{p}$ chart) by applying the homography characterized by

$$\mathbf{H} = \bar{\mathcal{P}} \, {}^{\mathfrak{C}}_{\mathfrak{W}}\bar{\mathcal{E}} \, \mathbf{M}(\bar{\mathbf{p}}_1, \bar{\mathbf{k}}) \, {}^{\mathfrak{W}}_{\mathfrak{C}}\bar{\mathcal{E}} \, \bar{\mathcal{P}}^{\dagger} \tag{4.2.3}$$

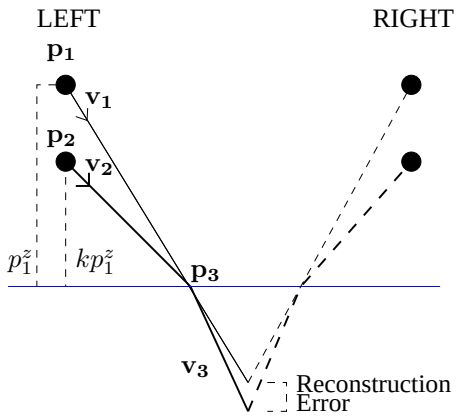


Figure 4.3.1: Illustration of the reconstruction error using Snell correction. Note that the exact trajectory followed by a light beam under the interface is along $\mathbf{v}_3$ and not along $\mathbf{v}_1$.

where $\bar{\mathcal{P}}$ and $\bar{\mathcal{P}}$ are the projection and pseudo-inverse homography of the camera as described in section 2.4. ${}^{\mathfrak{W}}_{\mathfrak{C}}\bar{\mathcal{E}}$ is the homography defining the camera to world coordinate change and ${}^{\mathfrak{C}'}_{\mathfrak{W}}\bar{\mathcal{E}}$ is the world to virtual camera transformation described in section 4.2.1, in particular by equation 4.2.1.

In short, it is shown that for small angles Snell's law is equivalent to considering a virtual camera with a different projection center.

4.3 Reconstruction

After the previous correction is applied to the images, equations 4.1.2-4.1.3 are no longer valid in describing Snell's law in these images, making them unsuitable in the triangulation step of a reconstruction algorithm. These equations will now be corrected so they are valid for images previously transformed by the above procedure. As figure 4.3.1 shows, it is necessary for each pixel in each image to calculate the pair $(\mathbf{p}_3, \mathbf{v}_3) \in \mathbb{E}^3 \times \mathbf{T}_{\mathbf{p}_3}\mathbb{E}^3$ from $(\mathbf{p}_1, \mathbf{v}_1) \in \mathbb{E}^3 \times \mathbf{T}_{\mathbf{p}_1}\mathbb{E}^3$ since these are what define the light rays' real trajectory after hitting the interface.

Once again consider a cartesian chart with the interface at $z = 0$. Defining parametrically the straight line through $\mathbf{p}_1$ with direction $\mathbf{v}_1$ and finding its intersection with the plane $z = 0$:

$$\mathbf{p}_3 = \left(p_1^x - \frac{p_1^z}{v_1^z}v_1^x, p_1^y - \frac{p_1^z}{v_1^z}v_1^y, 0\right) \tag{4.3.1}$$

Snell's correction translated the projection center of the camera according to 4.2.1, so:

$$\mathbf{p}_2 = (p_1^x, p_1^y, kp_1^z)$$

which results in

$$\mathbf{v}_2 = \mathbf{p}_3 - \mathbf{p}_2 = \left(-\frac{p_1^z}{v_1^z} v_1^x, -\frac{p_1^z}{v_1^z} v_1^y, -k p_1^z \right)$$

Now, equations 4.1.2-4.1.3 are again valid. Noting that

$$a = \operatorname{sgn}(a) \cdot |a| = \operatorname{sgn}(a)\sqrt{a^2}$$

it follows then

$$\mathbf{v}_3 \propto \left(v_1^x, v_1^y, -\sqrt{\frac{1-k^2}{k^2}\left((v_1^x)^2 + (v_1^y)^2 \right) + (v_1^z)^2} \right) \tag{4.3.2}$$

4.4 Summary of the proposed algorithm

The algorithm is applied in two steps, altering the usual stereo reconstruction process:

- **Image Correction** - This step consists of applying the homography 4.2.3 to each image while performing image rectification as indicated in section 3.2. Thus the complete rectification process becomes:

  1. Each pixel commonly described in the image chart ($\mathfrak{i}_1$) is first described in the natural projection chart through ${}^{\mathfrak{p}}_{\mathfrak{i}_1}\mathcal{E}$.
  2. Once on this plane it is possible to compensate non-linear distortions introduced by the camera acquisition hardware. This distortion is usually radial in nature, characterized through the even powers of a polynomial in $r = \sqrt{({}^{\mathfrak{p}}p^1)^2 + ({}^{\mathfrak{p}}p^2)^2}$, but tangential distortion is also correctable.
  3. Apply Snell homography $\mathbf{H}$ given by equation 4.2.3.
  4. If an extrinsic correction is necessary (a change in the desired projection plane with the projection center fixed) it is possible to do so at this point.
  5. It is then possible to choose desired intrinsic parameters for the new virtual camera and change back to an image chart using ${}^{\mathfrak{i}_2}_{\mathfrak{p}}\mathcal{E}$. These new extrinsic parameters are usually chosen so as to minimize information loss contained in the image.

  Steps 3 and 4 are illustrated in figure 4.4.1.

- **Reconstruction** - Once matching has been performed, the actual reconstruction is obtained using equations 4.3.1 and 4.3.2 to define each line used in triangulating the reconstructed point.

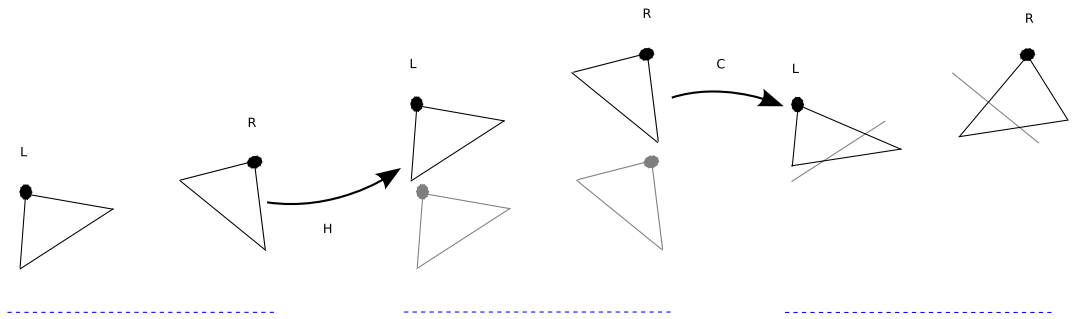


Figure 4.4.1: Extrinsic Snell correction step and camera alignment. The first step is to project on a new virtual camera with altered projection center, the second step (used in conventional stereo) is to align both camera's projection plane so epipolar lines are horizontal.

Since conversion from disparity space to a cartesian chart is (for scenes not submerged) a projective transformation, it is possible to describe a given plane in projective space. It is known that an hiper-plane $\bar{\mathbf{s}} \in \mathbb{P}^3$ divides the Euclidean space in two regions (denoted $+$ and $-$) and that two points $\bar{\mathbf{p}}_1, \bar{\mathbf{p}}_2 \in \mathbb{P}^3$ belong to the same region if the sign of $\bar{\mathbf{s}}^T\bar{\mathbf{p}}_1$ agrees with the sign of $\bar{\mathbf{s}}^T\bar{\mathbf{p}}_2$. Please note that both $\bar{\mathbf{s}}$ and $-\bar{\mathbf{s}}$ denote the same plane so there is no natural convention for the sign of either region. It is only being possible to check if two points fall on the same region. This is enough for the purpose at hand since it is possible to compare points to a pre-defined submerged point.

In the calibrated chart, the water plane is at $z = 0$, described in projective coordinates as $\bar{\mathbf{k}} = [0 : 0 : 1 : 0]$. This plane can be taken to the camera's local chart using the homography resulting from the change of coordinates ${}^{\mathfrak{L}}_{\mathfrak{W}}\mathcal{E}$ and then to disparity space through ${}^{\mathfrak{D}}_{\mathfrak{L}}\mathcal{E}$. It can then be used to identify which pixels show submerged scene elements. This is important since if Snell's correction has been applied an image is only valid for pixels corresponding to submerged features, or, if Snell's correction has not been applied, for pixels not corresponding to submerged features.

If an underwater reconstruction is pretended, the interpretation of $\mathfrak{D}$ as a chart is only approximate, the exact result obtained using 4.3.1 and 4.3.2 to each matched pixel, followed by an intersection of the two straight lines as described in appendix B. Note though that for points directly on the interface the approximation is exact (Snell's correction maps the interface plane on itself) and the distortion is continuous so the previous paragraph's discussion still applies.

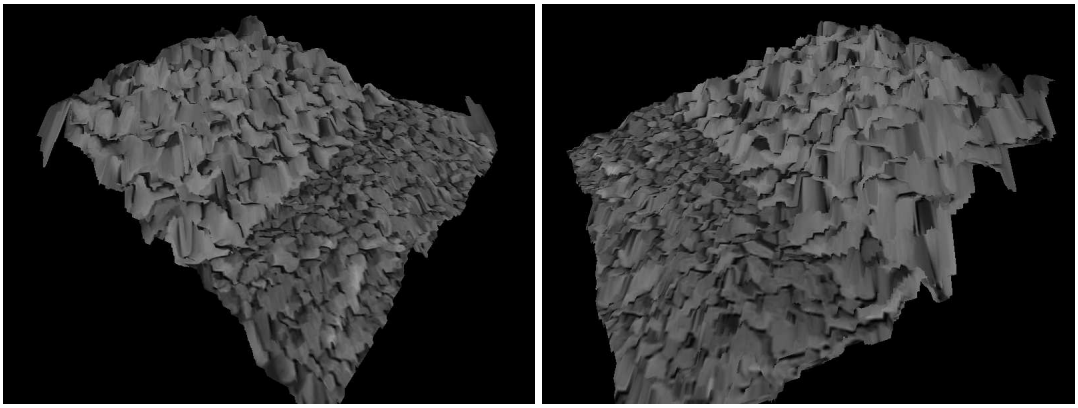
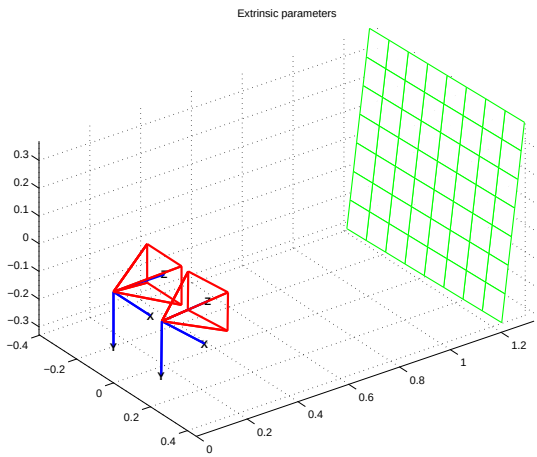


Figure 4.5.1: Reconstruction visualization.



Parameter	Camera	
	Left	Right
Position (m)	(.53,.36,1.26)	(.53,.61,1.26)

Figure 4.5.2: Extrinsic parameter visualization of the stereo system for the synthetic images. The parameters used are those considered typical. The table indicates the camera position with respect to the world referential, calibrated to be centered at the top left corner of the observed grid.

4.5 Results

To help visualizing the reconstructed environment, a C application using OpenGL (with a matlab interface) was developed, allowing for smooth navigation in the environment using the mouse and keyboard. Figure 4.5.1 presents a screenshot of the running application.

To test the viability of Snell correction a few synthetic images of a submerged plane parallel to the interface at various depths were rendered. The cameras were positioned at 1.3m above the interface with a baseline of about $0.3m$ (see figure 4.5.2). Although it is not important which matching algorithm is used, in the experiments described next, Sun's matching algorithm [13] is used.

Once a disparity map has been obtained it is already a coordinate representation for the

observed scene in what was called the $\mathfrak{D}$ chart. Unfortunately as seen above this chart is not valid for images observing submerged scenes. Let's assume that Snell correction completely eliminates the distortion introduced by the interface (assume that the first order simplification is exact). This condition implies that the disparity map is a projective reconstruction of the scenery. The reconstructions shown in figure 4.5.3 use this assumption. As expected, the error increases with the depth at which the plane is positioned and with the angle at which the light rays hit the interface. Particularly notice the top corners where a lot of noise is present due to correction errors at a high incidence angle (about 25 degrees for the top right corner of the left camera image). Discarding these zones where the matching algorithm clearly fails and looking at the plane at a depth of 1.5m, errors of about 25cm are seen near the bottom corners of the image (incidence angle of around 15 degrees on the left camera image). Given the position of the plane with respect to the interface and the cameras, this results in a relative error of about[1] 10%. These results are shown to emphasize that although Snell correction helps the matching process, it is not enough to consider the disparity space a projective reconstruction of the scenery. When converting from disparity to world coordinates, it is necessary to use line intersection as described in section 4.3 and appendix B.

Using the same disparity maps, a better reconstruction can be obtained if equations 4.3.1 and 4.3.2 are used as described in that section. The results obtained are shown in figure 4.5.4). Each reconstructed point is now obtained through the intersection of two refracted rays. Although the existence of noise is still evident on the top corners (since this is a matching problem, not a reconstruction problem), the error on the overall image fell drastically to a worst case of about 3cm at a depth of 1.5m. This results in a relative error of about 1%. It is worthy to note that this is also the expected error due to quantization of the disparity maps for the given distance.

The best results though are obtained when disparity is taken in both dimensions. The results are shown in figure 4.5.5. The iterative algorithm described in section 3.3.3 converged correctly on these images and as the reconstruction shows the noise on the top corners is no longer present. On the whole image there are no errors with magnitude greater than the 3cm expected due to quantization.

Another experiment with camera parameters adequate for underwater reconsctruction (such that the matching algorithm does not fail) was performed. The setup is illustrated in figure 4.5.6 and the results are shown in figure 4.5.7. Note that if Snell correction is not applied, the plane is reconstructed in the wrong place (at a depth of about $0.35m$. When Snell correction is applied to the exact same images, the plane is again reconstructed at the correct depth.

The reconstruction algorithm was also applied to real world scenery. Figures 4.5.8 and 4.5.9

[1]Relative error in this context is the reconstruction error over the distance of the plane to the left camera.

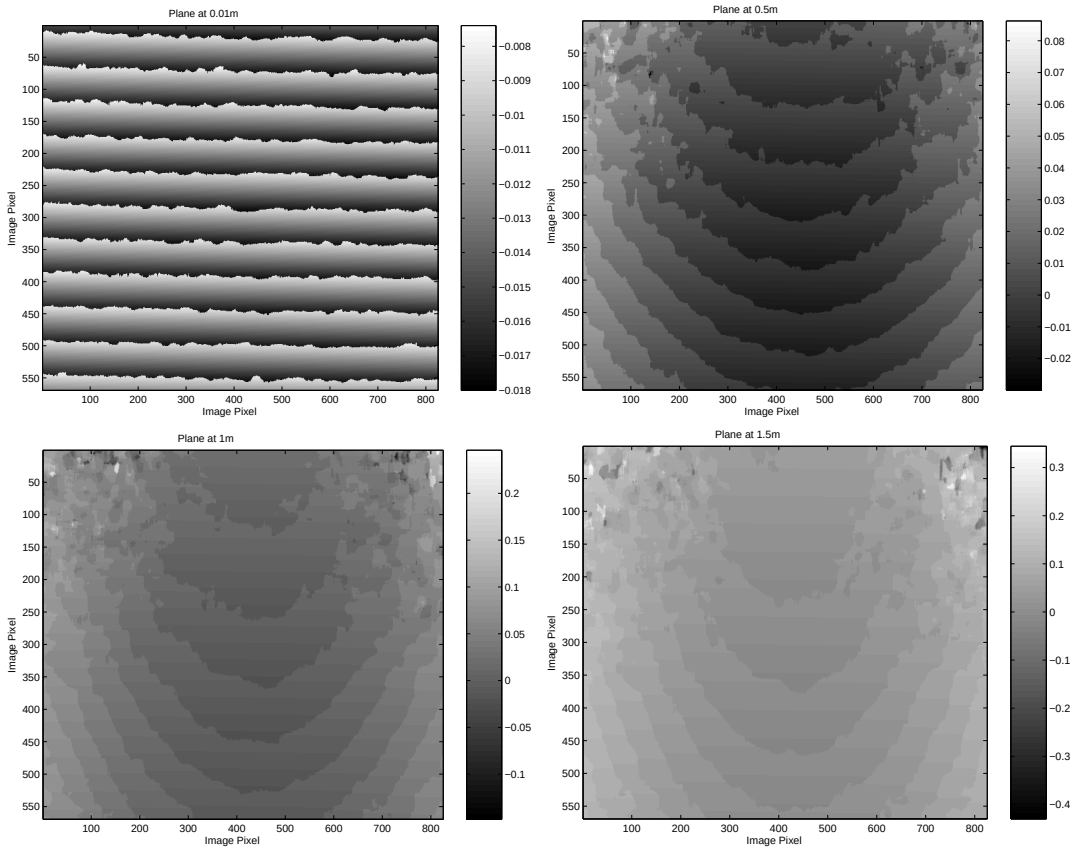


Figure 4.5.3: Reconstruction error (in meters) for each image pixel using only Snell correction, assuming it is a projective reconstruction of the observed scenery. The observed scene is a plane at the indicated depth.

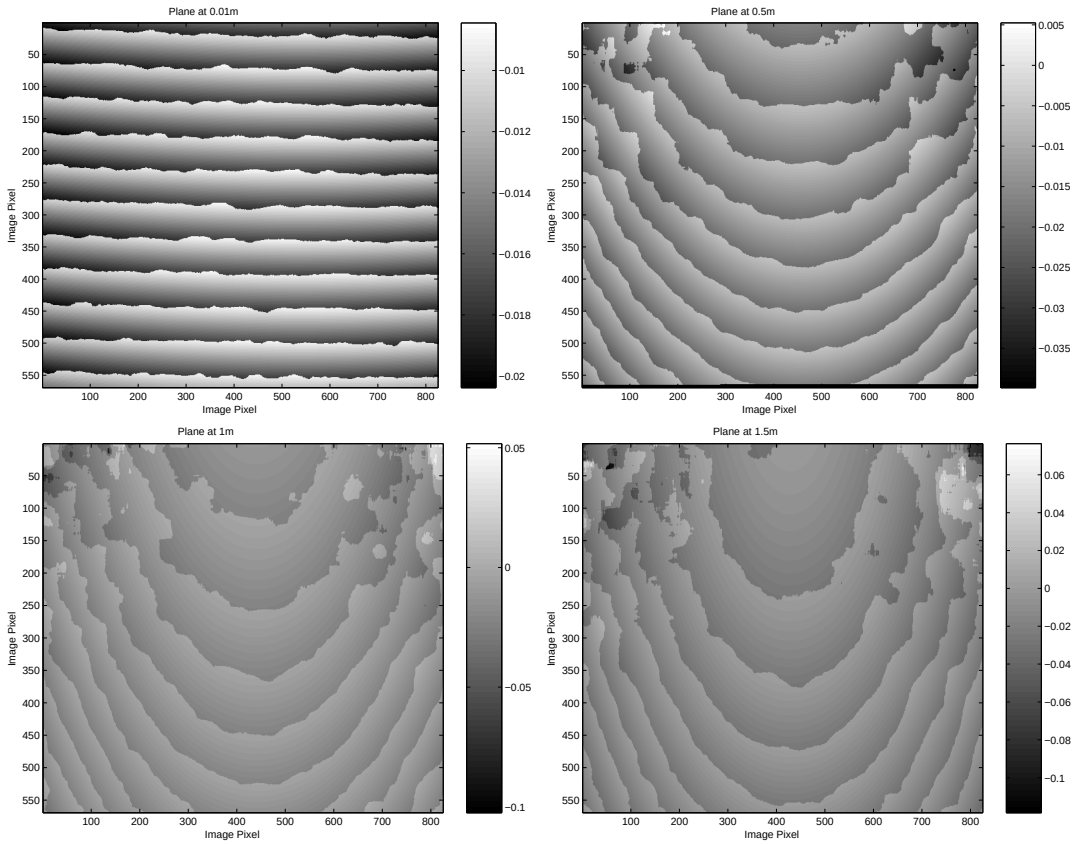


Figure 4.5.4: Reconstruction error (in meters) for each pixel in the image using Snell correction and the conversion of disparity to world coordinates described by equations 4.3.1 and 4.3.2. The reconstructed scene is a plane at the indicated depth.

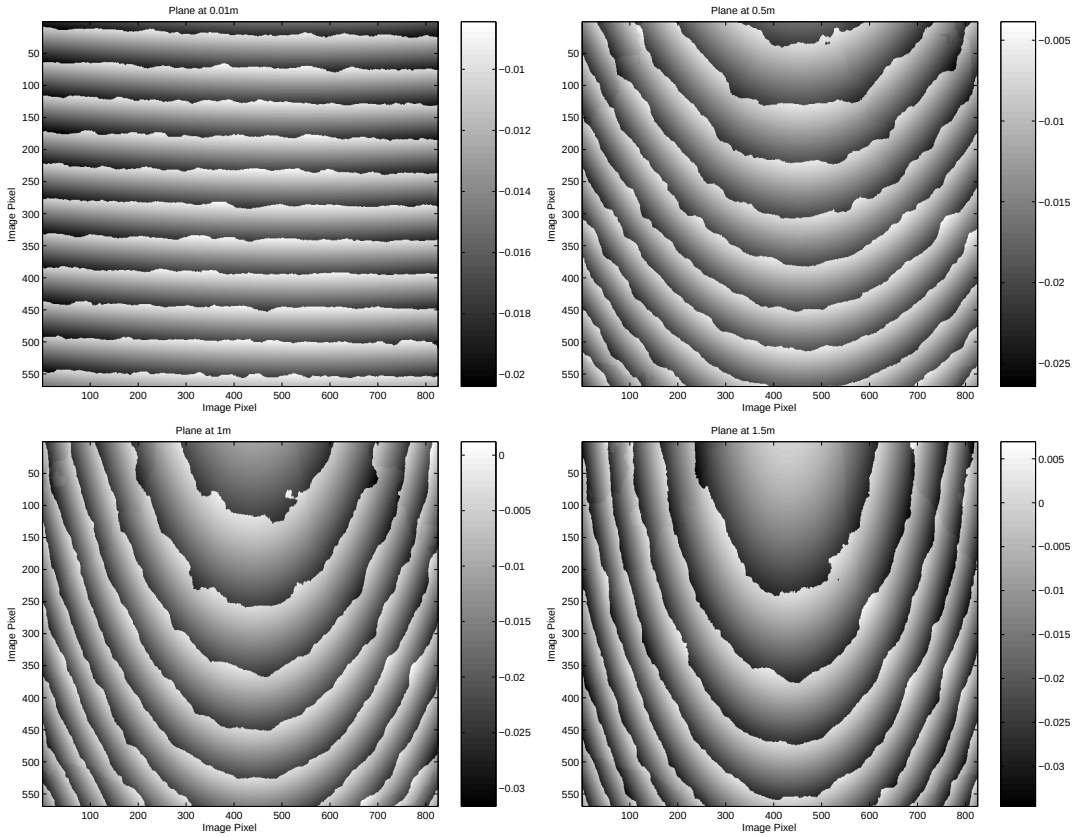


Figure 4.5.5: Reconstruction error (in meters) for each pixel in the image using 2D matching and the conversion of disparity to world coordinates described by equations 4.3.1 and 4.3.2. The reconstructed scene is a plane at the indicated depth.

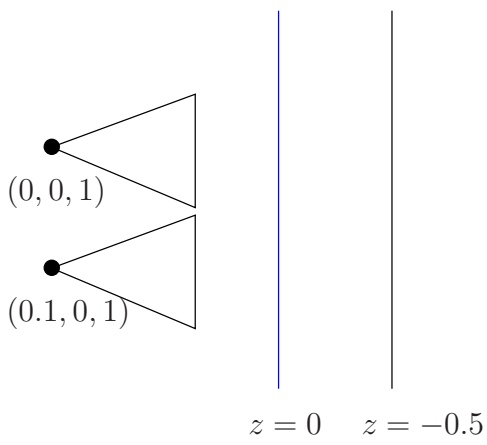


Figure 4.5.6: Synthesised scenery observed by two cameras placed side by side at a height of $1m$ above the interface, looking in the $-\hat{\mathbf{e}}_z$ direction. The interface is in the plane $z = 0$ and the scenery is a textured plane placed at $z = -0.5m$.

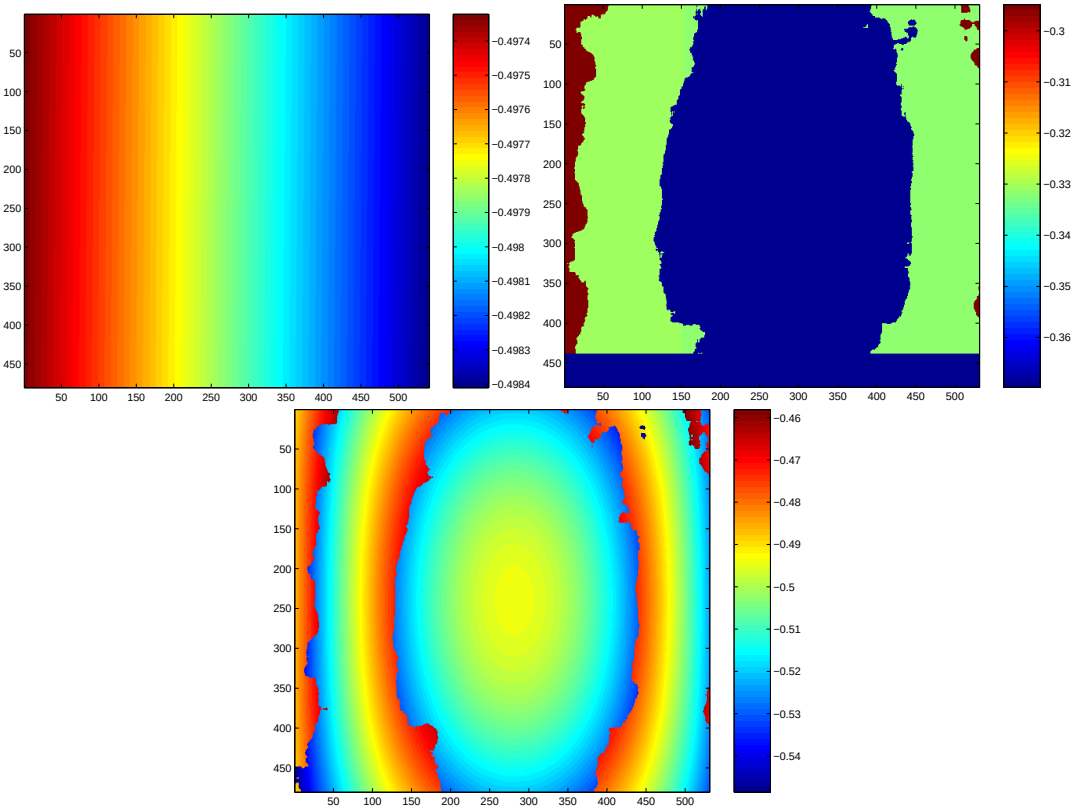


Figure 4.5.7: Results of the reconstruction of a plane. Depth of a reconstructed plane placed at $z = -0.5m$ (world coordinates) observed without the presence of an interface (top left) and with the interface when no Snell correction is performed (top right). On the bottom the reconstruction of the same images now with Snell correction applied.

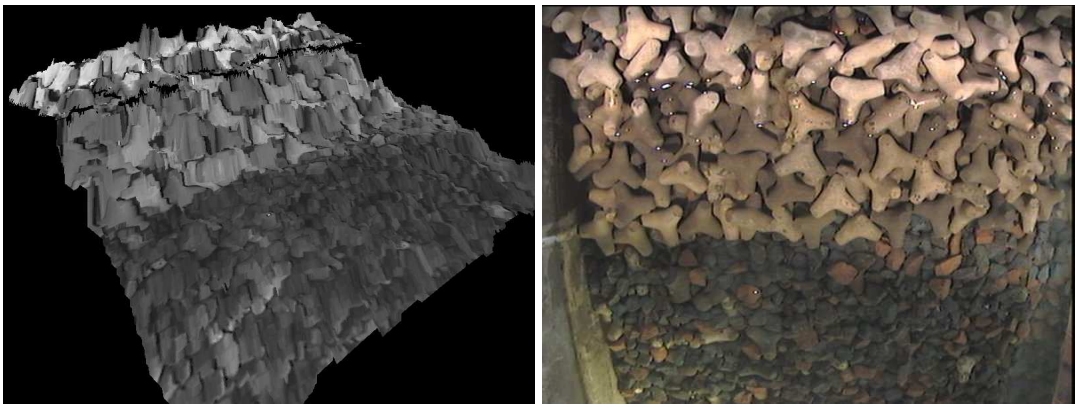


Figure 4.5.8: 3D view and left image of a model breakwater partially submerged.

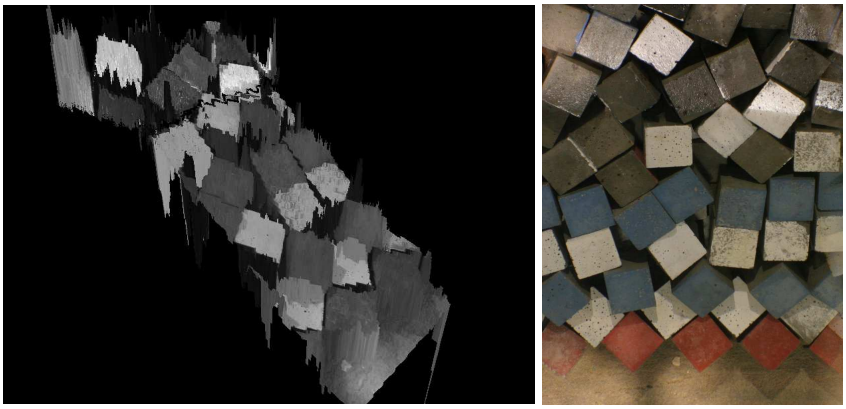


Figure 4.5.9: 3D view and left image of another model breakwater partially submerged.

show two reconstructions of a real breakwater physical model. The first uses images taken with video low resolution PAL cameras with a baseline slightly below $40\mathrm{cm}$ and about $1.2\mathrm{m}$ above the water. The second uses images taken with a beam splitter mounted on a 6 megapixel still camera. The baseline is about $5\mathrm{cm}$ at $1.2\mathrm{m}$ above the interface. Notice in both reconstructions the discontinuity near the top where the underwater and overwater reconstructions are fused. Unlike the synthetic images these are not so feature rich (for example dark shadows appear between rocks), resulting in some matching errors. Better results should be possible with algorithms that deal with occlusions and lack of rich texture.

Chapter 5

Interface Estimation

This chapter's intention is to describe an algorithm for estimating the shape of an interface between two media when observed by a pair of calibrated cameras. It is assumed that, when written on a chart, the interface is a function of two coordinates (for example $z = f(x, y)$), imposing some restrictions on its shape. This is not too restrictive though, for most water surfaces obey this restriction unless in the presence of heavy undulation (although it is by no means restricted to water surfaces).

Each image's reconstruction is also assumed to be known, in other words that for each point $\mathbf{q} \in \mathbb{E}^2$ on each image it is known which $\mathbf{p} \in \mathbb{E}^3$ originated it. How to obtain these correspondences is beyond the scope of this chapter, although a few possibilities were mentioned in chapter 3.

The final algorithm assumes a form similar to the ones used for dense stereo matching using dynamic programming, with only the cost function adapted to this specific problem.

5.1 Problem Formulation

Figure 5.1.1 illustrates the problem at hand. When using a single camera, there is no way to obtain the parameters which define the interface even if the correspondence of points on the camera with points on the scenery is available as described above. The problem is that there is still an undefined degree of freedom where the interface can accommodate itself by changing its position and orientation accordingly as described next.

Consider once again (as in section 4.1) that $\hat{\mathbf{u}} \in \mathbf{T}_{\mathbf{p}}\mathbb{E}^3$ denotes the unit vector normal to the interface at a given point $\mathbf{p} \in \mathbb{E}^3$ and $\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2 \in \mathbf{T}_{\mathbf{p}}\mathbb{E}^3$ the incident and refracted unit vectors (respectively). As seen, these entities are related through the equation

$$k_1(\hat{\mathbf{v}}_1 \times \hat{\mathbf{u}}) = k_2(\hat{\mathbf{v}}_2 \times \hat{\mathbf{u}})$$

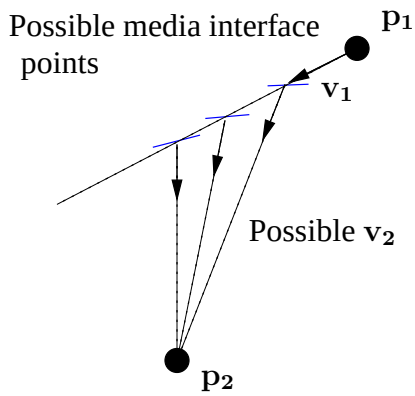


Figure 5.1.1: Graphical representation of the possible media transition points. As illustrated, each of these will have a different tangent plane consistent with the observed data.

Note that the properties of the cross product allow for the equation to remain valid if $\hat{\mathbf{u}}$ loses its unit norm attribute so this imposition is relaxed. The same does not hold true for $\hat{\mathbf{v}}_1$ and $\hat{\mathbf{v}}_2$ which have to have equal norm (unit norm is chosen). Rewriting the equation:

$$(k_1\hat{\mathbf{v}}_1 - k_2\hat{\mathbf{v}}_2) \times \mathbf{u} = 0$$

This equation states that $\mathbf{u}$ must be collinear with $k_1\hat{\mathbf{v}}_1 - k_2\hat{\mathbf{v}}_2$. Since $\mathbf{u}$'s norm is not important, the previous system is under-specified, the solution being given apart from a scale factor. One possible solution is then

$$\mathbf{u} = k_1\hat{\mathbf{v}}_1 - k_2\hat{\mathbf{v}}_2 \tag{5.1.1}$$

Note that although $\mathbf{v}_1$ is fixed when choosing a given point on the image, (and its correspondence on $\mathbb{E}^3$), the same does not happen for $\mathbf{v}_2$ since it depends on the actual location of the interface so it is not possible to solve for $\mathbf{u}$. This is illustrated in figure 5.1.1.

The problem can be solved though if another image observing the scenery from a different (calibrated) viewpoint is available. Assuming the interface passes through a certain point, the orientation that the interface has to have to be consistent with the first image is calculated, followed by the orientation consistent with the second image. This results in an error measure measuring the difference in orientation.

Lets assume then that the interface passes through the point $\mathbf{p}_i$. Since it is assumed that the correspondence to world points is known, $\mathbf{p}_2 \in \mathbb{E}^3$ and $\mathbf{q}_2 \in \mathbb{E}^3$ (see figure 5.1.2) are known. From the first image $\mathbf{v}_1$ and $\mathbf{v}_2$ are completely characterized:

$$\mathbf{v}_1 = \mathbf{p}_i - \mathbf{p}_1$$
$$\mathbf{v}_2 = \mathbf{p}_2 - \mathbf{p}_i$$

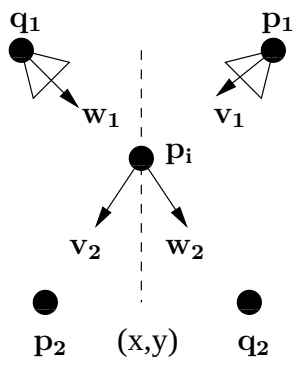


Figure 5.1.2: Interface estimation algorithm representation.

which results in a possible orientation for the interface at $\mathbf{p}_i$ given by equation 5.1.1:

$$\mathbf{u}_1 = k_1\hat{\mathbf{v}}_1 - k_2\hat{\mathbf{v}}_2$$

repeating the same for the second image's information, a second possible orientation for the interface $\mathbf{u}_2$ is obtained.

By definition, the angle of these two vectors is given by

$$\cos(\theta) = \frac{\langle \mathbf{u}_1, \mathbf{u}_2 \rangle}{\|\mathbf{u}_1\| \, \|\mathbf{u}_2\|}$$

And this is the value used as a cost function

$$C(\mathbf{p}_i) = \frac{\langle \mathbf{u}_1, \mathbf{u}_2 \rangle}{\|\mathbf{u}_1\| \, \|\mathbf{u}_2\|}$$

If no other restrictions (such as smoothness) are intended, the best candidate for the interface to pass through on a set of points S is given by

$$\mathbf{p}^* = \arg\min_{\mathbf{p}\in S} C(\mathbf{p})$$

Please note that even in the absence of mismatches this optimization problem is very sensitive due to quantization noise in the disparity maps necessary for the application of the algorithm. Figure 5.1.3 illustrates this problem, where the sensitivity of the algorithm is obvious since there is not a clearly defined minimum, but rather a noisy valley. As described later, smoothing the input disparity maps can help reduce this problem if some smoothness assumptions of the observed scene and interface are imposed. The figure shows an example with previous smoothness of the disparity maps and two without (one of a synthetic scene, the other of a real scene).

Since the surfaces considered are surely to be smooth, it makes sense to include this cost function on a dynamic programming algorithm of the same type as those widely used in stereo reconstruction.

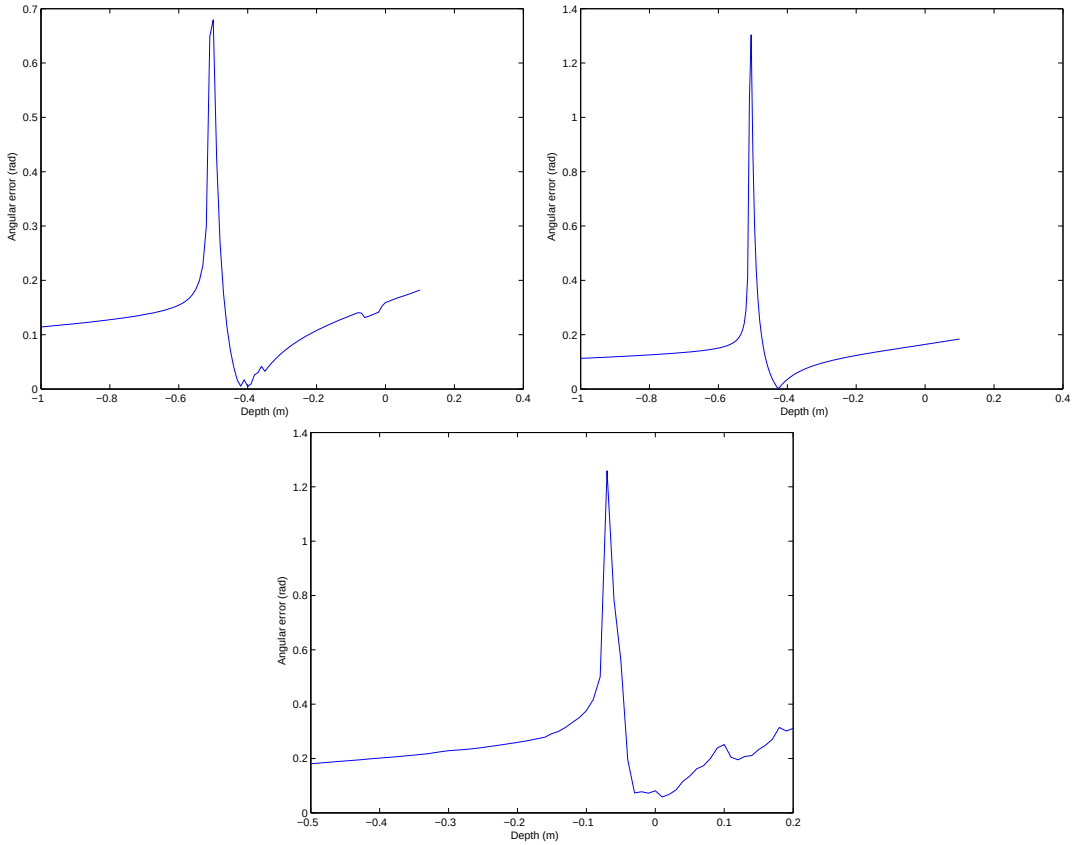


Figure 5.1.3: Interface estimation error function along the z coordinate when the x and y coordinates in the world referential are fixed. Top left: no disparity map smoothing in a computer generated scene; Top right: with disparity map smoothing in a computer generated scene; Bottom: No disparity map smoothing for real breakwater model images.

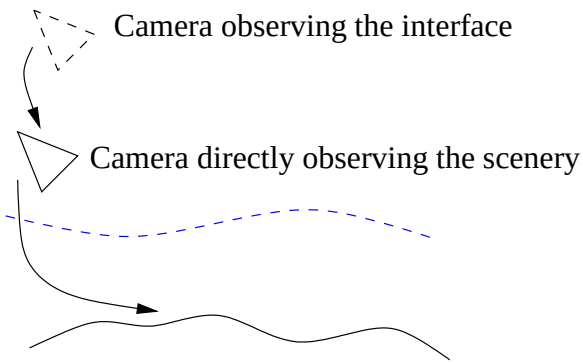


Figure 5.2.1: Two sets of stereo pairs are are obtained, one where it is possible to reconstruct the scenery (taken without an interface, or with the interface in a planar configuration), the other one observing the interface to be estimated. A dense matching algorithm is applied to each image of the first pair with the corresponding image of the second pair. Since the first pair can be reconstructed, it is possible to follow the disparity maps to obtain the reconstruction of the second pair.

5.2 Implementation Considerations

Since what is usually needed is to obtain a dense reconstruction of the interface for all points on a camera, it makes sense to build the cost function on the referential $\mathfrak{D}$. In a dynamic programming setting, the cost volume $C(i, j, k)$ is built by evaluating the cost function above at the point ${}^{\mathfrak{D}}\mathbf{p} = (i, j, k)$ and then extracting the surface $S(i, j)$ such that

$$c = \sum_{i,j} C(i, j, S(i, j))$$

is minimum. The surface S must obey some smoothness constraints, which fit nicely in the dynamic programming setting. As an example, Sun's algorithm can once again be used.

The presented algorithm requires that an image's reconstruction be known apriori. This can be accomplished by previously observing the scenery with the interface in a planar configuration and applying the reconstruction algorithm as described in chapter 4. When an interface estimation is needed, newly acquired images are first matched to the previously taken ones of which the reconstruction information is already known. Notice that both the newly acquired left and right images need to be matched to the previously taken stereo pair. These stereo matches are not trivial to obtain since there are no constraints limiting the search space. If the distortion of the interface with respect to its planar configuration is small, it is possible to search a given feature in a restricted rectangle centered at its nominal position on the other image. Figure 5.2.1 illustrates this description.

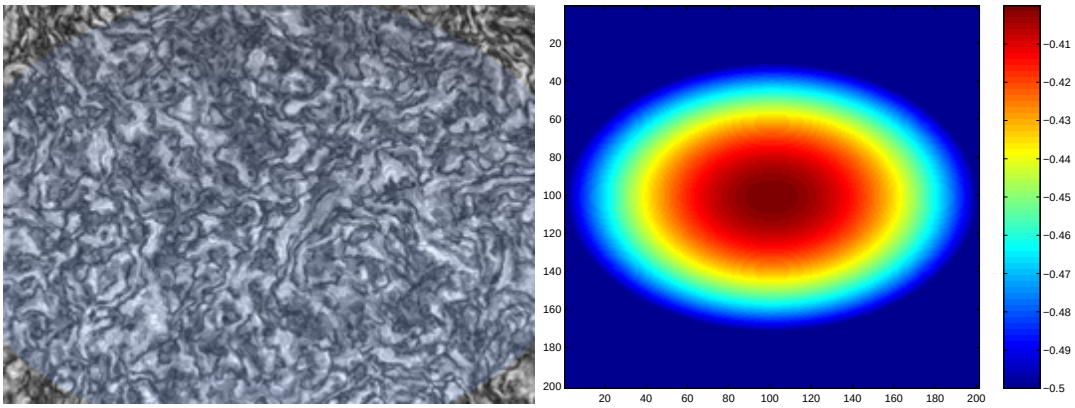


Figure 5.3.1: Synthetic image used for interface estimation. The scene is a textured plane at a distance of about $1.5m$ from the cameras, with a water bubble with a $1dm$ width at the center. Left: image observed by the left camera (the water bubble is seen in a bluish shade); Right: orthographic map of the bubble height (in meters).

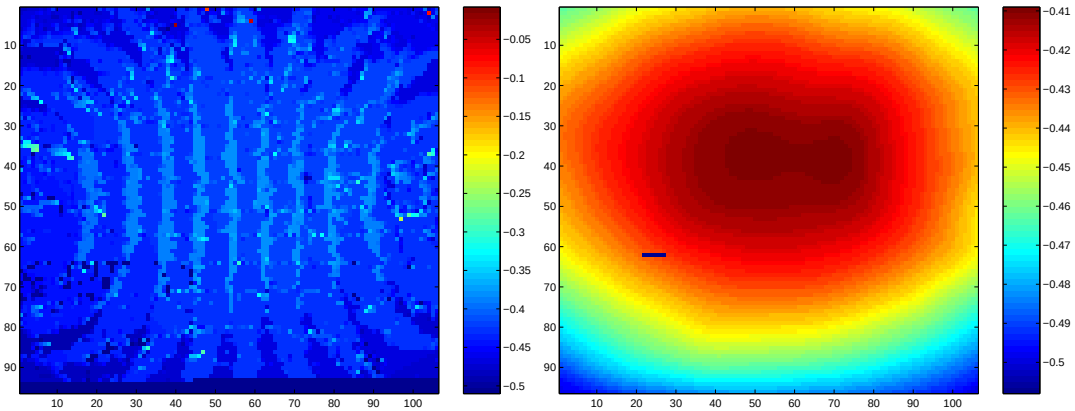


Figure 5.3.2: Obtained results using low pass filtering of the input disparity map. Left: no smoothing applied; Right: Low pass filtering of input data. Depths are in meters.

5.3 Results

The results obtained are shown for two different images. The first is a synthetic, computer generated, image of a richly textured plane on which a “drop” of water was placed. This image is usefull for error measurement and is shown in figure 5.3.1. The second image is a real world image with the interface in a planar configuration so that error can also be measured (since its position can be calibrated using a calibration rig).

Unfortunately reconstruction errors (including quantization errors) present in the reconstructions given to the interface estimation algorithm introduce too much noise for it to be usefull (the first image in figure 5.3.2 shows this clearly). For the purposes described, it is safe to assume that the interface has a very smooth variation allowing for low pass filtering of the input

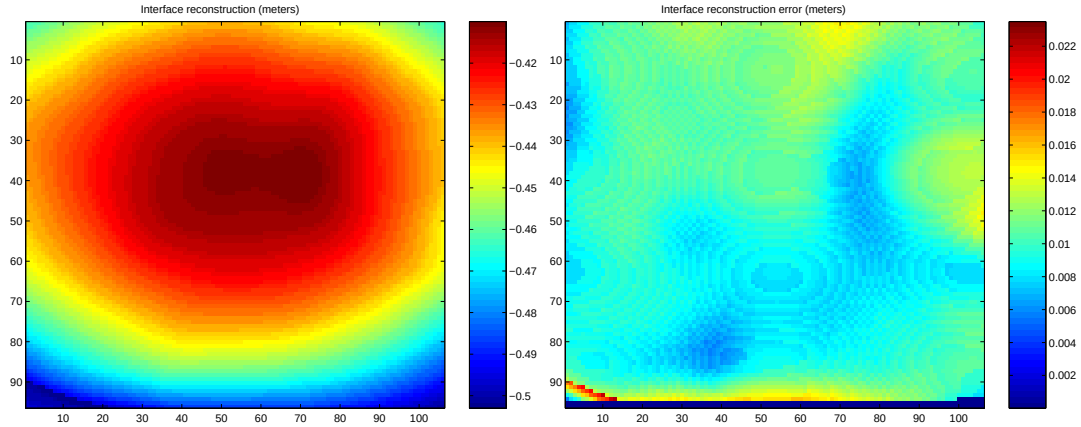


Figure 5.3.3: Global interface estimation error. Left: Interface estimation using low pass filtering; Right: corresponding error images of the estimation algorithm. Depths are in meters.

data. Figure 5.3.2 shows the obtained results after applying several low pass filters with different bandwidth. Please note that it is the input data that is smoothed and not the results given by the algorithm, emphasizing that the problem is highly dependent on the quality of its input disparity maps. Since the observed scene to be reconstructed is a plane, it is also possible to apply a linear regression to its reconstruction prior to the application of the algorithm.

Figure 5.3.3 shows the reconstruction error for each estimated point. As shown, the error is not greater than about $1.5cm$ in almost the whole image, except as should be evident the places where the bubble touches the plane since these zones do not obey the smoothness criterion necessary to justify the application of the lowpass filter.

Instead of smoothing by a low pass filter, it is also possible to apply a higher order polynomial regression. The results are shown in figure 5.3.4 (see appendix A for the theory behind polynomial regressions). Unfortunately they show that this particular interface configuration does not fit well with a global polynomial regression.

If the underwater observed scene is not planar, the disparity maps cannot be smoothed. Figure 5.3.5 shows an estimation of a planar interface at $z = 0$ when observing real images taken of a partially submerged model breakwater without any kind of smoothness to the input disparity maps. A median filter is applied to the results obtained in the hopes of canceling the noise present in the images. Note that the scenery is only partially submerged so at the top of the image were there is no interface, it is “seen” as being glued to the breakwater model itself.

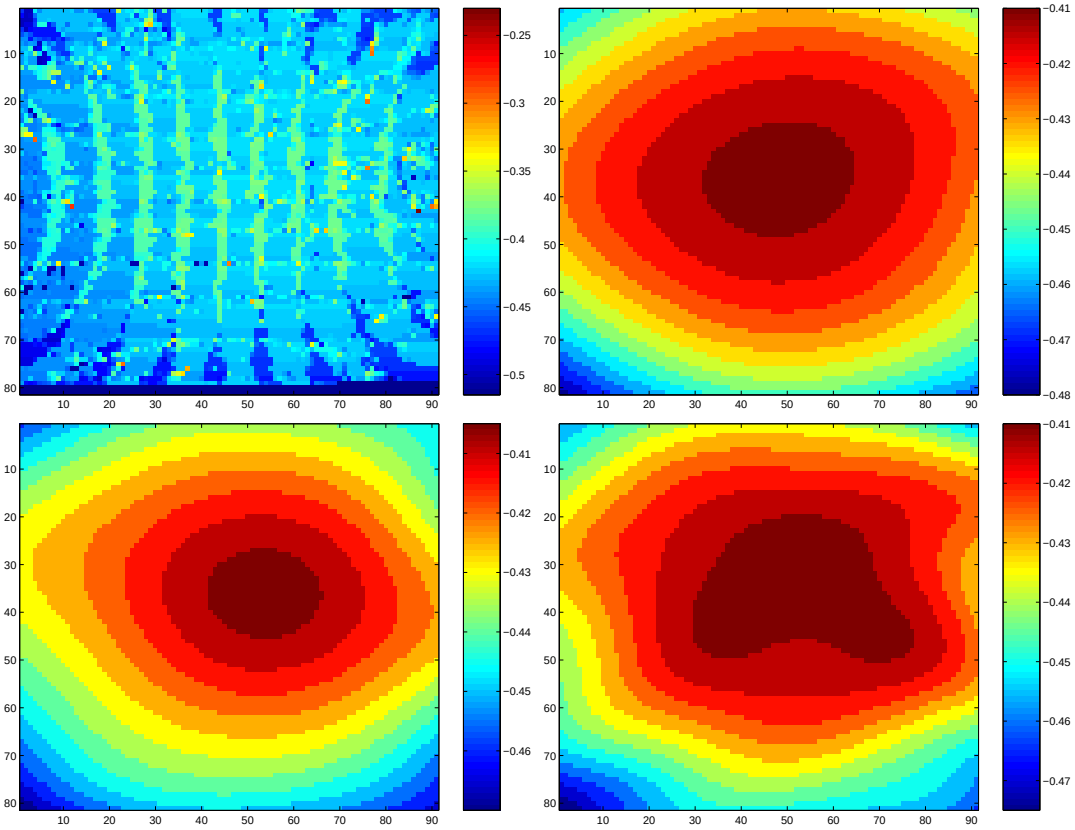


Figure 5.3.4: Obtained results using polynomial regression of the input disparity map. Top Left: no regression; Top right: 4th order bivariate polynomial approximation; Bottom left: 6th order bivariate polynomial approximation; Bottom right: 9th order bivariate polynomial approximation. Depths are in meters.

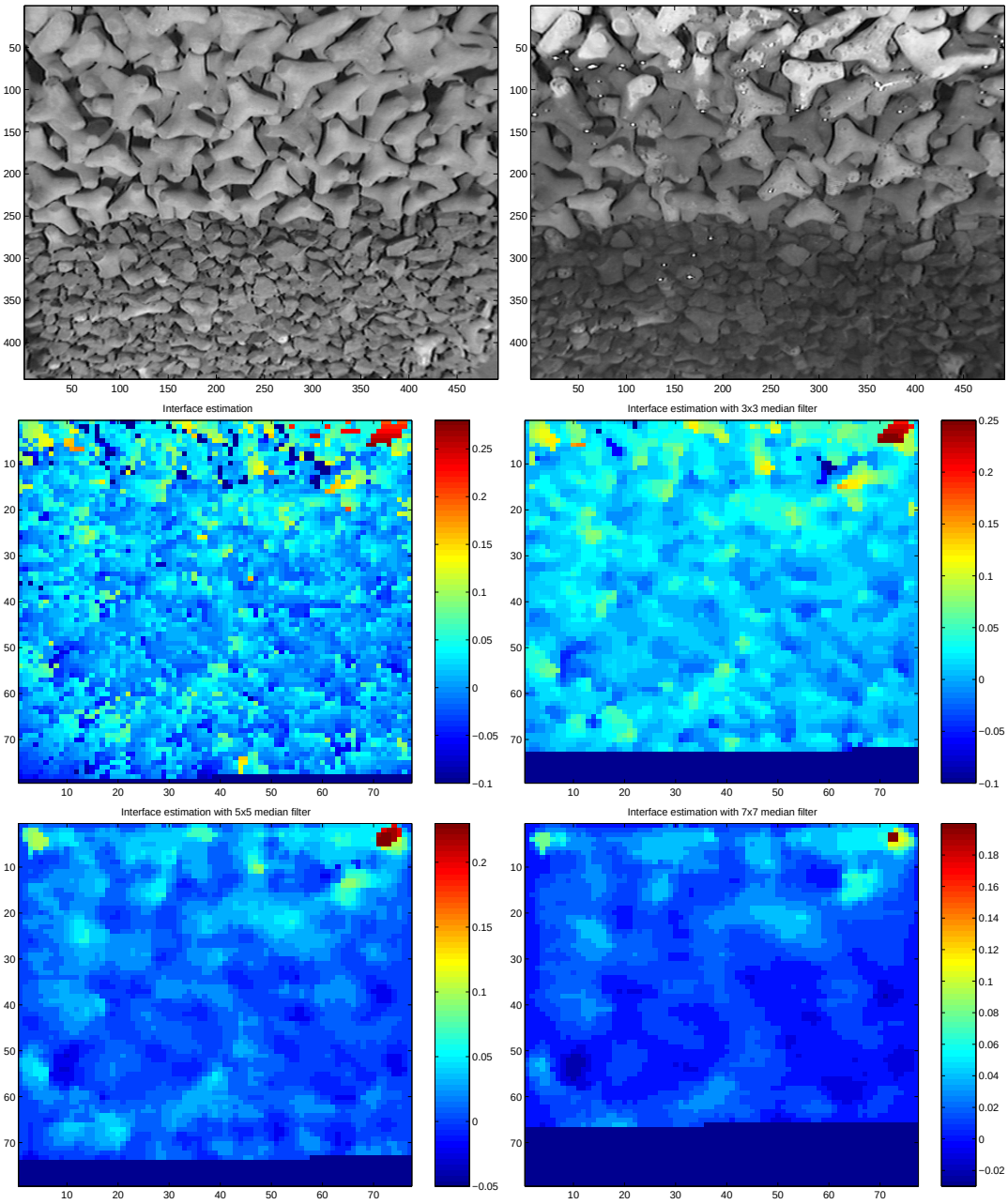


Figure 5.3.5: Interface estimation with images of a real breakwater model. Top images show the left image without and with the interface. The rest other images illustrate the results of the estimation of this interface. Due to the high noise present in the estimation, median filters of varying width are applied as indicated. Depths are in meters.

Chapter 6

Conclusion

Stereo reconstructions of submerged scenes present a few additional, hard to solve, difficulties when compared to standard stereo in the absence of an interface. These difficulties arise from the refraction effect which bends light rays that pass through it, breaking epipolar geometry and introducing a magnification effect on the observed image when the interface assumes a planar shape. If the interface is allowed to assume other shapes, the distortion introduced can vary greatly.

The described method, although not completely solving the problem, allows the use of standard stereo algorithms when the interface assumes a planar shape as long as the incidence angle is constrained to a cone of about ± 15 degrees. The method consists of a preliminary image correction applied directly to each image due to an extrinsic parameter correction, turning the epipolar restriction "almost valid". This step is easilly inserted in the image rectification step commonly used to make epipolar lines horizontal. After the matching process is complete, the actual image reconstruction needs a slight, exact and closed form adjustment as well.

Experience shows that if only single dimension matching is used, the quality of the reconstruction begins to degrade after incidence angles greater than about 15 degrees for a conventional stereo setup due to epipolar geometry failure. If two dimentional matching is to be used, this angle is only limited by the computational resources available.

An interface estimation algorithm that allows for the obtention of its surface from stereo image pairs was also described. It is very similar to a conventional dynamic programming stereo algorithm, where only the cost function needs to be adapted. Unfortunatelly it is sensitive to noise (discretization noise or matching failures) so it works best when the shape of the submerged scenery allows for a regression to be performed. The ideal observed scenery is a richly textured submerged plane. If all goes well, the reconstruction errors expected are of the same order as those of a standard stereo reconstruction with the same resolution and distance.

Appendix A

Polynomial Regression

It is possible to apply a polynomial regression to a set of data points with the intent of smoothing (and interpolating) the set. Suppose $P_{mn}(X,Y) \in \mathscr{P}^{mn}$ is the element of order mn of a base for the bivariate polynomials[1]. It is then possible to describe any polynomial of lower order as a linear combination of this basis:

$$P(X,Y) = \sum_{i,j=0}^{N} a_{ij} P_{ij}(X,Y)$$

where N is the maximum order pretended for both variables of the polynomial.

Supposing that what is wished is to minimize the square error of the regression, the cost function for a point might be

$$e_l^2 = (P(X_l, Y_l) - Z_l)^2 = \left(\sum_{i,j=0}^{N} a_{ij} P_{ij}(X,Y) - Z_l \right)^2$$

where (X_l, Y_l, Z_l) is the l'th sample of a total of K samples to which the regression is to be applied. This resulting in a global error given by

$$E = \sum_{l=1}^{K} e_l^2 = \sum_{l=1}^{K} \left(\sum_{i,j=0}^{N} a_{ij} P_{ij}(X_l, Y_l) - Z_l \right)^2$$

Since the function to be minimized is a positive definite quadratic with no restrictions, the necessary and sufficient optimality condition is

$$\frac{\partial E}{\partial a_{ij}} = 0$$

[1]Bivariate polynomials have two independent variables. Thus the first index corresponds to the maximum order of the first variable (X) and the second index the maximum order of the second variable (Y).

So

$$\frac{\partial E}{\partial a_{mn}} = 2\sum_{l=1}^{K}\left(\sum_{i,j=0}^{N} a_{ij}P_{ij}(X_l,Y_l) - Z_l\right)P_{mn}(X_l,Y_l) = 0$$
$$\Leftrightarrow \sum_{i,j=0}^{N} a_{ij}\sum_{l=1}^{K} P_{ij}(X_l,Y_l)P_{mn}(X_l,Y_l) = \sum_{l=1}^{K} Z_l P_{mn}(X_l,Y_l)$$

Resulting in a system of N^2 linear equations with N^2 variables. It can be re-written in matrix form where to abbreviate the notation $(\cdot) \equiv \sum_{l=1}^{K}(\cdot)$ and $P_{ij} \equiv P_{ij}(X_l,Y_l)$ is used:

$$\underbrace{\begin{bmatrix} (P_{00}^2) & (P_{10}P_{00}) & (P_{01}P_{00}) & \dots & (P_{NN}P_{00}) \\ (P_{00}P_{10}) & (P_{10}^2) & (P_{01}P_{10}) & \dots & (P_{NN}P_{10}) \\ (P_{00}P_{01}) & (P_{10}P_{01}) & (P_{01}^2) & \dots & (P_{NN}P_{01}) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (P_{00}P_{NN}) & (P_{10}P_{NN}) & (P_{01}P_{NN}) & \dots & (P_{NN}^2) \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} a_{00} \\ a_{10} \\ a_{01} \\ \vdots \\ a_{NN} \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} (Z_l P_{00}) \\ (Z_l P_{10}) \\ (Z_l P_{01}) \\ \vdots \\ (Z_l P_{NN}) \end{bmatrix}}_{\mathbf{b}}$$

The solution seeked is then obtained as the solution of the system $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$. Note though that this system is usually numerically very ill conditioned if the standard polynomial base ($P_{ij}(X,Y) = X^iY^j$) is used. The problem is due to the high powers involved, making the last line usually take values much greater (for data spread in an area not close enough to the origin) than the first.

In order to help solve the problem an orthogonal polynomial base is chosen (see [19]), in an interval from -1 to 1. In particular Legendre polynomials are chosen as described next. So

$$\int_{-1}^{1}\int_{-1}^{1} P_{ij}P_{mn}dXdY = c_{ij}\delta_{im}\delta_{jn} \quad \forall i,j,m,n \in [0..N] \tag{A.1}$$

where c_{ij} are non zero constants and δ_{ij} is the Kronecker delta function[2]. If $c_{ij} = 1\ \forall i,j \in [0..N]$ the base is said to be orthonormal. Note that if the data are uniformly distributed in the interval (as is of interest to the problem in this work), matrix $\mathbf{A}$ will be almost diagonal and as such, non-singular.

Note that a single variable polynomial base can be used to build the bivariate polynomial base as $P_{ij}(X,Y) = P_i(X)P_j(Y)$. It is easy to check that this construction results in orthonormal polynomials:

[2] $\delta_{ij} = 1$ if $i = j$ and 0 otherwise.

$$
\begin{aligned}
\int_{-1}^{1}\int_{-1}^{1} P_{ij}(X,Y)P_{mn}(X,Y)dXdY &= \int_{-1}^{1}\int_{-1}^{1} P_i(X)P_j(Y)P_m(X)P_n(Y)dXdY \\
&= \int_{-1}^{1} P_i(X)P_m(X) \int_{-1}^{1} P_j(Y)P_n(Y)dYdX \\
&= c_j\delta_{jn}\int_{-1}^{1} P_i(X)P_m(X)dX \\
&= c_i c_j \delta_{im}\delta_{jn}
\end{aligned}
$$

Otherwise, the basis consisting of single or bivariate orthonormal polynomials can be constructed through the Gram-Schmidt orthonormalization procedure applied to the previously denoted "conventional" basis. Next the basis used for single variable polynomials is presented. It is commonly known as the Legendre basis and is the result obtained by the Gram-Schmidt procedure:

$$
\begin{aligned}
P_0(X) &= 1 \\
P_1(X) &= X \\
P_2(X) &= \frac{1}{2}(3X^2 - 1)) \\
P_3(X) &= \frac{1}{2}(5X^3 - 3X) \\
P_4(X) &= \frac{1}{8}(35X^4 - 30X^2 + 3) \\
P_5(X) &= \frac{1}{8}(63X^5 - 70X^3 + 15X) \\
P_6(X) &= \frac{1}{16}(231X^6 - 315X^4 + 105X^2 - 5) \\
P_7(X) &= \frac{1}{16}(429X^7 - 693X^5 + 315X^3 - 35X) \\
P_8(X) &= \frac{1}{128}(6435X^8 - 12012X^6 + 6930X^4 - 1260X^2 + 35) \\
P_9(X) &= \frac{1}{128}(12155X^9 - 25740X^7 + 18018X^5 - 4620X^3 + 315X)
\end{aligned}
$$

Note though that the basis is orthonormal only in the interval $X \in [-1..1]$ thus it requires a pre-scaling of the data to this interval.

An additional property of the use of orthogonal polynomials over uniformly distributed data points in the interval $[-1..1] \times [-1..1]$ is that the solution for the regression of a certain order includes all the information for lesser order regressions. In particular, if

$$
\begin{bmatrix} a_{00} & a_{01} & a_{10} & \dots & a_{nn} \end{bmatrix}^T
$$

Is the solution for a regression of order nn,

$$
\begin{bmatrix} a_{00} & a_{01} & a_{10} & \dots & a_{ll} \end{bmatrix}^T
$$

will be the solution for the regression of order l, where $l < n$.

Appendix B

Intersection of Two Straight Lines

Consider the problem of finding the intersection of two straight lines in $\mathbb{R}^n$, with the possibility that parameter noise exists so that the lines "almost" intersect. It is then necessary to find the midpoint of the shortest line segment connecting two points on the two lines. Parameterizing each line as

$$\mathbf{r}(t) = \mathbf{p} + t\mathbf{v}$$

where $\mathbf{p} \in \mathbb{R}^n$ and $\mathbf{v} \in \mathbf{T}_{\mathbf{p}}\mathbb{R}^n$. The cost function that needs to be minimized to find t_1 and t_2 (characterizing the closest points on the two lines) is thus

$$E = \|\mathbf{r}_1(t_1) - \mathbf{r}_2(t_2)\| = \sum_{l=1}^{n} \left(p_1^l + t_1 v_1^l - p_2^l - t_2 v_2^l\right)^2$$

The necessary, and in this case sufficient, optimality condition is thus

$$\frac{\partial E}{\partial t_i} \propto \sum_{l=1}^{n} v_i^l \left(p_1^l + t_1 v_1^l - p_2^l - t_2 v_2^l\right) = 0 \quad i = 1,\ 2$$

This equation can be geometrically interpreted as an orthogonality condition of the searched line segment connecting the two straight lines. It has to be orthogonal to each. The linear equation can be described as a matrix using the usual dot product $\langle \cdot, \cdot \rangle$:

$$\begin{bmatrix} \langle \mathbf{v}_1, \mathbf{v}_1 \rangle & -\langle \mathbf{v}_2, \mathbf{v}_1 \rangle \\ \langle \mathbf{v}_1, \mathbf{v}_2 \rangle & -\langle \mathbf{v}_2, \mathbf{v}_2 \rangle \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} \langle \mathbf{v}_1, \mathbf{p}_2 - \mathbf{p}_1 \rangle \\ \langle \mathbf{v}_2, \mathbf{p}_2 - \mathbf{p}_1 \rangle \end{bmatrix}$$

After solving the given system (it is well defined unless the vectors are parallel), the solution seeked is given by the mid point of the line segment

$$\mathbf{q} = \frac{1}{2}\left(\mathbf{p}_1 + t_1\mathbf{v}_1 + \mathbf{p}_2 + t_2\mathbf{v}_2\right)$$

This point $\mathbf{q} \in \mathbb{R}^n$ is the intersection of the two lines.

References

[1] http://kwon3d.com/theory/calib.html

[2] http://www.coachesinfo.com/category/swimming/158/

[3] G. Hough and D. Phelp. *Digital Imaging Procesing Techniques for the Aerial Field Monitoring of Harbour Breakwaters*, 1998.

[4] Lee, J. *Introduction to Smooth Manifolds*, University of Washington, 2000.

[5] Boothby, W. *An Introduction to Differentiable Manifolds and Riemannian Geometry*, Academic Press, 1975.

[6] Carmo, M. *Geometria Riemanniana*, IMPA, 1988.

[7] Spivak, M. *A Comprehensive Introduction to Differential Geometry, Volume I*, Publish or Perish, Inc., 1979.

[8] Kanatani, K. *Geometric Computation for Machine Vision*, Oxford University Press, 1993.

[9] Pollefeys, M. *Tuturial on 3D Modeling from Images*, In conjuntion with ECCV 2000, Dublin, Ireland, Jun, 2000.

[10] Zhang, Z. *Flexible Camera Calibration By Viewing a Plane From Unknown Orientations*, Microsoft Research, 1999.

[11] Heikkilä, J. and O. Silvén. *A Four-step Camera Calibration Procedure with Implicit Image Correction*, University of Oulu, 1997.

[12] Maciel, J. and J. Costeira. “A Global Solution to Sparse Correspondence Problems”, *IEE Transactions on Pattern Analysis and Machine Intelligence*, vol 25, no. 2, Fev, 2003.

[13] Sun. C. “Fast Stereo Matching Using Rectangular Subregioning and 3D Maximum-Surface Techniques”, *International Journal of Computer Vision*, vol.47 no.1/2/3, pp.99-117, Mai, 2002.

[14] Harris, C and M. Stephens, ”A Combined Corner and Edge Detector”, *Proc. 4th Alvey Vision Conf.*, pp. 147-151, 1988.

[15] Kolmogorov, V. and R. Zabih. “Multi-Camera Scene Reconstruction Via Graph Cuts”, *European Conference in Computer Vision*, May, 2002.

[16] Meyer-Arendt, J. *Introduction to Classical and Modern Optics*, Prentice-Hall, 1995.

[17] Klein, M., and T. Furtak. *Optics*, John Wiley and sons, 1986.

[18] Hecht, E. *Optics, 2nd Ed*. Addison Wesley, 1987.

[19] Shacham. M. and N. Brauner. “Minimizing the Effects of Colinearity in Polynomial Regression”, *Ind. Eng. Chem. Res.*, no. 36, pp-4405-4412.

[20] Carpentier, M. *Análise Numérica - Teoria*, Departamento de Matemática, AEIST, pp.61-62, Fev, 1993.