

IMAGE-BASED WATER SURFACE RECONSTRUCTION WITH
REFRACTIVE STEREO

by

Nigel Jed Wesley Morris

A thesis submitted in conformity with the requirements
for the degree of Master of Science
Graduate Department of Computer Science
University of Toronto

Copyright © 2004 by Nigel Jed Wesley Morris

Abstract

Image-based Water Surface Reconstruction with Refractive Stereo

Nigel Jed Wesley Morris

Master of Science

Graduate Department of Computer Science

University of Toronto

2004

We present a system for reconstructing water surfaces using an indirect refractive stereo reconstruction method. Our work builds on previous work on image-based water reconstruction that uses single view refractive reconstruction techniques. We combine this approach with a stereo matching algorithm. Depth determination relies upon the refractive disparity of points on a plane below the water. We describe how the location of points on the water surface can be determined by hypothesizing a depth from the refractive disparity of one camera view. Then the second camera view is used to verify the depth. We compare two potential metrics for this matching process. We then present results from our algorithm using both simulated and empirical input, analyzing the results to determine the primary factors that contribute toward accurate surface point determination. We also show how this process can be used to reconstruct sequences of dynamic water and present several result sets.

Acknowledgements

I would like to acknowledge the insightful support given to me by my supervisor Kiriakos Kutulakos. I would also like to thank Allan Jepson for his thorough examination of my work and helpful comments. Thanks to all the members of the DGP Lab group for interesting discussions and for making my Masters experience enjoyable. Thanks especially to Joe Laszlo, Paul Yang and Mike Wu for mulling over my ray tracing and refraction problems and always being ready to lend a hand. Finally I wish to thank my parents and my brothers for their consistent support and encouragement.

Contents

1	Introduction	1
1.1	Motivation	2
1.2	Contributions	3
1.3	Thesis outline	3
2	Related work	5
2.1	Appearance modeling	6
2.1.1	The plenoptic function and light fields	6
	Plenoptic measurement	7
2.1.2	Matting and environment matting	8
	Matting for composition	8
	Environment matting for transparent and reflective objects	8
	Environment matting extensions	10
	Environment matting extended to multiple viewpoints	10
2.2	Stereo reconstruction of Lambertian scenes	10
2.2.1	Basic stereo reconstruction	11
	Dense stereo vs. feature-based and sparse reconstruction	13
	Global vs local/window disparity	13
2.2.2	Matching cost determination	14
2.2.3	Cost Aggregation	15

2.2.4	Computation and optimization of the disparity	15
2.3	Reconstruction of opaque non-Lambertian scenes	16
2.3.1	Stereo reconstruction	16
2.3.2	Shape from reflection	17
2.3.3	Shape from polarization	18
2.3.4	Laser rangefinders	18
2.4	Reconstruction of transparent media	19
2.4.1	Computerized Tomography	19
2.4.2	Multi-view reconstruction with transparency	19
2.4.3	Shape from distortion	20
2.5	Reconstruction of water	20
2.5.1	Reconstruction using transparency	21
2.5.2	Reconstruction using light reflection	21
	Shape from shading	21
	Shape from polarization	22
2.5.3	Shape from refraction	23
	Shape from refractive distortion	23
	Shape from refractive irradiance	23
	Laser rangefinders	24
2.6	Simulation of water	26
2.7	Summary	27
3	Imaged-based reconstruction of Water	29
3.1	Imaging water	30
3.1.1	Physical and optical properties of water	30
3.1.2	Imaging of water surfaces	31
3.2	The geometry of stereo water surface reconstruction	37
3.2.1	Deriving the surface normal from the incident and refracted rays	37

3.2.2	The geometry of indirect stereo triangulation	40
3.3	Practical water surface reconstruction	44
3.3.1	Pattern specification for feature localization and correspondence .	47
3.3.2	Implementation of indirect stereo triangulation	48
3.3.3	The algorithm	50
4	Results	54
4.1	Apparatus and Physical Setup	54
4.1.1	Apparatus and imaging system	55
4.1.2	Camera calibration	55
4.2	Water surface reconstruction simulation	58
4.2.1	Simulation implementation	58
4.2.2	Error metric analysis	64
4.2.3	Feature localization error and calibration error comparison	65
4.2.4	Analysis of localization error at varying depths	65
4.2.5	Simulation data compared to real world data	70
4.3	Water surface sequences	74
5	Conclusion	87
5.1	Future Work	88
A	Simulation algorithm	90
	Bibliography	92

List of Figures

2.1	Environment matting setup	9
2.2	Stereo disparity	12
2.3	The general shape from refraction setup	25
2.4	Two frames from water simulation results [26]	27
3.1	Refraction of light at the air-water interface	31
3.2	Imaging of points beneath water surface	32
3.3	Solutions space of (normal, depth) pairs	34
3.4	Stereo Imaging constrains the normal	35
3.5	Stereo Imaging in 3D	36
3.6	Determining the surface normal	38
3.7	Indirect stereo triangulation	43
3.8	Improved Error Metric	45
3.9	Interpolation for hypothesis verification in 2D	49
3.10	Bilinear interpolation of refractive disparities	51
4.1	Physical setup and apparatus	56
4.2	Trade-off between baseline length and reconstructable region size	57
4.3	Calibration error	60
4.4	Localization error	61
4.5	Reconstruction gauges	62

4.6	Error metric analysis	66
4.7	Simulation graphs for varying calibration and localization errors	67
4.8	Simulation graphs for varying calibration and localization errors	68
4.9	Simulation graphs for varying calibration and localization errors	69
4.10	Simulation graphs for varying depths and localization errors	71
4.11	Simulation graphs for varying depths and localization errors	72
4.12	Simulation graphs for varying depths and localization errors	73
4.13	Simulation results compared to empirical results	75
4.14	Frame of sequence POUR-A	77
4.15	Frame of sequence POUR-A	78
4.16	Frame of sequence POUR-A	79
4.17	Frame of sequence POUR-A	80
4.18	Frame of sequence POUR-B	81
4.19	Frame of sequence POUR-B	82
4.20	Frame of sequence POUR-B	83
4.21	Frame of sequence POUR-B	84
4.22	Pattern distorted by splash	85
4.23	Reconstruction from RIPPLE sequence	86

Chapter 1

Introduction

*“The world turns softly
Not to spill its lakes and rivers,
The water is held in its arms
And the sky is held in the water.
What is water,
That pours silver,
And can hold the sky?”*

-Hilda Conkling

Water has fascinated mankind since the earliest times. It is more than just a necessity of life; water has inspired art, poetry, myth and science. Thales the ancient Greek philosopher described water as the primary principle, or the foundation of all matter. The polymorphism of water and its optical magnificence demands awe and often trepidation upon the high seas. It inspired the ancient Greek god Poseidon, ruler of the seas lending him the ability to change shape at will.

This thesis engages the problem of capturing the shape of dynamic water from images. We present a method for finding points on water surfaces using images from a stereo

camera rig. Our system is able to generate sequences of captured surface geometry of flowing water.

1.1 Motivation

The goal of producing realistic imagery of water was long been sought after by the computer graphics community [60, 46, 29, 28, 25]. Work in this area has taken the form of simulating the flow of water by modeling approximations of the physical laws that govern fluids. At best these are approximations and often subtle water surface effects are missing.

The approach taken by our work and those in the computer vision community as well as those in oceanography has been to extract the shape of water surfaces from images of water [69, 78, 41, 56, 42]. This previous work has sought to take advantage of water’s optical properties to reconstruct a surface. Techniques that have used water’s reflectivity to reconstruct the surface have had less success than those that utilize the refractive property of water [42]. Most of these refraction methods use a single viewpoint and assume an orthographic projection [56, 47, 23]. This requires a relatively distant camera to minimize the projection distortion. Our work, in contrast, utilizes a refractive approach with stereo cameras. We thus avoid some of the assumptions and inaccuracies of these previous methods. Our solution also requires no special sensors or equipment such as laser rangefinders or external lenses.

Although the motivation for this reconstruction in the oceanography community is often for analysis of wind-driven waves, we are also motivated by the possibilities of using this data to obtain or enhance the appearance of novel computer generated images of water. We expect that our work could contribute to any of the following applications:

- The capture of liquid phenomena for composition into animation or film footage.
- Creation of a library of liquid effects allowing the generation of arbitrary liquid

flows out from a composition of the library effects.

- Oceanographic studies of wind-driven waves.
- For precise measurement of transparent objects for Engineering.
- As a first step toward determination of internal fluid flow.

1.2 Contributions

Here is a summary of the primary contributions of this thesis:

- We provide a review of image and sensor-based reconstruction techniques for specular, transparent and refractive objects. We build up this by examining the viability of using these techniques for water surface reconstruction.
- We present a novel reconstruction algorithm for refractive liquids that combines stereo reconstruction with a shape from refractive distortion approach.
- We present and analyze two metrics for testing the validity of surface points on refractive surfaces within the context of a multi-view system.
- We propose an experimental configuration for our algorithm. We present and discuss the results achieved from this setup and compare them to a simulation of our algorithm.

1.3 Thesis outline

This thesis is organized into five chapters. Following this introduction, in Chapter 2, we present background on reconstruction techniques in computer vision. We review stereo reconstruction methods followed by techniques for determining the shape of transparent,

shiny and refractive objects. We also examine previous techniques for reconstructing water surfaces. A review of water simulation is also included.

In Chapter 3 we present our algorithm for reconstructing water surfaces. We discuss issues involved in the implementation as well as our solutions and the resulting implications.

We present our results in Chapter 4. We provide details on the experimental setup and imaging procedure. This is followed by a description of our experimental simulation algorithm. We present and discuss the simulation results, comparing them to experimental results. Finally we present results from reconstructed sequences of dynamic water.

The thesis is concluded in Chapter 5 where we discuss the implications of our work and future avenues of research that build upon this foundation. We include additional algorithmic details in the Appendix.

Chapter 2

Related work

“Let us have wine and women, mirth and laughter, sermons and water the day after.”

-Lord Byron

The complex dynamics of water present a major challenge when attempting to capture the behaviour and appearance of water in a virtual environment. Two primary approaches have been explored to reach this end. The first approach involves the simulation of both the hydrodynamics and the optical properties of water in order to generate a virtual model of water. Rendering techniques have been developed to produce realistic images from these models. The second approach attempts to interpret and exploit images or sensory data of water in order to infer the physical shape and behaviour of water, thus allowing the creation of new images.

In fact the necessity of shape inference in this second path is not clear. Therefore we initially present techniques that attempt to model phenomena directly from images and then transition to techniques that infer the geometric shape. We examine reconstruction methods that reconstruct scenes with simple lighting models proceeding to more complicated systems that handle specular reflections and transparency. We then discuss how effective each technique is for water surface reconstruction.

Subsequently we present research that takes the approach of simulating liquid phe-

nomena and how improved physical models have enhanced the accuracy of these techniques.

We conclude the chapter with a summary of the major obstacles and short-comings of the current techniques for generating virtual models and imagery of water.

2.1 Appearance modeling

Appearance modeling seeks to generate novel views of scenes without attempting to infer shape information from the scene itself. The focus is to capture the appearance of the scene through images and then produce novel views of the scene from either a new viewpoint, or by modifying another aspect of the scene, such as the background. We will examine a number of techniques that follow this process.

2.1.1 The plenoptic function and light fields

One key concept in appearance modeling is the plenoptic function. It is a function that fully describes all the light rays converging at a particular point from every direction [1]. The plenoptic function is directionally parameterized by spherical coordinates θ and ϕ . The light intensity of the rays is also dependent on wavelength (λ). Three more parameters specify the location of the point in space (V_x, V_y, V_z), and a temporal parameter (t) can also be included when measuring a temporal sequence. Here is the full description of the plenoptic function P :

$$P = P(\theta, \phi, \lambda, t, V_x, V_y, V_z) \quad (2.1)$$

Typically, a camera view of a scene captures a pencil of rays converging on the centre of projection of the camera. If the plenoptic function were to be known for every point in a scene, then it would be possible to view the scene from any position and angle. Knowing the plenoptic function at every point allows us to compute the plenoptic function

directionally parameterized by the camera's field of view and located at the camera's centre of projection.

Plenoptic measurement

One branch of research in computer vision has developed around utilizing the idea of the plenoptic function sampling for recreating novel views of a scene. This work leverages on the idea that the plenoptic function is redundant in 'free-space', where there are no occluding objects. In other words, a ray through a scene has the same intensity at every point as long as it does not strike any occluding object. Thus any light ray in a scene can be parameterized by two points on two parallel planes, rather than the five parameters described above [48, 34].

Images are used to sample the light rays converging on the centre of projection of a camera. The CCD elements of the camera record the light intensity converging from a particular directional footprint, rather than individual rays. Thus the image pixels represent the average intensity of bundles of rays. So measurement of the plenoptic function starts with many images or samples of the scene. Then new views of the scene are generated by interpolating between sampled light rays collected from the set of images. Interestingly, Chai et al. have shown that fewer images are necessary when some geometric properties of the scene are known [20].

Sampling the plenoptic function for water is especially problematic. Water is not static, so the plenoptic function may change at each time instant. This means that sampling must be done instantaneously from all expected angles. Water's optical properties present another challenge, as its appearance is predominantly a reflection or refraction of light emitted from the rest of the scene. This means the plenoptic sampling may need to be performed within the desired scene, rather than a controlled laboratory environment. The reflective and transmissive properties of water may also cause elements of the sampling rig to appear in the images.

2.1.2 Matting and environment matting

Matting for composition

Matting is a technique for separating the background from the foreground in images, usually so that the foreground can be composited over a new background. Typically a *matte* is formed that is opaque over the background, partially transparent at the edges of the foreground and fully transparent over the rest of the foreground.

Environment matting for transparent and reflective objects

Matting can be used to approximate the appearance of transparent objects by blending the matte with the background in those areas that are transparent. This technique breaks down when the foreground object significantly refracts or reflects light, since a direct blend with the background is insufficient to describe the distortion actually occurring. A technique called *environment matting* attempts to resolve these issues by determining what background footprint best maps to a particular pixel in the refracting or reflecting foreground object [81]. The end result is a function for every foreground pixel that includes the traditional matte, as well as the contribution of light from refraction or reflection of the surrounding environment.

The general approach is to take a series of images of the foreground object with structured textures on screens surrounding the object. The texture set consists of a hierarchy of vertical and horizontal stripes of varying thickness and are used to determine the best axis aligned rectangular region whose average pixel value maps through a particular foreground pixel. This rectangular region is computed by optimizing over the set of images that were collected with the set of environment textures.

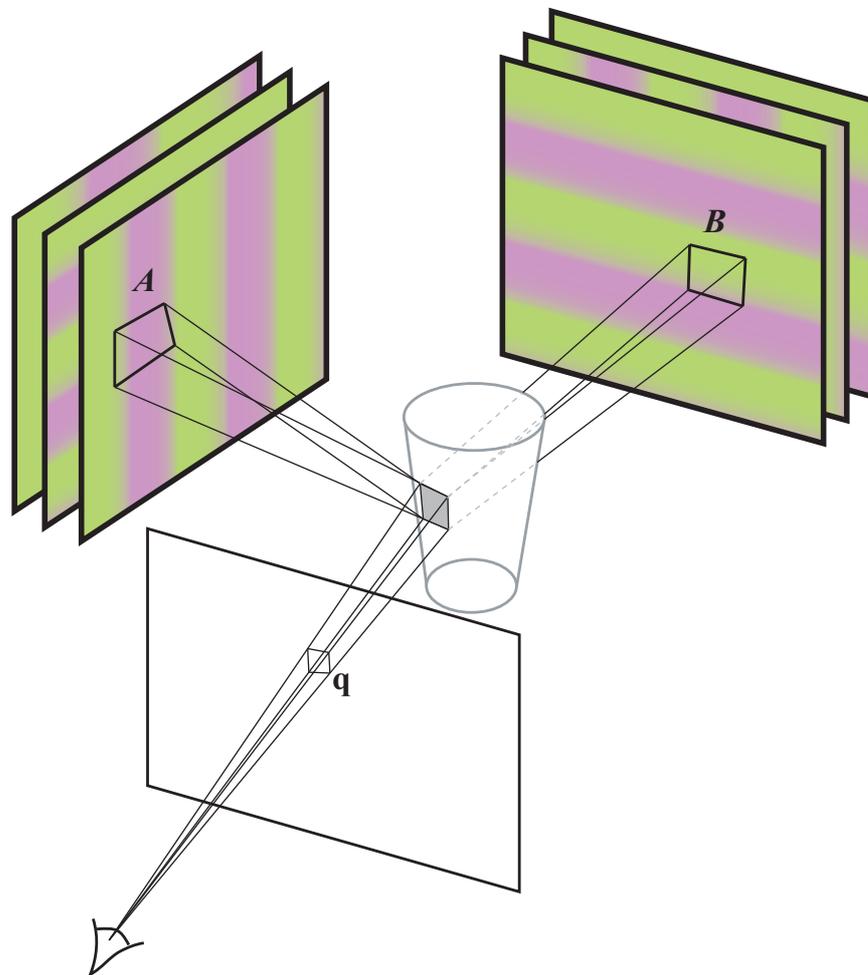


Figure 2.1: Environment matting setup. For a given pixel \mathbf{q} that is part of the image of the object, the colour of the pixel is composed of a reflected region of the pattern on the side A and a refracted region of the background pattern B . Several images are taken with varying stripe thickness and orientations for the patterns.

Environment matting extensions

The original technique for environment matting requires static objects, since multiple images must be captured to determine the background mapping as described above. This limitation can be overcome by simplifying the model and capture setup [21]. This is achieved by capturing the object in a darkened room, with a single colour gradient map background. A background smoothness constraint is used to reduce the complexity and the lack of ambient light allows the foreground colouring to be discarded. Given these assumptions, sequences of dynamic refractive objects, such as water pouring into a glass, can be captured and matted against arbitrary backgrounds. Unfortunately this reconstruction is only suitable for a single viewpoint.

Environment matting extended to multiple viewpoints

Another major restriction of the environment matting technique is its fixed viewpoint. Several methods that capture the reflectance field of objects from multiple viewpoints have been proposed [52, 24]. One of these focuses on reconstructing transparent objects using an extension of the environment matting technique. They utilize a rotating camera and lighting rig that captures the visual hull of objects as well as the environment matte from multiple viewpoints. A form of unstructured light field interpolation is used to determine the lighting for every visible point of the object. Although this technique is only suitable for static objects, it effectively captures many optical effects not previously attempted. Since objects are captured from multiple viewpoints, the data is much more useful in terms of animation and visualization.

2.2 Stereo reconstruction of Lambertian scenes

Appearance modeling avoids making inferences about the shape of the scene by relying on large numbers of images of the scene to generate novel views. By extracting information

about the scene geometry, fewer images are needed to construct new views. This is especially important for reconstructing dynamic phenomena such as water.

Before we can examine the complex case of water, we will first look at some of the background of stereo work for simple Lambertian scenes. We will cover previous work on stereo reconstruction techniques that extract geometric scene information from binocular parallax. We then outline several different approaches to the problem and the issues involved with each.

2.2.1 Basic stereo reconstruction

Stereo reconstruction is one of the most common techniques for determining geometric information about a scene. It is derived from the human visual system, and works by leveraging the parallax between corresponding points in two views of the scene (see Figure 2.2). The relative displacement of the corresponding points in the two views is known as their *disparity*. Conventional stereo vision determines the depth (z) of the point from the *stereo baseline* or the line connecting the centre of projection of both views [59]. The formulation for the depth given an binocular view with parallel optical axes is:

$$z = \frac{BF}{d}, \quad (2.2)$$

where B is the length of the stereo baseline, F is the focal length of the cameras and d is the disparity between the images of the point.

Most stereo algorithms have at least a subset of the following stages:

- **Matching cost determination** - Determination of point correspondences between views and the assignment of a cost to each candidate
- **Cost aggregation** - Aggregation of the costs of all points
- **Computation and optimization of the disparity**

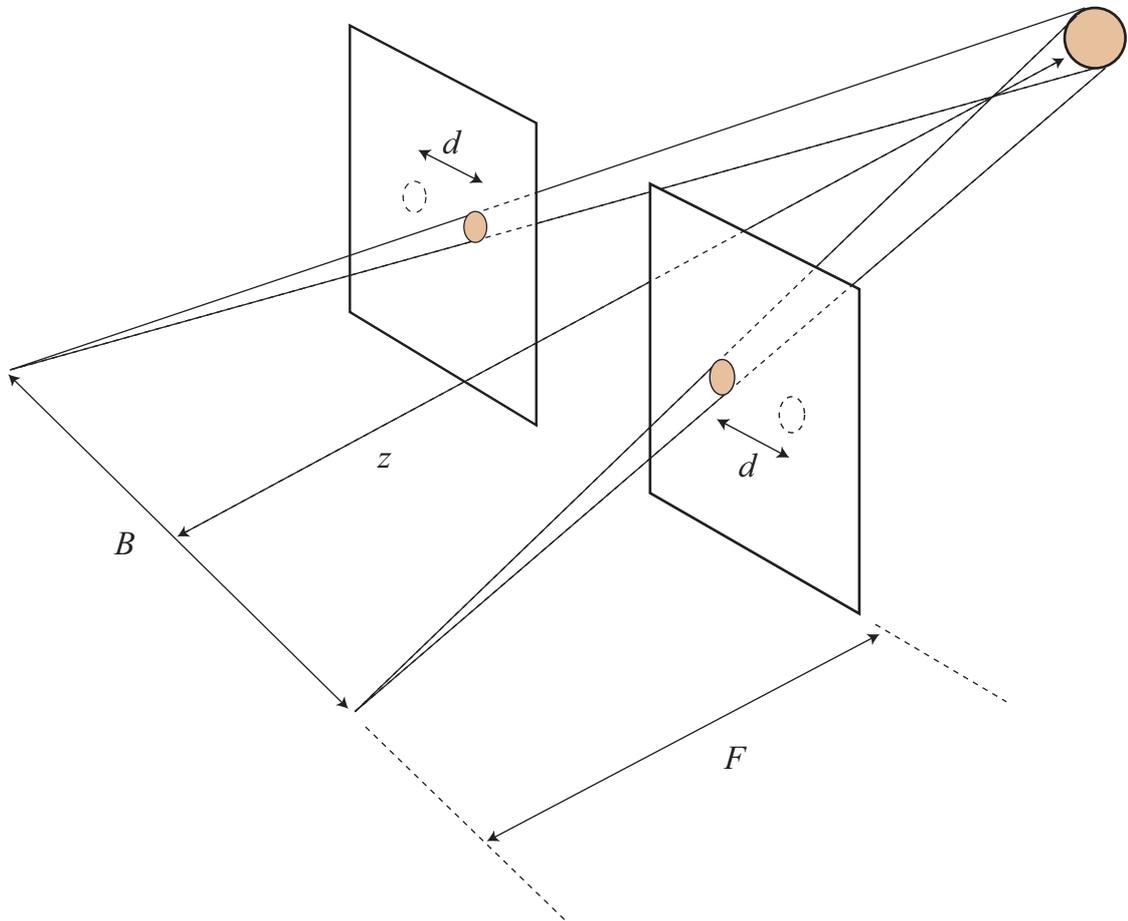


Figure 2.2: The figure shows two cameras separated by a baseline B . Both cameras image an object at depth z away from the baseline on their image planes. Both cameras have a focal depth of F . The dotted circles on the image planes indicate the image location of the object in the other view. The disparity of the image of the object between the two views is d .

The rest of this section will discuss various techniques for solving parts of the stereo reconstruction problem as well as some of the difficulties faced in stereo reconstruction. We examine whether stereo is appropriate for extracting geometric shape from images of water surfaces and how the challenges of conventional stereo approaches apply to our problem area. An extensive review of dense, binocular stereo algorithms can be found in [66].

Dense stereo vs. feature-based and sparse reconstruction

Stereo reconstruction techniques can be divided into dense and sparse point matching methods. Dense methods attempt to find a correspondence between every pixel in the stereo images [27, 14]. Often optical flow or incremental algorithms are used in this case and the displacement between corresponding pixels constitutes the stereo disparity [13].

Sparse stereo methods often rely upon feature matching between images such as edges or corners that can be accurately localized [54, 49, 79]. Correspondences between image features in the images are then determined.

Although water surfaces are smooth and relatively featureless, sparse techniques can be used for indirect stereo reconstruction. The water's reflectivity or refractivity can be utilized to redirect a sparse pattern that can be used in turn for reconstruction purposes.

Global vs local/window disparity

Most techniques can also be divided into global or local reconstruction algorithms. Global techniques typically compute disparity values for every point and then minimize an energy function based on the sum of costs for every point along with a smoothness term. Local or window approaches attempt to optimize each point separately by aggregating the cost of the neighbourhood or support region around the point, given some disparity estimate.

2.2.2 Matching cost determination

The most basic methods for matching pixels between stereo views measure the squared [59, 37] or absolute difference between pixels [44]. In order to reduce the impact of mismatched pixels, several techniques have been developed. These include robust estimators using truncated quadratics and contaminated Gaussians that help to eliminate outliers [13, 14]. Another matching technique is normalized cross-correlation that is similar to the sum of squared differences but also normalizes the matching window before comparison.

Intensity gradients are sometimes used for matching, having the benefit of being insensitive to camera bias and gain. Often these camera artefacts are removed in a pre-processing stage [22]. Sparse reconstruction techniques sometimes use a binary matching technique when seeking to match detected edges or other features [4, 35, 19].

An important innovation suggested by Birchfield and Tomasi is to match pixels in one image with interpolated sub-pixel offsets in the other image, rather than merely seeking matches at integral offsets [12]. Matching can be especially problematic when there are objects with repeated textures or edges. This can easily lead to mismatches, although reconstruction with multiple views can help to alleviate this [59].

Stereo reconstruction on water surfaces has some apparent advantages over general scenes. Typically a water surface is smooth and exhibits few occlusions when viewed from overhead as long as splashes are discounted. Thus many of the matching cost techniques for handling discontinuities are unnecessary. On the other hand, stereo matching with water is non-trivial due to its specular reflectance and its refractive nature. Matching would have to be performed indirectly using either a reflection or a refracted image which may be discontinuous or warped by the water surface.

2.2.3 Cost Aggregation

Stereo reconstruction techniques have used a wide range of support regions to enhance matching. Two-dimensional support regions can be aggregated over square windows [59], shiftable windows [16, 2], and windows with adaptive size [58, 45]. Three-dimensional aggregation techniques attempt to match surfaces with areas of similar disparity or a similar disparity gradient [61, 62]. This permits sloping surfaces to be more accurately detected.

Aggregation on fixed windows can be performed by convolution or box filters. Another method that is used is *iterative-diffusion*, where the weighted cost of neighbouring pixels are added to the local pixel [65, 68].

2.2.4 Computation and optimization of the disparity

Local methods typically just take the disparity associated with the minimal cost as determined by the aggregation stage. This has the problem that points in the reference image may not have a one-to-one mapping to points in the second image.

Global methods tend to concentrate on this stage. They typically minimize an energy function as follows:

$$E(d) = E_{data}(d) + \lambda E_{smooth}(d) \quad (2.3)$$

The data function, $E_{data}(d)$, measures how the disparity function d matches the reference image to the second image using some aggregate matching cost function. The smoothness term $E_{smooth}(d)$ measures the energy associated with smoothness or discontinuity in the support region around the point. Previous work has focussed on robust smoothing functions that handle smooth surfaces as well as discontinuities [73, 14, 65, 33]. Colour and intensity discontinuities have also been used to predict surface discontinuity [18, 30, 16].

Another area of research has looked at how best to minimize the energy function defined above. Some traditional energy minimization routines are continuation [15], simulated annealing [33, 50, 6], highest confidence first [19] and mean-field annealing [31].

Another class of global optimization algorithms use dynamic programming to minimize Equation (2.3) on a scanline basis by finding the minimum-cost path through a matrix of matching costs of pixels in the two corresponding scanlines [8, 7, 32, 22, 16, 12].

2.3 Reconstruction of opaque non-Lambertian scenes

Many reconstruction algorithms make the implicit or explicit assumption of view independent lighting, or a Lambertian shading model. This assumption breaks down for shiny or specularly reflective surfaces. This is particularly relevant to water reconstruction, since water surfaces are highly specular.

In this section we examine a variety of reconstruction methods that attempt to reconstruct shape in non-Lambertian, opaque scenes. We look at several stereo-based techniques that model general non-Lambertian scenes. We then examine methods that focus on purely specular or mirroring surfaces. The first technique of this type infers shape from distortions in the reflected images of curved specular surfaces. This is followed by a description of a voxel-based technique for mirroring objects. Then we discuss how polarization sensors and laser rangefinders may be used to determine the shape of reflective objects.

2.3.1 Stereo reconstruction

Recently there has been a concerted focus on reconstruction of specular surfaces with stereo [10, 11]. One approach is to remove specular highlights in a pre-processing stage before reconstructing as before [57]. Another technique leverages Helmholtz reciprocity

to capture the shape of objects with arbitrary reflectance properties. An image pair of the object is taken with a reciprocating camera and light. This guarantees that the pixel intensities in both images of corresponding points on the object depend only on the surface shape and not on the object's reflectance properties [80].

More recently Treuille et al. also proposed a method for capturing the shape of objects with general reflectance properties [74]. Their technique avoids the reciprocity constraint of the camera and light set up. Instead they rely on observations of the target object along with a known example object that exhibits the same reflectance properties. They use the known normals and observations of the reference object to determine orientation consistency in the target object. In addition they describe a technique for handling self-shadowing on the objects.

2.3.2 Shape from reflection

Another technique that has been applied to determine the shape of objects that mirror light has been to leverage distortion and non-linearity that occurs during this redirection. Although reflected images project linearly across flat mirrors, distortions in the mirror will in turn distort the reflected image allowing shape inference of the mirror surface.

Curved specular surfaces have been reconstructed by inferring shape from the distortion of lines and line intersections [64]. A formula was derived to determine the tangent normal of the specular object for a calibrated point defined by the intersection of two lines. The formula utilized the curvature of the images of the intersecting lines to determine the surface normal and then the surface location at that point. Other methods have used the distortion of patterns to infer surface slope [36].

Another approach to reconstructing purely specular surfaces is to model the surface by localizing features or a pattern in the reflected image. One technique that seeks to do this uses a multi-view voxel carving technique with a normal consistency check [17]. The technique reconstructs mirror-like surfaces, discretizing the space around the surface into

voxels. Next, each voxel is assigned a normal from each camera view that would account for the reflected image had the specular surface passed through that voxel. Voxels which have inconsistent normal sets are then eliminated, leaving the voxels that best represented the true surface.

2.3.3 Shape from polarization

When light reflects off of a surface, some of the light becomes polarized in the direction of the surface normal. The phase image of the object encodes the orientation of the reflection plane which is defined as the plane spanned by the surface normal and the incident ray.

Several methods exist for determining the surface normal once the reflection plane is determined. One technique is to use a second view to constrain the normal to an epipolar line and then use a global minimization approach to solve the surface normals as well as depths [63]. Another approach assumes surface smoothness and the normal at object boundaries is perpendicular to the viewing angle. Once the normal is determined at these edge points, degree of polarization images are used to propagate the solutions over the rest of the object [53].

2.3.4 Laser rangefinders

An alternative method to determining shape of objects from images is to use laser rangefinders. These typically work by projecting laser light onto the object surface and measuring this reflected light at a known receiver. The process accurately triangulates points on the object surface but usually requires a Lambertian surface. Recently, laser rangefinders have been developed that are able to effectively reconstruct the shape of specular objects as well [3]. This is done by restricting the angle of the incident light to a single direction by attaching several parallel plates at an angle in front of the CCD elements. The vertical plates, along with a horizontal slit block incident light except from the one

expected angle, allowing surface triangulation. Clearly this technique would not be suitable for scanning fluctuating water, yet it is a certainly an advance for reconstructing static specular surfaces.

2.4 Reconstruction of transparent media

For many years, transparency reconstruction has been common in medical imaging systems. These approaches are meant for purely transparent scenes and do not deal well with occlusions. Recently methods have been developed to integrate common computer vision techniques with scenes containing opaque and transparent objects. Instead of treating transparent objects as an obstacle, another approach has been to utilize the refractive properties of transparent objects as a means to reconstruct the surface of the objects.

2.4.1 Computerized Tomography

Transparent media have long been reconstructed with medical imaging systems using Computerized Tomography. CT techniques use density images to reconstruct slices of the structure of a volumetric object [43]. Each image records the density of the object within the each projected pixel cone. This information is all compiled together and then interpreted to produce a density map of the transparent object. One of the primary methods for this compilation is called back-propagation [43].

2.4.2 Multi-view reconstruction with transparency

Voxel carving techniques have been applied to transparent objects. One such volumetric carving technique seeks to deal with transparent objects through a modified version of voxel carving. For each ray through each pixel the voxels along the ray are assigned weights that govern how much that particular voxel contributes to the pixel colour. The weights translate to transparency values. The algorithm uses an iterative approach to

find the most consistent set of voxels and weights given all the views of the object. These weighted voxels are known as ‘Roxels’. In this technique, uncertainty is modelled by transparency, so if the precise location of a surface edge is uncertain it will appear to be blurry.

Tsin et al. provide a method for handling stereo reconstruction in the presence of translucency and reflections [75]. They describe how a scene can be reconstructed with multiple layers under these conditions when computing depth. So reflected objects are assigned a depth layer as well as the reflector. The work also describes a method of extracting the correct colours of the component layers.

2.4.3 Shape from distortion

Shape from distortion techniques can also be applied to transparent objects by inferring surface shape of a refractive object from the distortion it causes to light transmitted through it.

One recent method has been presented for inferring shape and pose of transparent objects from a moving camera’s image sequence [9]. Features are tracked throughout the sequence and an objective function that characterizes the shape and pose of the transparent object is minimized. The work restricts the target objects to be parameterized by a single parameter such as super-quadrics. This is a clear step forward in reconstructing shape from transparent media, although the low-dimensional parameterizations reduce the generality of the method and makes it inappropriate for dynamic transparent objects such as water.

2.5 Reconstruction of water

In order to reconstruct water surfaces, its optical properties must be exploited to infer the surface shape. When a light ray strikes a water surface from air, part of it is mirrored and

reflects off of the surface. The rest of the light is refracted and transmitted through the water. Water can be considered as a transparent, reflective or refractive object, leading to a multitude of reconstruction approaches. In this section we examine the feasibility of techniques that attempt to reconstruct water in each of these ways.

2.5.1 Reconstruction using transparency

Most of the methods for reconstructing transparent surfaces break down when they are applied to water in a similar way to plenoptic sampling. CT techniques would require many simultaneous images of the water and it would be difficult to avoid imaging the capture equipment at the same time. The imaging technique also presents a problem. Neither direct optical images nor ultrasound will work due to the surface refraction. Magnetic resonance is also unusable due to the slow rate of capture. The Roxel algorithm has the same problems as CT techniques since it does not consider refraction and must be simultaneously imaged from multiple views.

2.5.2 Reconstruction using light reflection

Shape from shading

Shape from shading techniques attempt to infer geometric shape from the shading of a surface given some expected or known reflectance and lighting model [40]. If the object's reflectance properties are known and the light source location is known, then the surface shading depends only on the surface normal. Thus from an image of the object, it is possible to infer surface normals from the pixel intensities.

Traditional shape from shading algorithms assume a Lambertian reflectance model as it is isotropic and the shading is independent of the viewing angle. Reconstructing purely specular surfaces, such as water, presents several challenges. Using a single point light source is often insufficient, as the surface will only receive a highlight where the

viewing angle and the light incident angle on the surface are equal. With a point light source only part of surface will be lit, where the surface normal is such that the incident light reflects directly toward the camera.

Several attempts have been made to reconstruct water surfaces using reflected light. One approach utilizes stereo images taken under natural lighting conditions and then uses traditional stereo image matching for Lambertian surfaces [69]. The resolution of the reconstruction appears to be insufficient for the determination of small wavelength waves. The second problem is that of correspondence error resulting from specular bias between the binocular views. Other specular artefacts plague waves with limited amplitude.

Another technique directly uses the specular highlight falloff to compute shape [67]. Several images of the surface from different orientations are used to determine surface slope at various points on the surface. Once these seed slopes are found, solutions are grown around these points by searching for the best surface orientation that minimizes the difference between the expected irradiance given that orientation and the observed irradiance.

Reconstruction of water surfaces typically do not have some of the common problems of occlusions or discontinuities found in many reconstruction scenarios except when splashing occurs. There is high degree of non-linearity when determining surface slope from irradiance due to the transparent nature of water [42]. Reflectivity on the water surface is governed by Fresnel's coefficients, causing a high degree of reflection at grazing angles but very little at acute angles. Also a very large light source is required for reconstruction at grazing angles.

Shape from polarization

Shape from polarization algorithms typically cannot handle internal reflections although some predict general internal reflections and reduce the polarization images accordingly. This is only an approximate solution and error is still accumulated from the

inter-reflections.

Theoretically, a multi-view or binocular stereo shape from polarization approach could reconstruct water, although according to our knowledge this has not been attempted. Polarization methods tend to deviate from our motivation to design a simpler, pure image-based system for accurate water surface reconstruction.

2.5.3 Shape from refraction

Water reconstruction techniques that have treated water as a refractive medium have produced the most promising results and avenues of research. Determining shape from refraction techniques avoid many of the problems associated with shape from reflection algorithms. Refraction non-linearities are much lower than those for reflection, allowing a smaller light source or pattern and since most refraction techniques light the surface from below, specular artefacts do not occur.

Shape from refractive distortion

Water surfaces have also been reconstructed through refractive distortion [56]. One algorithm for reconstruction has four parts: First optical flow is computed on the image of the pattern as it is distorted by the water. Then the average of the optical flow displacements is taken to be the true location of a particular pattern point. Then the surface normal for every point in every frame is computed given the displacement from the computed ‘true’ location. Finally a surface is integrated from the surface normals. This technique assumes a distant camera and only works on low amplitude waves and the surface is reconstructed up to some unknown scale factor.

Shape from refractive irradiance

Several image intensity based techniques for recovering surface shape from transparent media using refraction have been presented in the past [78, 41, 47, 23]. Most techniques

have been designed to determine the slope of water surfaces. The classic imaging setup is show below in Figure 2.3.

Light rays from a screen pass through the lens that collimates them so that certain intensities or colours correspond to parallel light ray columns and are then refracted by the water surface to the distant camera. This has the result of associating colour or intensity with particular surface slopes.

There are several techniques for generating the screen, some using an attenuated light source from one end, some using an HSV coloured gradient and others just a lit monotone intensity gradient [78].

An important assumption in all these techniques is that of an infinitely distant camera. This is to assure parallel incoming rays from the water surface. Yet distortions are still going to affect results as this assumption cannot be modeled precisely. Also error is bound to be introduced by the collimating lens. Light attenuation from the water will also affect the slope intensities differently in different parts of the image as the underwater path lengths will vary.

Laser rangefinders

Laser rangefinders have been developed to measure water surfaces typically by projecting a laser ray through the water and measuring the ray's deflection due to refraction. This has been done both by firing the ray from beneath the surface and detecting it's projection on a screen above the water [77], or the reverse where the deflected ray projects on to a screen beneath the water. Geometrically the surface normal can be determined by the detection of the refracted ray, and an iterative method can be used to determine the water surface intersection point.

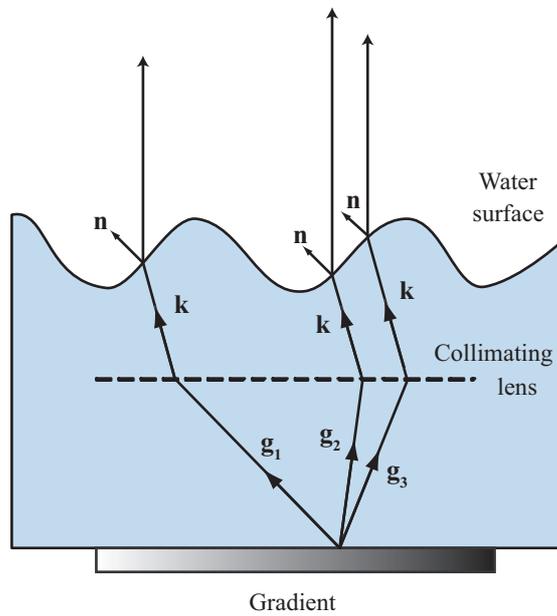
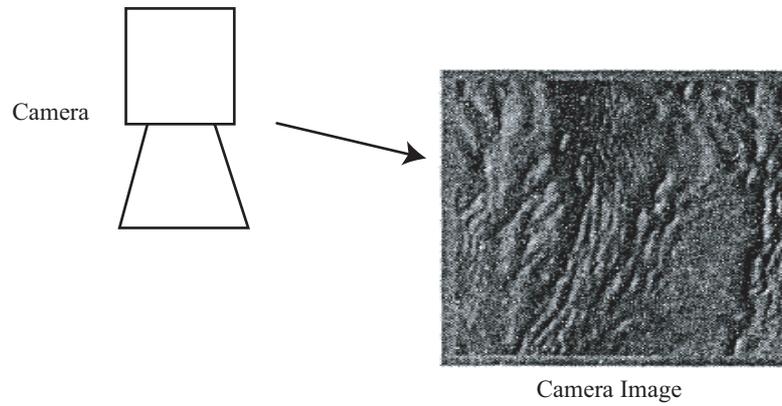


Figure 2.3: Rays of light (\mathbf{g}_i) from a point on the gradient radiate out and are collimated by the lens into a common direction (\mathbf{k}). These rays strike the water surface and only one certain surface normal (\mathbf{n}) will refract them toward the distant camera. Thus in the camera's image, pixel colours correspond to surface normals.

2.6 Simulation of water

Early work in fluid simulation typically focused on wave generation and used simple hydrodynamic models for sinusoidal waves [60, 51]. Splines were used to simulate wave refraction [76]. Detailed fluid pressure and viscous effects were largely ignored or approximated by particle systems for splashing or breaking waves.

The progression in fluid simulation in computer graphics has been to more closely approximate physical models and the result has been increased realism. Fluid advection and pressure flow are governed by the Navier-Stokes equations and many papers have attempted to approximate these non-linear equations to capture the desired realism. Some early attempts, such as the work by Kass and Miller [46], simplified the equations for shallow water and used them to generate animated height fields. This work did not take into account rotational or pressure based effects, preventing the characteristic eddying and swirling effects of fluid. Following this work, Foster and Metaxes [29] utilized work done in the Computation Fluid Dynamics field by Harlow and Welch [38] who described the full characterization of the Navier-Stokes equations. The fluid was discretized into a grid of cubes. The Navier-Stokes equations were then solved explicitly and the fluid advected. Further attempts to improve efficiency and robustness of the system were examined [28]. Stam presented an improvement in his ‘Stable Fluids’ [71] to implicitly solve the system with much larger time steps while still maintaining robustness.

An important aspect of fluid simulation research is to improve the visualization of the fluid effects. Work on liquid surface representation using level sets introduced the most realistic looking examples seen so far. Level sets were combined with particles to allow for splashing [25]. Liquid rendering was then further improved by focusing on accurately representing and rendering the liquid surface using an improved particle and level set approach [26]. Photorealistic results have been produced by such simulations, yet these methods are computationally intense and by nature simplifications of the actual physical processes, potentially losing secondary motion and subtle effects (Figure 2.4).

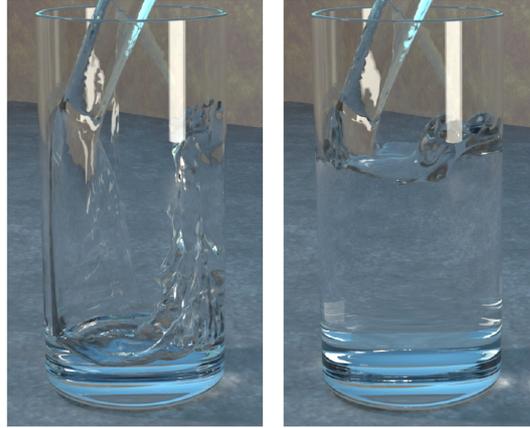


Figure 2.4: Two frames from water simulation results [26]

Significant work has gone into simulating fluids with particle systems, often to simulate waterfalls or other dynamic effects [70]. Recently this work has begun to generate fluid effects at interactive rates. One effective method has been to simulate a liquid with particles but to render the surface using an interpolation method known as Smooth Particle Hydrodynamics to achieve interactive simulation rates [55]. The method computes a Navier-Stokes simulation for each particle and interpolates between particles using a radial basis function to determine the fluid surface.

2.7 Summary

The simulation of water in computer graphics has received a good deal of attention in recent years and impressive images have been developed. Despite this, simulations still rely on simplified physics models and complex phenomena such as breaking waters are difficult to produce.

Water simulation must deal with complex hydrodynamics and surface tension, solving or approximating non-linear partial derivative systems in order to generate believable images and flows. In contrast to this, water capture techniques manipulate the relatively simple optical properties of water to capture the shape of water, without the need for

hydrodynamic models.

The decision for what optical property to use is vital to accurate reconstruction. Techniques that use specular reflection suffer from water's non-linear Fresnel reflection coefficient. This results in very little reflection when viewing a surface perpendicularly but much greater reflection at grazing angles. The inverse is true of refraction. In view of this, it is not surprising that refraction based techniques have been more successful at reconstruction water surfaces.

Although the sensor-based techniques appear to produce effective results, we are more interested in the more accessible image-based approaches. Of the image-based refraction based techniques, shape from refractive irradiance and shape from distortion techniques seem to be the most effective. Despite this success, these techniques often suffer from inaccuracies in their image modelling assumptions, such as a distant orthographic camera and a collimating lens.

These inaccuracies could be improved by combining the refractive reconstruction approach with the well developed stereo techniques seen earlier. In light of this, we propose that a multi-view stereo approach that uses an indirect matching technique similar to the shape from distortion technique in [56] could improve reconstruction accuracy and remove some of the imaging assumptions.

Chapter 3

Imaged-based reconstruction of Water

“Only a fool tests the depth of the water with both feet.”

-African Proverb

In this chapter we discuss the physical properties of water, and how those properties influence our design for a system to reconstruct water surfaces from images. We will present a system that addresses many of the concerns with previous techniques outlined in the last chapter.

Our design attempts to fulfill the following goals:

- Physically-consistent water surface reconstruction,
- Reconstruction of rapid sequences of flowing, shallow water,
- High reconstruction resolution,
- Use of a minimal number of viewpoints and props.

Our work focuses on recreating a precise definition of the water surface from images. We consider the problem of reconstructing internal flow as beyond the scope of this work, although accurate knowledge of the surface can be considered an important first step.

We present a sparse multi-view approach to determine the water surface. Multi-view reconstruction approaches have been used before for water surfaces, but only within the context of shade-from-shading [69]. Instead we propose that stereo, combined with shape-from-distortion, is an effective and accurate approach to the problem, gaining from the benefits of refraction over reflection reconstruction. Previous work has utilized water surface distortion but only viewed from a single camera [56]. Also, our stereo technique does not assume distant, orthographic views of the surface, making our model more physically consistent. Having a stereo system also negates the need to have an extra collimating lens under the water, as used by some previous single camera techniques [78, 41, 47].

We also describe how our system is capable of accurately reconstructing very shallow water.

3.1 Imaging water

3.1.1 Physical and optical properties of water

Light is refracted or bent when there is a density change in the media it is traveling through. The well known Snell's law governs light refraction; its general form is as follows:

$$r_1 \sin \theta_i = r_2 \sin \theta_r \quad (3.1)$$

Where r_1 is the refractive index of the first medium, r_2 is the refractive index of the second and θ_i and θ_r are the incident and refracted angles. At the interface between water and air, there is a significant change in density and light rays are noticeably refracted. We can simplify Snell's law in this case, since the refractive index of air is 1 as in Equation (3.2). It is important to understand that the incident and refracted rays always lie on

a plane, regardless of the surface normal. Thus it is valid to visualize refraction at an interface in two dimensions (Figure 3.1). Snell’s law is written as

$$\sin \theta_i = r_w \sin \theta_r. \quad (3.2)$$

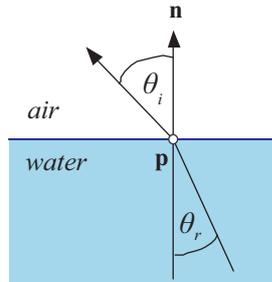


Figure 3.1: A ray is refracted at a surface point between water and air with a surface normal \mathbf{n} .

When light strikes the water-air interface, part of the light is reflected and part is refracted. The ratio of reflected to refracted light increases as the angle of incidence increases. If we continue to increase the incident angle, the refracted angle approaches 90 degrees. At this point we say that the incident angle has reached the critical angle. Any further increase in the incident angle results in total internal reflection, with no light refracted.

Refraction of light also depends on the wavelength of the light. So red light has a higher refractive index than blue light. This property is commonly utilized in prism light dispersion experiments.

3.1.2 Imaging of water surfaces

Water tends to exhibit slight absorption primarily in the green and red spectra, thus resulting in its typical blue hue. It would be possible to determine depth from absorption, but the absorption rates are so low (approximately 0.005 cm^{-1} for red light [72]) that

accurate measurements of shallow water would be difficult with typical imaging equipment. Thus, rather than directly attempting to image water, we examine constraints for indirect surface measurement.

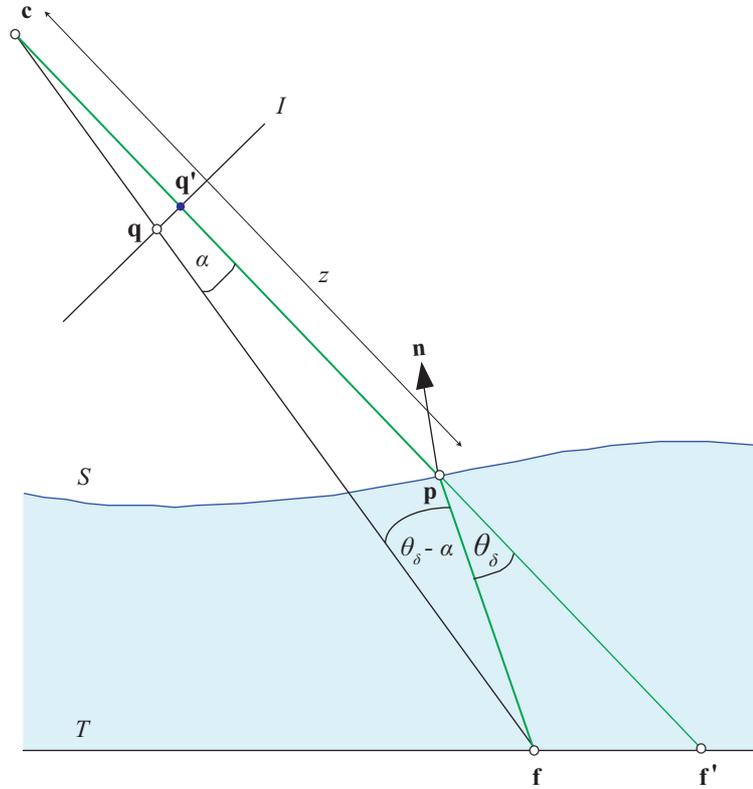


Figure 3.2: Imaging of points beneath the water surface. Feature \mathbf{f} is refracted at point \mathbf{p} toward the camera \mathbf{c} and is imaged on the image plane at \mathbf{q}' . When no water is in the tank, \mathbf{f} is directly imaged at \mathbf{q} . Feature \mathbf{f}' is the projection of the refracted image point \mathbf{q}' .

Consider the imaging setup in Figure 3.2. The figure shows rays traced from points beneath the water surface to an ideal camera, with its centre of projection located at \mathbf{c} . The points are imaged on the image-plane (I) where the rays intersect it (\mathbf{q} and \mathbf{q}'). \mathbf{q} corresponds to the image of the point \mathbf{f} without water and \mathbf{q}' is the image of the point \mathbf{f} with water. We have two unknowns, the distance of the surface point from the camera (z) and the surface normal (\mathbf{n}) that define our solution space. Figure 3.3 shows a

solution space for surface normal, depth pairs $(\mathbf{n}_1, z_1), (\mathbf{n}_2, z_2) \dots (\mathbf{n}_m, z_m)$. Note that for every depth value, we have a unique surface normal that could account for the refractive disparity. As the depth value is increased, the slope of the normal must also increase to compensate until the physical limits of refraction are reached. Depth is computed from the points as follows¹ :

$$z_i = \|\mathbf{p}_i - \mathbf{c}\|. \quad (3.3)$$

The solution space is restricted to surface normal and depth pairs that refract the ray of light coming from \mathbf{f} to the image point \mathbf{q}' . The physical properties constrain this solution, as light cannot be refracted beyond the critical angle. The other restriction is the maximum surface normal.

We also note that the distance to the water is not linearly related to the surface normal as can be seen in Equation (3.4) as a result of the non-linearity of Snell's law (Equation (3.2)). Equation (3.4) relates depth (z) to the angular difference between the incident and refracted rays (θ_δ) as well as the refractive displacement angle (α). This equation results from applying the sine law, given the geometric arrangement of Figure 3.2 as follows,

$$\begin{aligned} \frac{z}{\sin(\theta_\delta - \alpha)} &= \frac{\|\mathbf{f} - \mathbf{c}\|}{\sin(\pi - \theta_\delta)}, \\ z &= \|\mathbf{f} - \mathbf{c}\| \frac{\sin(\theta_\delta - \alpha)}{\sin \theta_\delta}. \end{aligned} \quad (3.4)$$

Bearing in mind that water is a highly dynamic liquid, we are unable to obtain multiple views of the surface from a single camera. So if we consider an imaging setup with a second camera as in Figure 3.4, we can use the second refractive displacement information to triangulate the common surface point and surface normal. Note that

¹In contrast, conventional stereo depth is determined as distance to the projection of the point onto the optical distance.

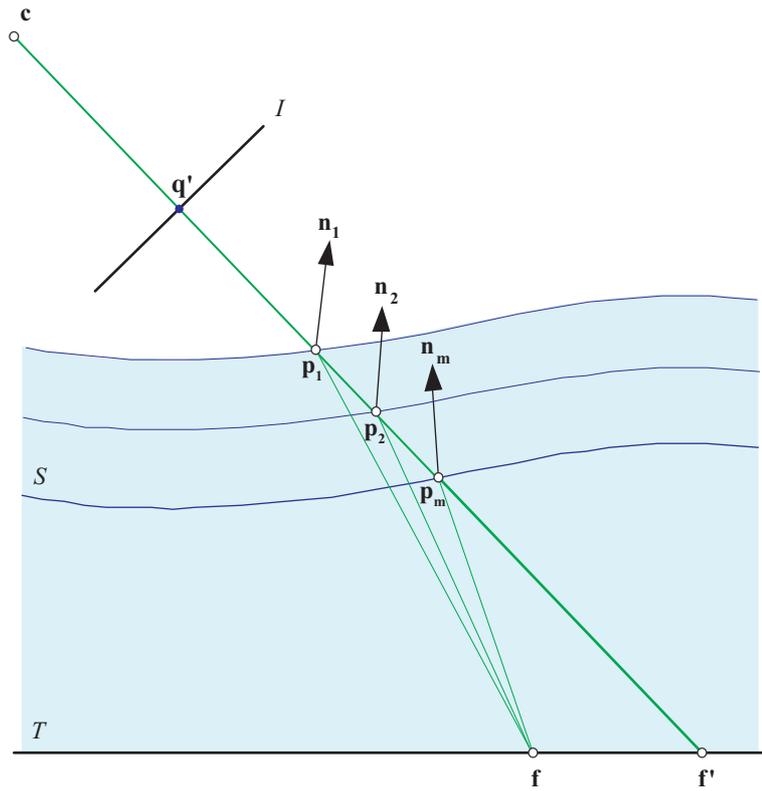


Figure 3.3: The Figure shows how a set of surface points ($\mathbf{p}_1, \mathbf{p}_2 \dots \mathbf{p}_m$) at different depths with corresponding normals ($\mathbf{n}_1, \mathbf{n}_2 \dots \mathbf{n}_m$) could all refract \mathbf{f} to the camera \mathbf{c} through \mathbf{q}' .

although the rays $\mathbf{c}_1\mathbf{p}\mathbf{f}_1$ and $\mathbf{c}_2\mathbf{p}\mathbf{f}_2$ are shown on the same plane, this is not a necessary requirement for our algorithm. In Figure 3.5, we illustrate how the cameras may be oriented to one another in three dimensions. For clarity all further figures are consistently presented in two dimensions even though the rays may not be coplanar. Also note that all points of intersection are marked on the figures.

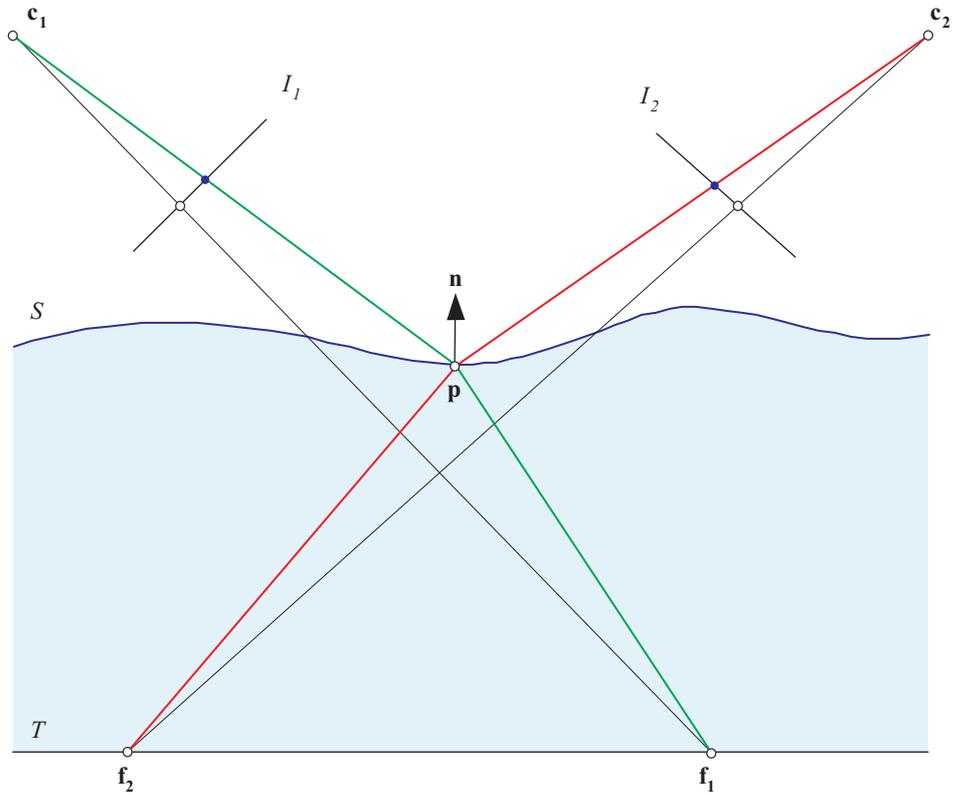


Figure 3.4: Points f_1 and f_2 on the plane T are both refracted at point p and imaged in camera c_1 and c_2 respectively. Since both rays $c_1 p f_1$ and $c_2 p f_2$ intersect the water surface at p , they share the common surface normal n . So, when two points are imaged through a common surface point, they also share a common surface normal. This gives us our stereo normal constraint for determining true surface points.

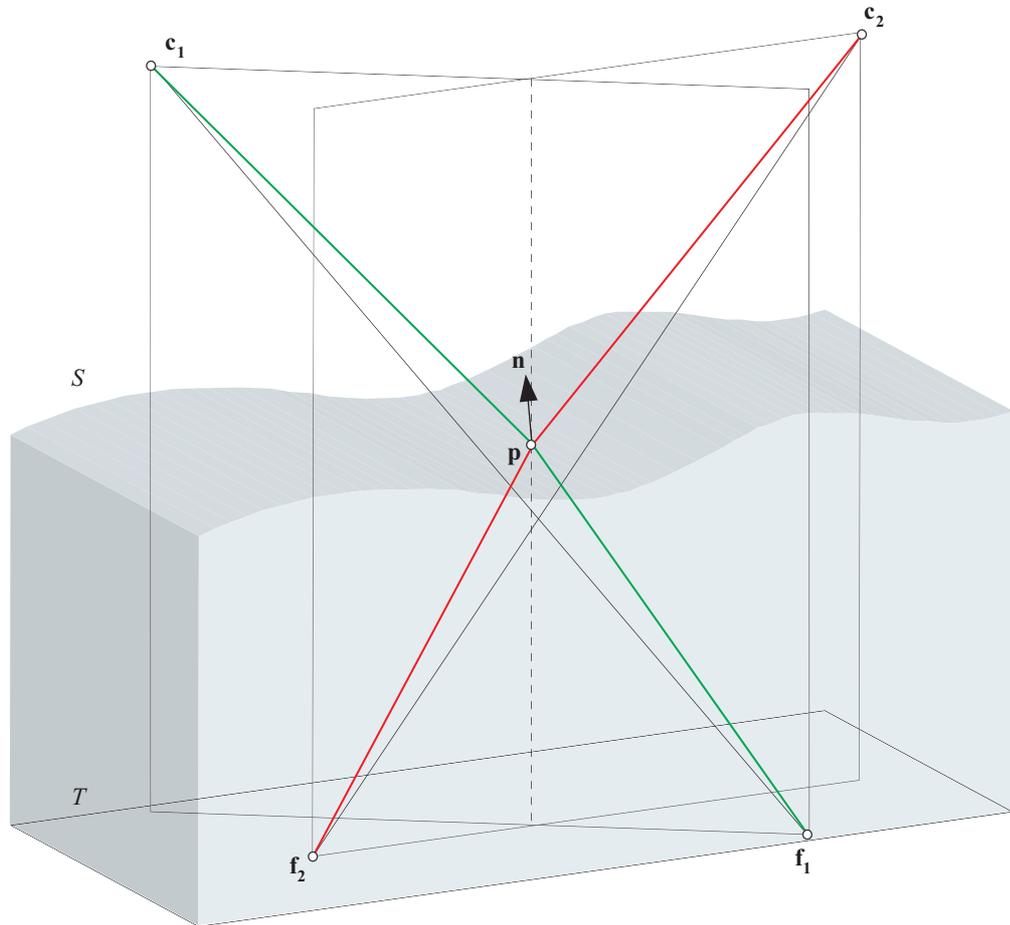


Figure 3.5: This figure shows the imaging system in three dimensions. Point \mathbf{f}_1 on T is refracted at \mathbf{p} toward camera \mathbf{c}_1 and point \mathbf{f}_2 is also refracted at \mathbf{p} toward camera \mathbf{c}_2 . These points make up two intersecting planes: points \mathbf{c}_1 , \mathbf{f}_1 , \mathbf{p} lie on one plane and points \mathbf{c}_2 , \mathbf{p} and \mathbf{f}_2 lie on another plane. Notice that \mathbf{p} lies on the intersection of the planes.

3.2 The geometry of stereo water surface reconstruction

In this section we examine the theory involved in determining surface points in our ideal imaging model. First we will look at how the surface normal can be determined given a known surface location. Then we will discuss our indirect stereo triangulation algorithm for determining depth and surface normals given stereo imagery of an arbitrary surface.

3.2.1 Deriving the surface normal from the incident and refracted rays

We can determine the surface normal \mathbf{n} that would cause the refraction of the incident ray if we know the location of the surface point \mathbf{p} . Refer to Figure 3.6 to see the imaging setup. Using Snell's law (Equation (3.2)) and our knowledge of the angle between the incident and refracted rays (θ_δ), we are able to derive a solution to the surface normal. We define θ_δ as

$$\theta_\delta = \theta_i - \theta_r. \quad (3.5)$$

But θ_i and θ_r are both unknown. In contrast, the following points are known: the surface point \mathbf{p} , the image of the feature point \mathbf{q}' and the feature point \mathbf{f} . Thus we can determine the vectors of the incident (\mathbf{u}) and refracted (\mathbf{v}) rays:

$$\mathbf{u} = \mathbf{q}'\mathbf{f}', \quad (3.6)$$

$$\mathbf{v} = \mathbf{p}\mathbf{f}. \quad (3.7)$$

Both \mathbf{u} and \mathbf{v} are normalized to obtain $\hat{\mathbf{u}}$ and $\hat{\mathbf{v}}$:

$$\hat{\mathbf{u}} = \frac{\mathbf{u}}{\|\mathbf{u}\|}, \quad (3.8)$$

$$\hat{\mathbf{v}} = \frac{\mathbf{v}}{\|\mathbf{v}\|}. \quad (3.9)$$

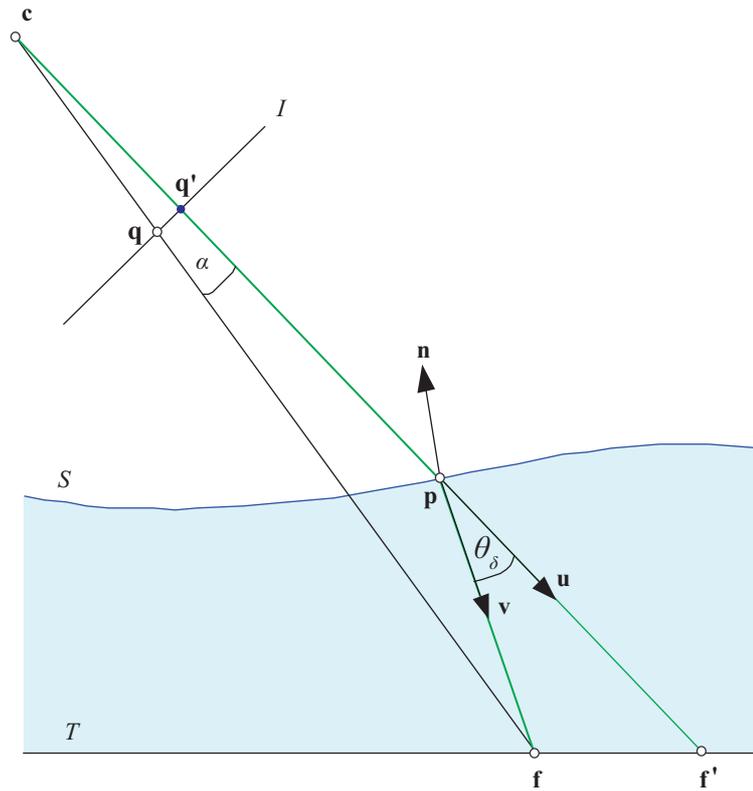


Figure 3.6: The figure shows the imaging of a point \mathbf{f} with water (\mathbf{q}') and without water (\mathbf{q}). If the vectors defined by $\mathbf{q}'\mathbf{f}'$ and $\mathbf{p}\mathbf{f}$ are known, we can determine the ray vectors \mathbf{u} and \mathbf{v} , and hence, the surface normal \mathbf{n} .

The inner product of $\hat{\mathbf{u}}$ and $\hat{\mathbf{v}}$ gives us θ_δ :

$$\theta_\delta = \hat{\mathbf{u}} \cdot \hat{\mathbf{v}}. \quad (3.10)$$

In order to find the surface normal, we need the incident angle θ_i (the angle between the incident ray \mathbf{u} and the normal \mathbf{n}). We substitute (3.5) into Snell's law (3.2) and apply trigonometric identities to find an equation for the incident angle θ_i :

$$\begin{aligned} \sin \theta_i &= r_w \sin(\theta_i - \theta_\delta), \\ \sin \theta_i &= r_w (\sin \theta_i \cos \theta_\delta - \cos \theta_i \sin \theta_\delta), \\ \tan \theta_i &= \frac{r_w \sin \theta_\delta}{r_w \cos \theta_\delta - 1}, \\ \theta_i &= \tan^{-1} \left(\frac{r_w \sin \theta_\delta}{r_w \cos \theta_\delta - 1} \right). \end{aligned} \quad (3.11)$$

So, given θ_δ and the refractive index for water (r_w), we can determine θ_i . The surface normal \mathbf{n} is then determined by rotating $\hat{\mathbf{u}}$ by θ_i about the axis defined by $\hat{\mathbf{u}} \times \hat{\mathbf{v}}$:

$$\mathbf{n} = \mathbf{R}(\theta_i, \hat{\mathbf{u}} \times \hat{\mathbf{v}}) \hat{\mathbf{u}}, \quad (3.12)$$

where $\mathbf{R}(\beta, \hat{\mathbf{x}})$ is the rotation matrix of an angle β about an axis $\hat{\mathbf{x}}$.

The size of the incident angle (θ_i) is strictly increasing as θ_δ is increased (within the physical constraints), so there cannot be multiple values of θ_i for a particular θ_δ . This supports our proposition that there is a unique normal for every depth.

Theorem 3.2.1 (Unique normal) *For every refractive disparity of a point \mathbf{f} imaged in a camera \mathbf{c}_1 and hypothesized surface point \mathbf{p} there is at most one normal \mathbf{n} such that the ray from the \mathbf{c}_1 to \mathbf{p} is refracted to \mathbf{f} .*

Proof Without loss of generality, we will show that there can be at most one incident angle which implies one surface normal.

The physics of refraction constrain the range of the incident angle, such that

$$0 \leq \theta_i < \pi/2.$$

By Snell's Law, the refracted angle is also constrained to the following range,

$$0 \leq \theta_r < \sin^{-1}(1/r_w).$$

Thus the difference between these angles, θ_δ , is physically constrained such that

$$0 \leq \theta_\delta < \pi/2 - \sin^{-1}(1/r_w).$$

The incident angle is computed in Equation 3.11. If we can show that this function is monotonically increasing, then there can be at most one incident angle for any given refraction.

Equation 3.11 can be written as follows:

$$\tan \theta_i = \frac{r_w \sin \theta_\delta}{r_w \cos \theta_\delta - 1}. \quad (3.13)$$

We know that the numerator is monotonically increasing within the specified range for θ_δ . We also know that the denominator is monotonically decreasing and approaches zero when θ_δ approaches $\pi/2 - \sin^{-1}(1/r_w)$.

This means that the right hand side of Equation 3.13 is monotonically increasing. The arctangent of this function is again monotonically increasing. ■

3.2.2 The geometry of indirect stereo triangulation

Figure 3.7 shows the geometric setup for indirect stereo triangulation in an ideal scene. Suppose that two cameras with their centres of projection at \mathbf{c}_1 and \mathbf{c}_2 image a water surface S above a plane T . The image planes of the cameras are denoted as I_1 and I_2 . Moreover, suppose that we take two pairs of images of the plane T , first without water and then with water. From these images reconstruction can proceed.

In order to determine a point on the surface, we use both cameras to triangulate the surface point. We designate one of the two cameras to be the *reference* camera and the other to be the *verification* camera. We present two metrics for measuring the correctness

of a surface point. Both of the metrics require us to determine a surface point from the reference camera and then match the expected surface point against the image data from the verification camera.

The basic reconstruction algorithm first selects a point \mathbf{f}_1 on the plane T . The images of this point are found in the image plane I_1 and are denoted as \mathbf{q}_1 without water and \mathbf{q}'_1 with water. We know from Section §3.1.2 that the water surface intersection point must lie along the ray traced through \mathbf{c}_1 and \mathbf{q}'_1 (\mathbf{u}). The next step of the algorithm is to traverse this ray, looking for the solution to \mathbf{p} that best fits the image data.

We begin this search by hypothesizing a depth from \mathbf{c}_1 that gives us some surface point (\mathbf{p}'). Given this surface point and the location of the imaged feature point (\mathbf{f}_1), we can determine the incident (\mathbf{u}) and refracted rays (\mathbf{v}) from Equations (3.6) and (3.7).

This allows us to compute θ_δ as in Equation (3.10). Next we substitute (3.10) into (3.11) to get θ_i and then compute the normal \mathbf{n}_1 , that would refract \mathbf{u} to \mathbf{f}_1 , from Equation (3.12).

Since we hypothesized \mathbf{p}' , we need some way to verify whether \mathbf{p}' is close to the actual surface location \mathbf{p} . This is where we utilize our second camera. We trace a ray from \mathbf{p}' back to \mathbf{c}_2 , finding the image of a feature (\mathbf{f}_3) at \mathbf{q}_3 . This gives us a new set of incident (\mathbf{u}_v) and refracted (\mathbf{v}_v) rays and difference angle ($\theta_{v\delta}$):

$$\mathbf{u}_v = \mathbf{c}_2\mathbf{q}'_3 \quad (3.14)$$

$$\mathbf{v}_v = \mathbf{p}'\mathbf{f}_3 \quad (3.15)$$

$$\theta_{v\delta} = \hat{\mathbf{u}}_v \cdot \hat{\mathbf{v}}_v \quad (3.16)$$

We then use Equations (3.10), (3.11) and (3.12) to compute a second normal \mathbf{n}_2 for \mathbf{p}' .

At this point we apply our error metrics to determine the validity of the hypothesized point \mathbf{p}' .

The algorithms for computing the metrics begin in the same way. They take a given surface depth (z) as input for a particular feature. Figure 3.2 shows z as the distance between the camera \mathbf{c} and the surface point \mathbf{p} . The feature imaged with and without water determines a solution set of depths with corresponding normals. Since a depth is given as input, the corresponding normal \mathbf{n}_1 is also determined (Figure 3.7). The surface point associated with this depth is viewed from the verification camera and has an associated refractive displacement. This refractive displacement also has a solution set of depths and normals. Since the depth is already constrained by the specification of the surface point \mathbf{p}' , we can compute a second normal \mathbf{n}_2 .

The first metric, which we call the *normal collinearity* metric, matches the normals computed by the reference and verification cameras. The value (E_{normal}) of the metric is determined as follows:

$$E_{normal} = \cos^{-1}(\mathbf{n}_1 \cdot \mathbf{n}_2) \quad (3.17)$$

The intuition for this matching differs from the classical stereo problem where points are matched and the stereo disparity corresponds directly to the depth of the surface point. In this case, we cannot directly image the surface point due to the refraction. Instead we must use the view dependent refracted images to find the position. The refraction is dependent on the orientation and depth of the water surface point and since we hypothesize a depth we must try to account for the refraction with the surface normal. Recall that surface normals translate to a unique water depth, so if the surface normals that explain the refraction from both views are collinear, then this is a strong indication that we have the true water depth. If the normals are not collinear, then angle between normals should give a smooth estimate of the depth error.

We call the second metric the *disparity difference* metric. This metric measures the difference in disparity that occurs when \mathbf{n}_1 is swapped for \mathbf{n}_2 and the incident rays from the respective cameras are refracted. Figure 3.8 shows the disparities between \mathbf{f}_1 and \mathbf{f}'_1

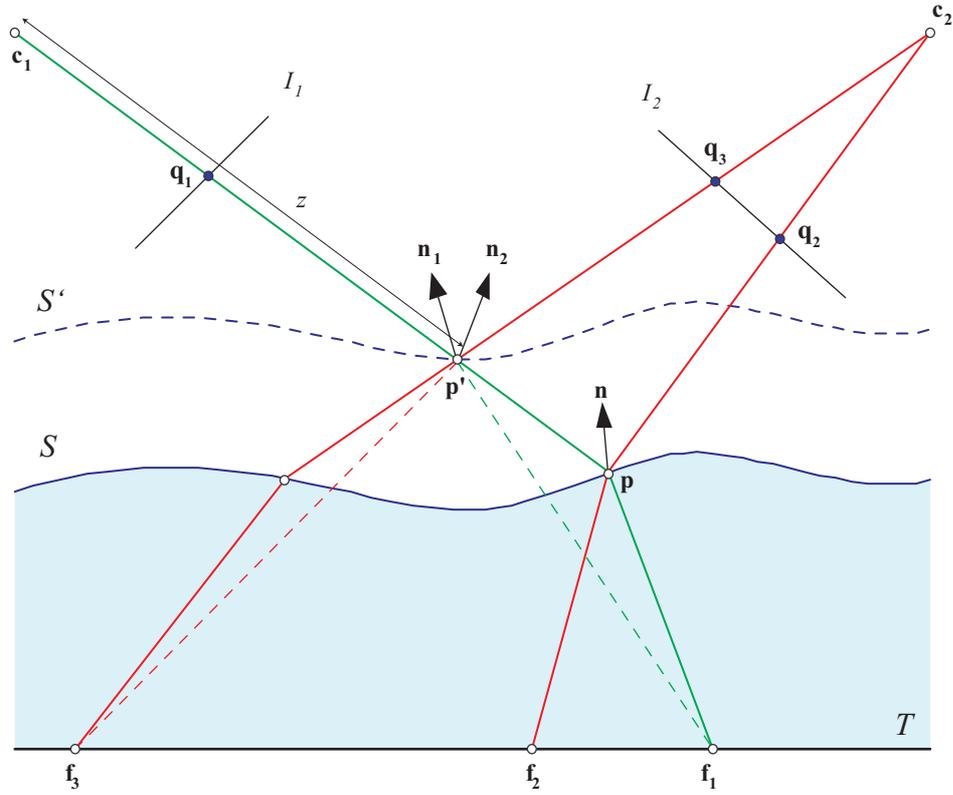


Figure 3.7: The Figure shows the reference camera (\mathbf{c}_1) viewing a point \mathbf{f}_1 on the plane T . Then a depth z is hypothesized, giving a surface point \mathbf{p}' and a normal \mathbf{n}_1 . The verification camera (\mathbf{c}_2) is used to verify the hypothesized surface point \mathbf{p}' , generating a second normal \mathbf{n}_2 . Point \mathbf{p}' coincides with the actual surface point \mathbf{p} if and only if the normals computed from both cameras are identical. When \mathbf{p}' is not equal to surface point \mathbf{p} , we therefore obtain two normals, \mathbf{n}_1 and \mathbf{n}_2 .

and between \mathbf{f}_2 and \mathbf{f}'_2 . We define the distances between these paired points as e_1 and e_2 respectively:

$$e_1 = \|\mathbf{f}_1 - \mathbf{f}'_1\| \quad (3.18)$$

$$e_2 = \|\mathbf{f}_2 - \mathbf{f}'_2\| \quad (3.19)$$

Then we define the error metric to be the sum of these distances:

$$E_{disp} = e_1 + e_2 \quad (3.20)$$

The disparity difference metric merges indirect stereo refraction with conventional stereo. We expect the disparity difference to provide a deeper error surface in shallow water where the surface normal has less bearing on the displacement. This metric builds on the same intuition as the first metric, since it also penalizes mismatched normals. When the water depth is shallow and feature localization errors become comparable to the water depth, the effect of the normal on refraction becomes insignificant. The metric models this by relating the error to the depth, so large normal differences at low depths aren't given as high an error as the same normal difference at a higher depth.

The theoretic process for verifying a hypothesized depth is shown in Algorithm 1.

Once the problem is broken down like this, we can perform a simple error minimization routine to discover the actual depth of the water and the surface point \mathbf{p} .

3.3 Practical water surface reconstruction

In the previous section we presented a method for determining points on the water surface given binocular stereo views of the water. The process relies upon pairs of images of points on the plane T with and without water. In this section we present our method for localizing points and determining the correspondence between the points imaged with and without water (*refractive correspondence*).

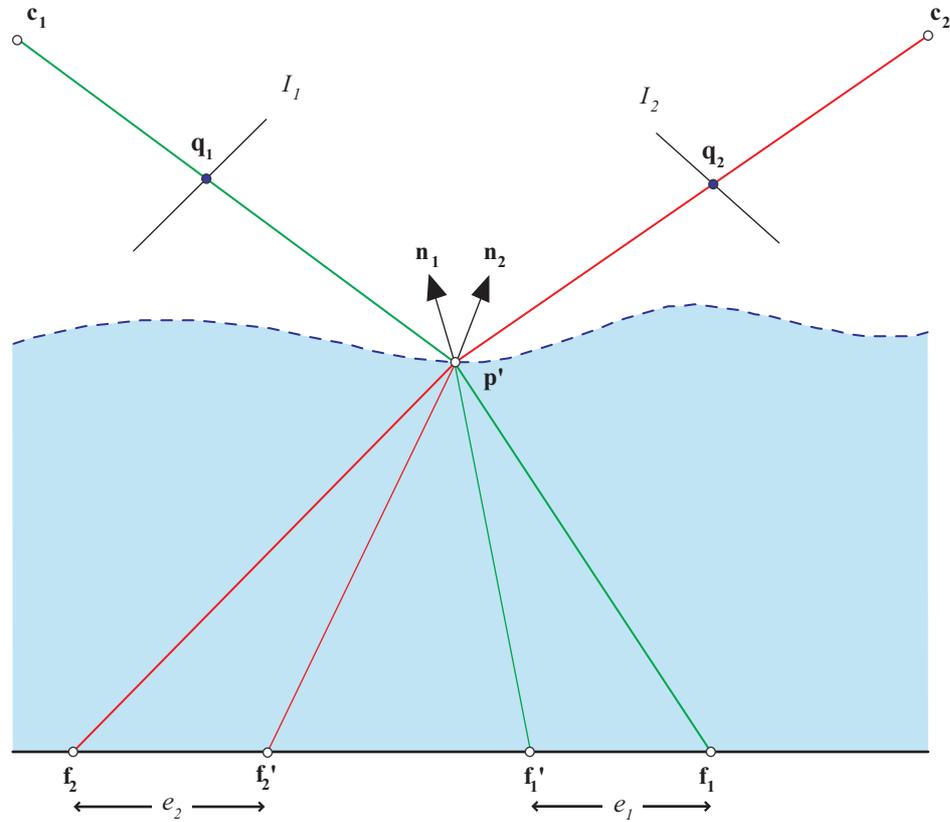


Figure 3.8: The Figure shows a point \mathbf{f}_1 image by the reference camera (\mathbf{c}_1) and a point \mathbf{f}_2 imaged by the verification camera (\mathbf{c}_2) generating normals \mathbf{n}_1 and \mathbf{n}_2 respectively. The normal collinearity metric measures the angle between \mathbf{n}_1 and \mathbf{n}_2 . In contrast, the disparity difference error metric is then determined by swapping normal \mathbf{n}_1 for \mathbf{n}_2 and tracing rays from each camera and refracting them by the swapped normals to get \mathbf{f}_2' and \mathbf{f}_1' . As in Equation (3.20), the metric is the sum of the distances e_1 and e_2 .

Algorithm 1: TheoreticDepthVerification

Input: Hypothesized depth z' , point on T : \mathbf{f}_1 , image of the point: \mathbf{f}'_1 , camera

centres of projection \mathbf{c}_1 and \mathbf{c}_2 , the refractive index of water r_w

Output: Error E

1. Compute \mathbf{p}' from hypothesized depth z' along ray $\mathbf{c}_1\mathbf{f}'$;
 2. Compute \mathbf{u} and \mathbf{v} using Equations (3.6) and (3.7);
 3. Find θ_δ from \mathbf{u} and \mathbf{v} using Equation (3.10);
 4. Given θ_δ and r_w , compute θ_i from Equation (3.11);
 5. Given θ_i , \mathbf{u} and \mathbf{v} , compute \mathbf{n}_1 as from Equation (3.12);
 6. Intersect $\mathbf{p}'\mathbf{c}_2$ with I_2 to get the image of a point \mathbf{q}'_3 . This image point corresponds to a point \mathbf{f}_3 on T ;
 7. Compute \mathbf{u}_v and \mathbf{v}_v using Equations (3.14) and (3.15);
 8. In the same manner as before, compute $\theta_{v\delta}$ and \mathbf{n}_2 using \mathbf{u}_v and \mathbf{v}_v ;
 9. Compute the error E from the disparity difference metric or normal collinearity metric using Equations (3.20) or (3.17) respectively;
 10. Return E ;
-

So far we have described a system for determining single surface points at a particular instant in time. Since we would like to be able to capture sequences of dynamic water surfaces, we also require our system to track the points on T between frames.

Finally, we present the implementation of our water surface reconstruction algorithm that uses a finite set of feature points on T . We also present our algorithm for reconstructing captured sequences.

3.3.1 Pattern specification for feature localization and correspondence

In order to locate points on T we require feature points that can be reliably localized in images. In our system we place a pattern with sharp features onto T in full view of both cameras. For reconstruction, the pattern must be fully visible, especially when covered by water.

There are several challenges to localizing the features on the pattern. We require both feature localization at particular frames and feature tracking of the apparent movement of the features over time. Note that it is not the features that move, but their refracted images that shift due to changes in the water surface between frames. We also need to compute two correspondences. First, we must match features between our binocular views of the pattern in order to determine refractive stereo disparity. Secondly, we need to be able to find correspondences between the images of the pattern and images of pattern through water.

The choice of pattern is crucial for our reconstruction algorithm and its accuracy. Our system is implemented to use a monotone chequered pattern that provides hard edges and distinct corners. The density of the pattern also affects reconstruction. If the pattern is too dense, localization may suffer since the support region for the corners is smaller. Also, a dense pattern is subject to a greater degree of feature elimination and separation due to refraction of opposing normals. Elimination occurs when a feature point becomes

invisible to the camera due to refraction limits and separation occurs when two adjacent features appear separated after refraction. These effects are also more pronounced in deeper water.

In order to determine both the frame to frame correspondence and the corner localization, our system utilizes a Lucas-Kanade type template matching technique [5]. Template images are generated around the checker corners from an image of the pattern without water. We then match these templates against the corners in subsequent frames. The support region around the corners allows for high localization precision. The templates are locally specific and will not match against any of the four nearest corners since those corners have reciprocated black and white checkers. This makes the algorithm more robust to some elimination.

Finally our system is designed to handle two cases of refractive correspondence. For reconstruction sequences that begin with no water, corner localization at the start of the sequence is used to locate the feature positions on T and subsequent images are used for reconstruction. The other case we handle is for sequences beginning from calm water. In this case we require an image of the T to locate the features without water. We then detect the features from the calm water images at the start of the sequence. Since the water is calm, we assume there is no elimination or separation and the only distortion is the monotonic refractive distortion. We are then able to locate the corners accurately by stretching the grid to match at the boundaries and relocalizing each corner, giving a correspondence for the refractive disparity.

3.3.2 Implementation of indirect stereo triangulation

Here we will outline the process for computing point locations and normals on the water surface given a finite set of feature points on T .

We implemented the surface point triangulation algorithm as a one dimensional minimization problem. The cost function (C) takes in a hypothesized depth z' and returns

an error associated with that depth:

$$E_{depth} = C(z') \quad (3.21)$$

Recall that in our error metrics, we utilize the second camera to verify the hypothesis. Section §3.2.2 describes how verification works in theory, where a feature point exists at the end of the verification ray, allowing direct verification of \mathbf{p} . In practice, we only have a finite number of spaced out features on T . We must therefore interpolate between the nearest features in order to perform the verification. Figure 3.9 shows the scenario in two dimensions with linear interpolation.

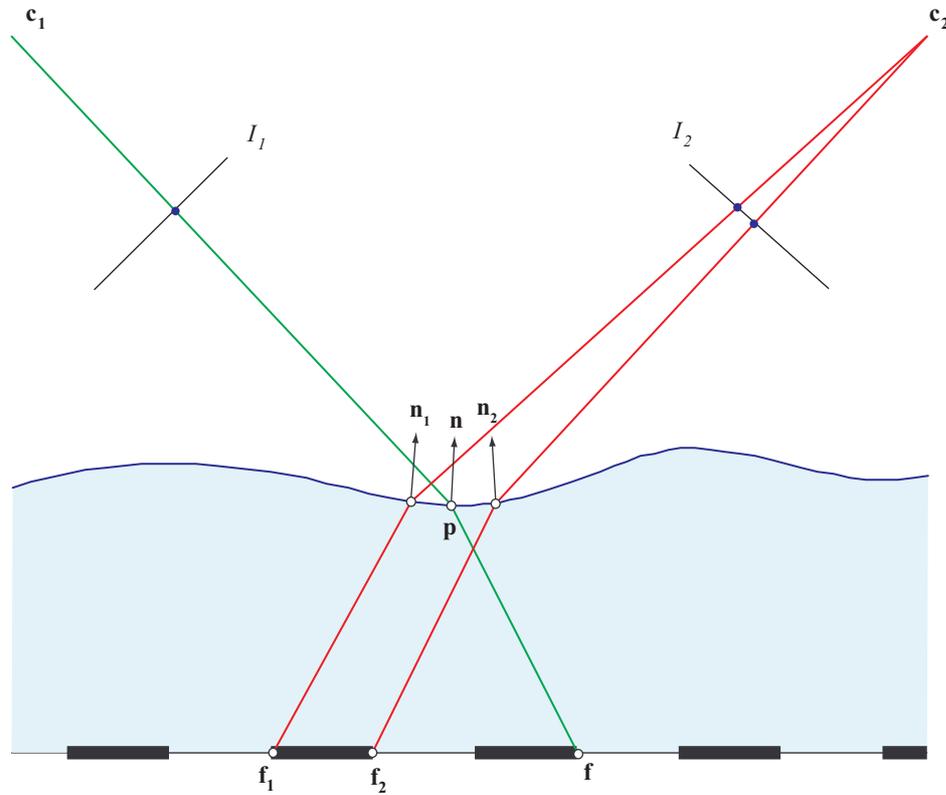


Figure 3.9: When verifying a surface point \mathbf{p} , due to the discrete placement of features, we cannot assume that there will be a feature projecting through \mathbf{p} to \mathbf{c}_2 . So we find the refractive disparity of the nearest features and interpolate to get an approximate disparity that we use to find the verification normal at \mathbf{p} .

Verification is computed through the determination of the surface normal along the verification ray. As can be seen in Figure 3.9, the normal of the verification ray itself is not known. Instead we must compute the refractive disparities of the features that project closest to the desired point. We then perform an interpolation step to find the approximate refractive disparity along the verification ray ($\mathbf{c}_2\mathbf{p}$). This is then used to find the verification normal. Although Figure 3.9 shows the scenario in two dimensions, our implementation had to be three-dimensional. Thus we implemented a bilinear interpolation to approximate the normal, interpolating the normals at the four nearest non-collinear corners. The interpolation is computed as shown in Figure 3.10.

Since the Snell’s law (Equation (3.2)) is non-linear and our surface is not necessarily linear, the bilinear interpolation is not absolutely accurate. Despite this, water’s inherent smoothness and a dense feature set with features located every few pixels means that we can reasonably approximate the verification surface normal.

3.3.3 The algorithm

We utilized some of the physical constraints of the system in our depth estimation routine. We assumed spatial smoothness of the water by limiting our depth search to values close to the depth of neighbouring points.

We can put together all the pieces described previously to form an algorithm for determining the error for a particular hypothesized water depth shown below:

Our global algorithm processes frame sequences and uses the DepthVerification algorithm to determine the water surfaces. The process cycles through each frame, tracking the feature points as they are distorted by the water. It passes the tracked features and a hypothesized depth into the DepthVerification algorithm which returns the error associated with the hypothesis. This process is repeated and the error is minimized in order to determine the best depth estimate and thus the location of the surface point. A surface mesh is then generated from all the surface points in each frame.

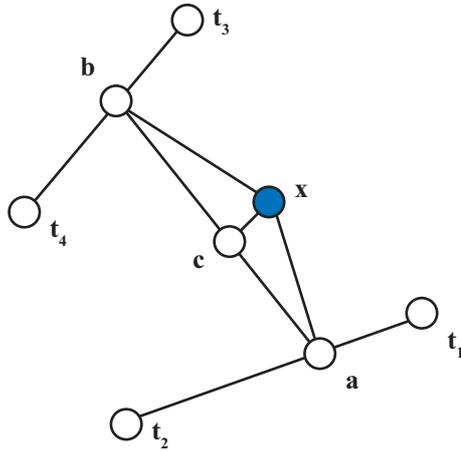


Figure 3.10: Bilinear interpolation on imaged feature points in the reference camera. Since our features are not dense and the refractive disparity is only known at these features, we must interpolate to get disparity values for points lying in between the feature points. The refractive disparity of a point \mathbf{x} is approximated from the known disparities of four localized feature points \mathbf{t}_1 , \mathbf{t}_2 , \mathbf{t}_3 and \mathbf{t}_4 . \mathbf{x} is projected onto $\mathbf{t}_1\mathbf{t}_2$ to get \mathbf{a} and onto $\mathbf{t}_3\mathbf{t}_4$ to get \mathbf{b} . Then \mathbf{x} is projected onto \mathbf{ab} to get \mathbf{c} and the disparities at the end points of $\mathbf{t}_1\mathbf{t}_2$ and $\mathbf{t}_3\mathbf{t}_4$ are interpolated to get disparities for \mathbf{a} and \mathbf{b} . Then the disparities of \mathbf{a} and \mathbf{b} are interpolated to get a final disparity for \mathbf{c} , which is the bilinear approximation of \mathbf{x} .

Algorithm 2: DepthVerification

Input: Hypothesized depth z' , feature \mathbf{f}_1 , feature image \mathbf{f}'_1 , set of all pattern features \mathbf{F} , set of features \mathbf{F}_2 imaged from the reference camera \mathbf{c}_2 , camera centres of projection \mathbf{c}_1 and \mathbf{c}_2

Output: Error E

1. Compute \mathbf{p}' from hypothesized depth z' ;
 2. Compute \mathbf{u} and \mathbf{v} using Equations (3.6) and (3.7);
 3. Given \mathbf{p}' , \mathbf{u} and \mathbf{v} compute \mathbf{n}_1 from Equations (3.10), (3.11) and (3.12);
 4. Find the four non-collinear features in \mathbf{F}_2 that project closest to the hypothesized surface point from the view of the verification camera;
 5. Bilinearly interpolate the refractive disparity of four features to get the approximate refractive disparity of the verification ray. Then compute the verification normal \mathbf{n}_2 ;
 6. Swap \mathbf{n}_1 and \mathbf{n}_2 to compute the error distances e_1 and e_2 ;
 7. Return $E = e_1 + e_2$;
-

Algorithm 3: SequenceReconstruction

Data : Binocular frame sequence of pattern through water. Calibrated camera

system. Initial feature locations. Start and end frames.

Result : Water mesh sequence

$i \leftarrow startFrame$;

while $i < endFrame$ **do**

foreach *feature point* **f** **do**

 Minimize DepthVerification using Golden section algorithm to give best-
 Depth;

 Determine surface point from bestDepth;

 Generate mesh from set of surface points;

 Perform Lucas-Kanade localization on each feature in the next image $i + 1$

 using the previous feature location as a seed point;

Chapter 4

Results

“If you wish to drown, do not torture yourself with shallow water.”

-Bulgarian Proverb

In this chapter we describe the apparatus and physical setup for our system. We then analyze the performance of our reconstruction system. We begin by selecting several parameters that govern the error in our reconstruction. In order to measure this error we present a set of metrics that allow us to examine the effect of our parameters. We then explain how we designed a simulation of our algorithm to test the error parameters. Subsequently, we present results from the simulation and compare them to results from real world data. Finally we present results of reconstructed water sequences.

4.1 Apparatus and Physical Setup

We are also forced to constrain our system due to physical limitations of our apparatus. Since our imaging system is not a perfect pinhole camera and nor does it produce an orthographic projection we were careful to calibrate our system to take into account a reasonable approximation of these imperfections. In this section we describe our physical apparatus and setup. We describe the assumptions we make and the constraints we

employ.

4.1.1 Apparatus and imaging system

We decided to use a glass tank to constrain the water we were reconstructing. The tank was raised on a frame (Figure 4.1) to allow an image to be projected onto the tank bottom. We placed a back-lit chequered screen on the tank bottom to allow the image to be viewed from above. The screen was in direct contact with the water to avoid any other refraction. During our experiments, the only lighting of the scene came from the lighting below the surface of the water.

We viewed the water from above with two cameras aiming from opposite ends of the tank. A trade off exists between baseline length and the size of the reconstructable area. A longer baseline produces greater disparity between refracted features, but the result is a smaller overlap between the refracted images and thus the reconstructable area, as can be seen in Figure 4.2. The overlap is necessary for our stereo triangulation as described in Section §3.2.2.

We used twin Sony DXC-9000 3CCD cameras in progressive scan mode to feed synchronized image data into two Matrox Meteor II video capture boards. The images were captured with a resolution of 640x480 pixels at 60 frames per second.

4.1.2 Camera calibration

In order to enhance the accuracy of our technique we wanted an accurate model for our cameras and physical setup. To this end we performed intrinsic and extrinsic calibration.

We performed intrinsic camera calibration according to the technique described in [39]. This allowed us to estimate the focal length, centre of projection and lens distortion.

We then extrinsically calibrated our stereo camera pair by imaging a common pattern on the bottom of the tank. This gave us the transformation for both cameras to a new coordinate system originating at the calibration pattern on the tank bottom. We per-

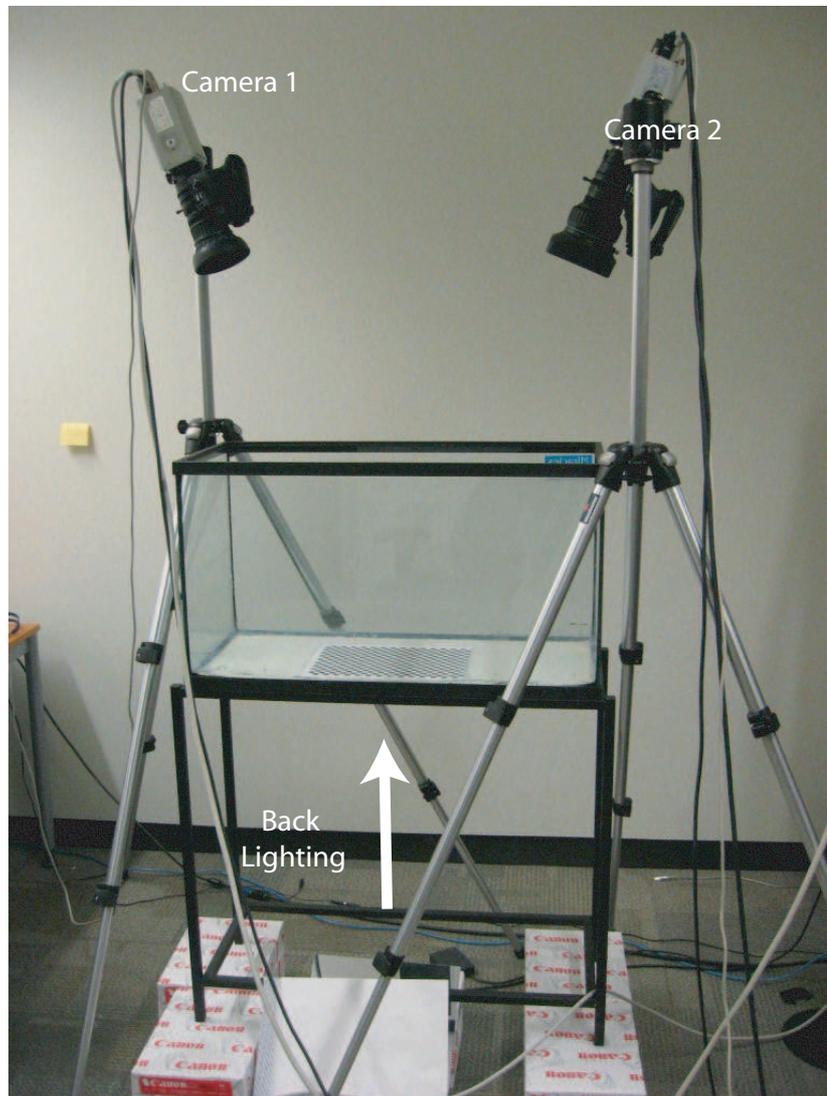


Figure 4.1: Physical setup and apparatus

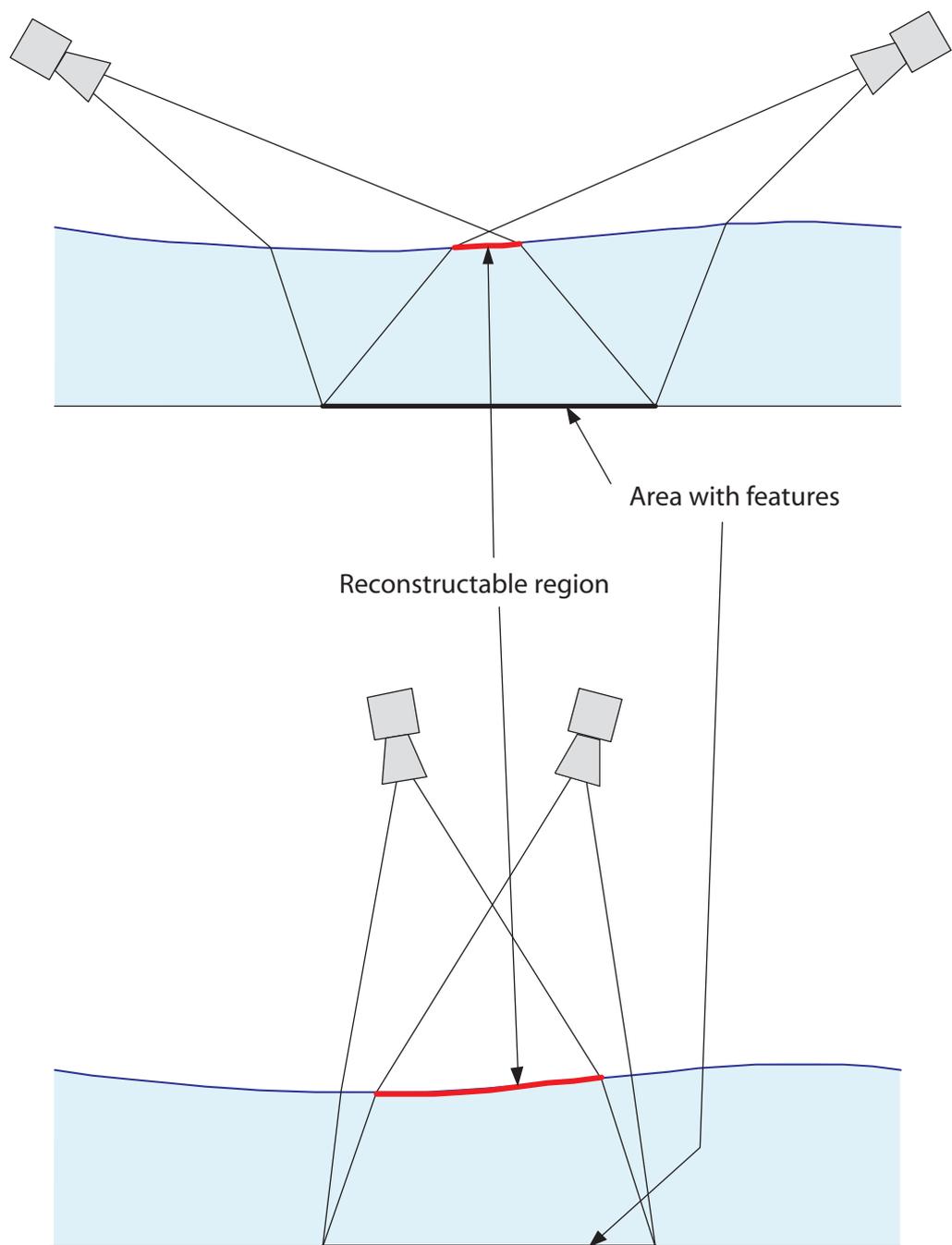


Figure 4.2: Trade-off between baseline length and reconstructable region size

formed the rest of the implementation using this coordinate system. Figure 4.1 shows the calibration pattern on the tank bottom, ready for extrinsic calibration. The calibration pattern was subsequently used in the reconstruction phase.

We calibrated the cameras using a short exposure time ($1/500$ s) so that motion blur would not affect the reconstruction process. Another important step was to make sure the camera was focused precisely on the pattern and calibrated well around that depth range. We required bright lighting to compensate for the quick shutter speed and to allow for as small an aperture size as possible. The small aperture was necessary to reduce depth of field blurring.

4.2 Water surface reconstruction simulation

We performed several experiments using a simulation of our system in order to analyse the expected performance and behaviour of the system on real data. First, we describe how the simulation was created and how it approximates real world results. We then analyze the performance of our two error metrics, selecting the disparity difference metric as more effective. The remainder of our results are all computed using this metric. Then we present and discuss results that compare the main error contributing factors in the system. Finally we compare our simulated results to real world data.

4.2.1 Simulation implementation

Since our system is image-based and all our measurements are computed from the images, reconstruction errors occur from the calibration of the cameras and the ability to localize features within the images. We selected two parameters to quantify the error in the system. These error parameters cover the two primary aspects noted and can readily be estimated in our experiments on real data. We also selected a third parameter that affects the system performance, the height of the water.

The first parameter is the calibration error. This is the error caused by misalignment of the homographies of feature points in the images of both cameras projected onto an extrinsic plane. The calibration error is caused by imperfect intrinsic calibration as well as errors in the calibration of the extrinsic plane for both cameras. Figure 4.3 shows how the calibration error affects the computation of the verification normal \mathbf{n}_2 .

The calibration error parameter is incorporated into our simulation by perturbing the feature homography of the verification camera by some amount ($\Delta \mathbf{f}_i$), normally distributed around a mean. This mean is our input parameter and we label it as the *calibration error* (ρ), measured in millimetres.

Secondly, our system cannot perfectly localize the features in the images due to camera noise and limited resolution. Figure 4.4 shows how the localization error affects point reconstruction. Since the reconstruction relies on the vectors formed from the imaged feature points, error in those points translates in to reconstruction error for the surface point. It is important to note that a drastic error in the localization may results in a physically impossible reconstruction scenario, where the surface normal or depth cannot achieve the refractive displacement. Our system disregards such points.

We incorporated this error into our *localization error* (ψ) parameter. The localization error parameter is the mean of a Gaussian perturbation on the image plane applied to all imaged feature points ($\Delta \mathbf{q}$), measured in pixels.

Our third parameter, the height of the water h , affects reconstruction as it affects the distance of the surface from the cameras, as well as the refractive disparity. In order to simplify the simulation, the interpolation step and error associated with it is ignored.

The simulation works in a similar way to the global sequence reconstruction algorithm as described in Section §3.3.3. Instead of tracking features through a sequence of images, we generate feature points and compute the refractive displacement given the simulation input parameters. The simulation works under the assumption of a flat water surface. The simulation algorithm is designed to compute a set of behaviour and performance

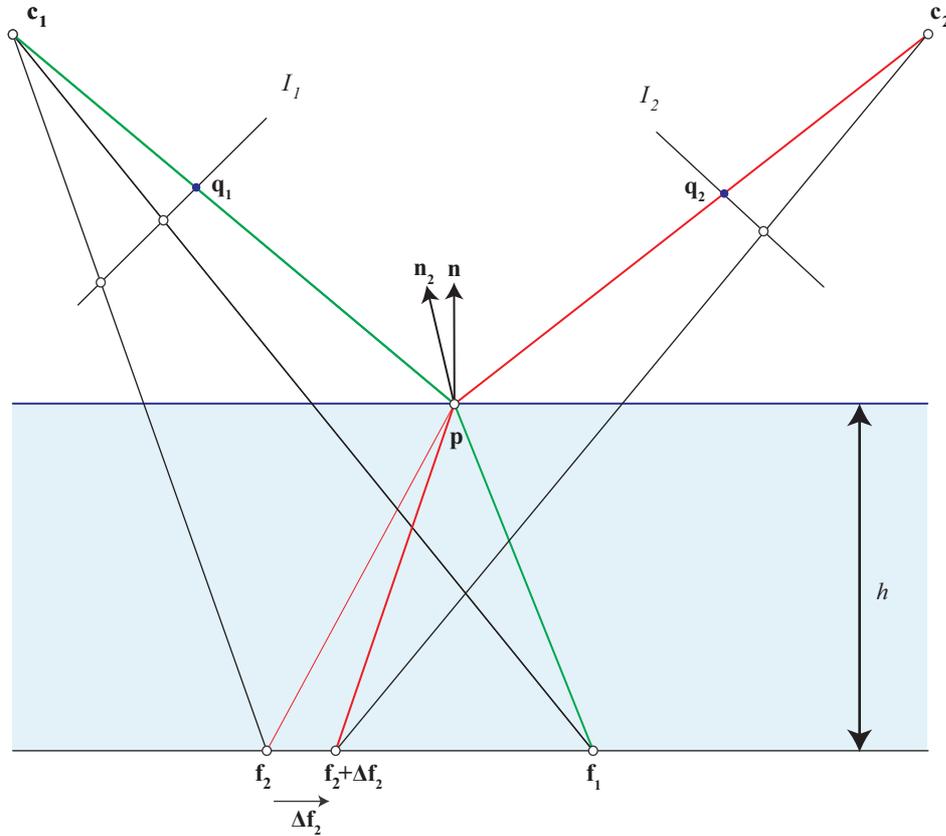


Figure 4.3: Calibration error. When c_2 is used to verify a surface point p , the point is projected into the image plane of c_2 . The feature imaged at the projection point q_2 is used to compute the verification normal as described in §3.2.1. Due to calibration error, the feature imaged at q_2 may in fact be offset from the feature f_2 imaged from the reference camera c_1 by some amount Δf_2 . This causes the verification normal n_2 to become slightly skewed.

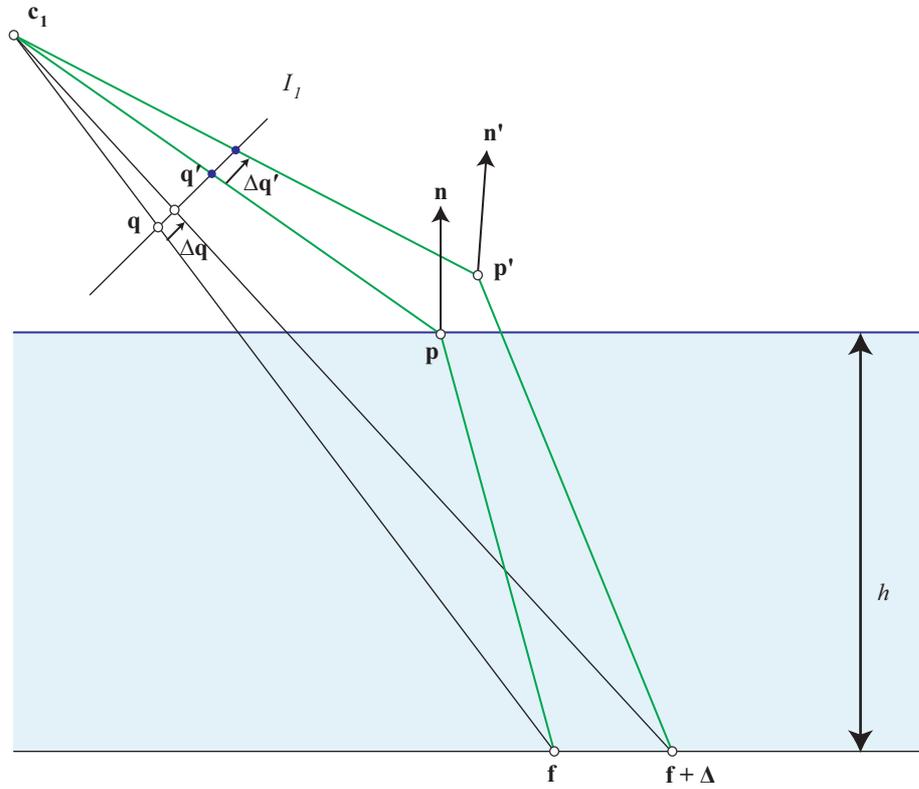


Figure 4.4: Localization error. Feature f is imaged on the image plane I_1 without water at q and with water at q' but due to noise and finite resolution, cannot be precisely localized. Thus there is some perturbation Δq_i in our image point. This perturbation in turn causes a shift in the reconstruction point from p to p' . The surface normal is also affected.

gauges given varying heights and system errors.

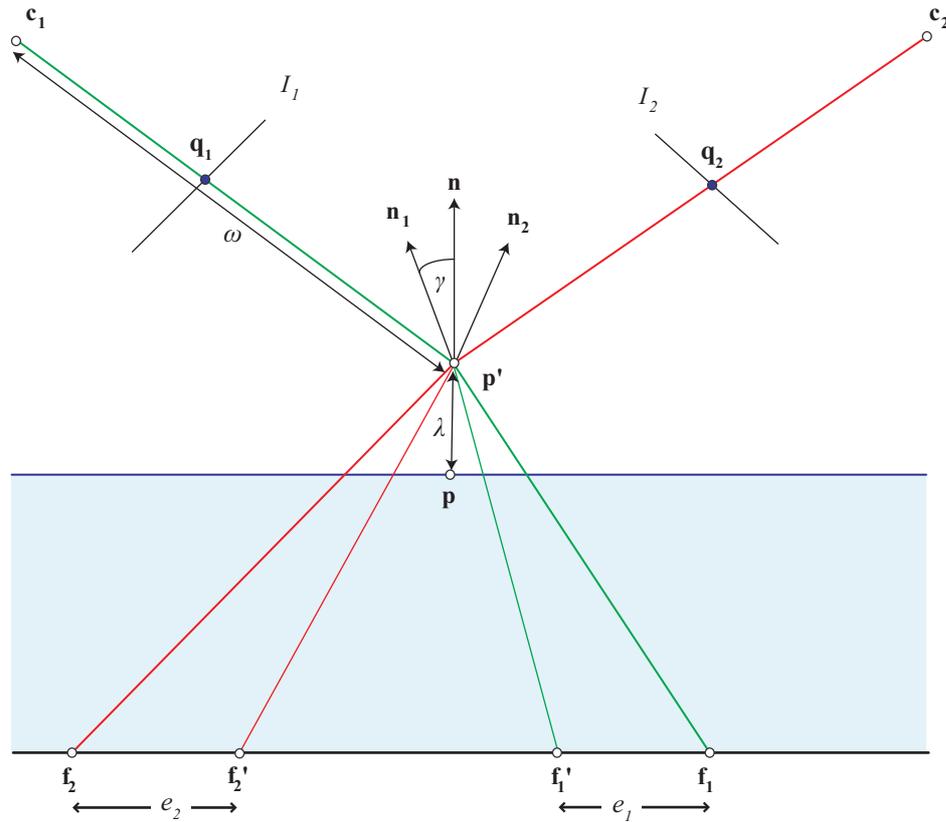


Figure 4.5: Reconstruction gauges. The Figure shows a reconstructed point \mathbf{p}' a distance λ away from the true location \mathbf{p} and a distance ω from \mathbf{c}_1 . The reference camera \mathbf{c}_1 produces a normal \mathbf{n}_1 and the verification camera generates \mathbf{n}_2 . The error metric $E_{disp} = e_1 + e_2$ computed as described in §3.2.2. The angle between the true normal \mathbf{n} and the reconstructed normal \mathbf{n}_1 is γ .

The simulation measures the following behaviour gauges. The measured quantities are displayed in Figure 4.5.

- The average error metric (E) returned by Algorithm `DepthVerificationSimulation` 4.2.1. The error metric is computed as $E = e_1 + e_2$, where e_1 and e_2 are determined as described in §3.2.2.
- The standard deviation and the mean distance (λ) between reconstructed and ac-

tual surface points. The distance is computed as $\lambda = \|\mathbf{p}' - \mathbf{p}\|$.

- The *reconstruction system accuracy*, defined as the average distance between the reconstructed point and the actual surface point divided by the distance to the camera (λ/ω).
- The mean and the standard deviation of the *normal error*, defined as the size of the angle (γ) between the reconstructed normal and the actual normal. It is computed as $\gamma = \cos^{-1}(\mathbf{n}_1 \cdot \mathbf{n})$.

We implemented a slightly simpler version of the DepthVerification algorithm described in Section §3.3.3. This algorithm computes the error associated with a given hypothesized depth but uses the input features, rather than searching for the closest verification features and interpolating.

Algorithm 4: DepthVerificationSimulation

Input: Hypothesized depth z' , shifted feature $\mathbf{f}_1 + \Delta\mathbf{f}_1$, image of shifted feature

\mathbf{q}_1 , shifted feature $\mathbf{f}_2 + \Delta\mathbf{f}_2$, Camera centres of projection \mathbf{c}_1 and \mathbf{c}_2

Output: Error E

1. Compute surface point $\mathbf{p}' = \mathbf{c}_1 + z'\|\mathbf{q}_1 - \mathbf{c}_1\|$;
 2. Compute incident ray $\mathbf{u}_1 = \mathbf{p}' - \mathbf{c}_1$;
 3. Compute refracted ray $\mathbf{v}_1 = \mathbf{f}_1 + \Delta\mathbf{f}_1 - \mathbf{p}'$;
 4. Given \mathbf{p}' , \mathbf{u}_1 and \mathbf{v}_1 compute \mathbf{n}_1 from Equations (3.10), (3.11) and (3.12);
 5. Compute incident ray $\mathbf{u}_2 = \mathbf{p}' - \mathbf{c}_2$;
 6. Compute refracted ray $\mathbf{v}_2 = \mathbf{f}_2 + \Delta\mathbf{f}_2 - \mathbf{p}'$;
 7. Given \mathbf{p}' , \mathbf{u}_2 and \mathbf{v}_2 compute \mathbf{n}_2 from Equations (3.10), (3.11) and (3.12);
 8. Return error metric value E ;
-

Given the DepthVerificationSimulation algorithm, we implemented a simulation algorithm that would take a given height and compute the behaviour gauges outlined above for a range of localization errors and calibration errors. We generated the appropriate refractive distortion given the input height and camera locations. Then we perturbed the features for the localization error and we shifted the feature homographies to approximate the calibration error. The algorithm is outlined in detail in Appendix A.

We implemented a second version of the Simulation algorithm that compared the localization error to varying heights, while maintaining a constant calibration error. The purpose of this was to examine the effect of water height upon the results.

4.2.2 Error metric analysis

In Section §3.2.2 we discussed two methods for matching features between the reference and verification cameras. We presented the normal collinearity error metric which measured the angle between the normals computed by the reference and verification cameras (4.1). The second metric, the disparity difference, measured the difference in disparity between features viewed through the surface point from both cameras and the corresponding projected features computed when the normals are swapped (4.2),

$$E_{normal} = \cos^{-1}(\mathbf{n}_1 \cdot \mathbf{n}_2), \quad (4.1)$$

$$\begin{aligned} E_{disp} &= \|\mathbf{f}_1 - \mathbf{f}'_1\| + \|\mathbf{f}_2 - \mathbf{f}'_2\|, \\ E_{disp} &= e_1 + e_2. \end{aligned} \quad (4.2)$$

We ran a set of simulation experiments using both metrics to determine the behaviour of each as seen in Figure 4.6. Both metrics showed similar behaviour above water heights of 1mm. It is in the relatively shallower water that differences can be seen. The key difference is in the distance error gauge (λ) where the error and error deviation for the normal collinearity rises sharply as the depth drops below 1mm. The disparity difference

in contrast exhibits a relatively slight peak at depths below 0.3mm. Both metrics produce a similar normal error (γ). The remaining experiments all employ the disparity difference error metric.

4.2.3 Feature localization error and calibration error comparison

Our comparison between the localization error and calibration error suggests that localization affects the reconstruction to a much greater degree than the calibration (Figures 4.7, 4.8 and 4.9). Although the calibration and localization errors are measured in different units, in our set up 1 pixel distance projected to approximately 1mm in the tank bottom. The results are all computed for a constant height of 5mm.

The calibration error causes the misalignment of the projected features from the cameras. This means that the verification test does not occur at precisely the correct location. Since we are dealing with flat water, the surface normal is constant over the water and the only difference is the angle of the incident ray. Our cameras are not oblique to the water surface and there is only a small change in the incident angle so only a small change in the refractive displacement occurs. The refractive displacement is what is used to determine the surface normal for verification, explaining why the calibration error has little effect on the reconstruction depth.

We can see that the error metric results closely match the depth error gauges, suggesting that it is a valid error metric.

4.2.4 Analysis of localization error at varying depths

We used the second version of the simulation algorithm to generate graphs comparing the effect of the localization error at varying depths (Figures 4.10, 4.11 and 4.12). We fixed the calibration error to be 0.55mm, comparable to the calibration error determined from

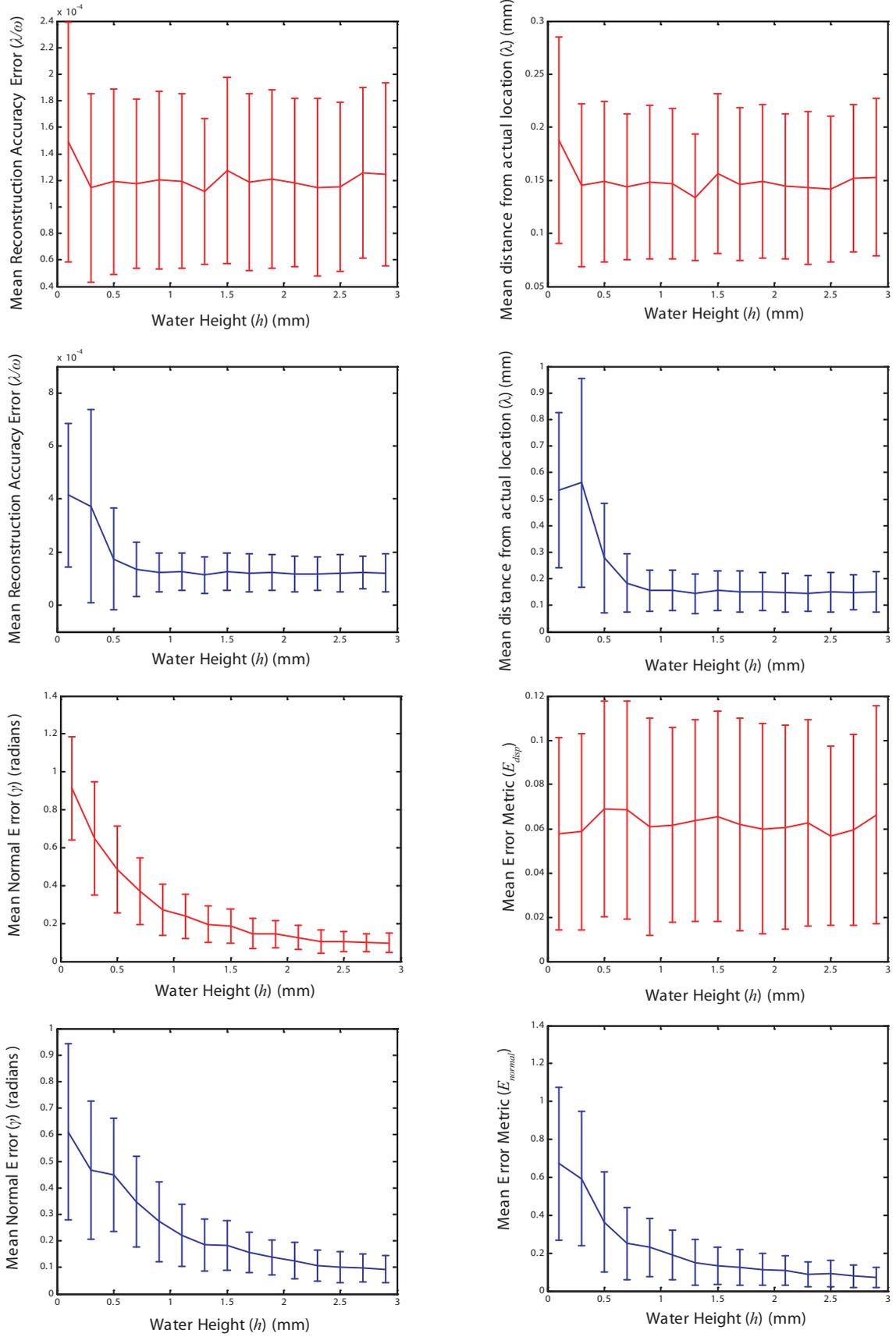


Figure 4.6: Error metric analysis. The normal collinearity metric is shown above the disparity difference metric.

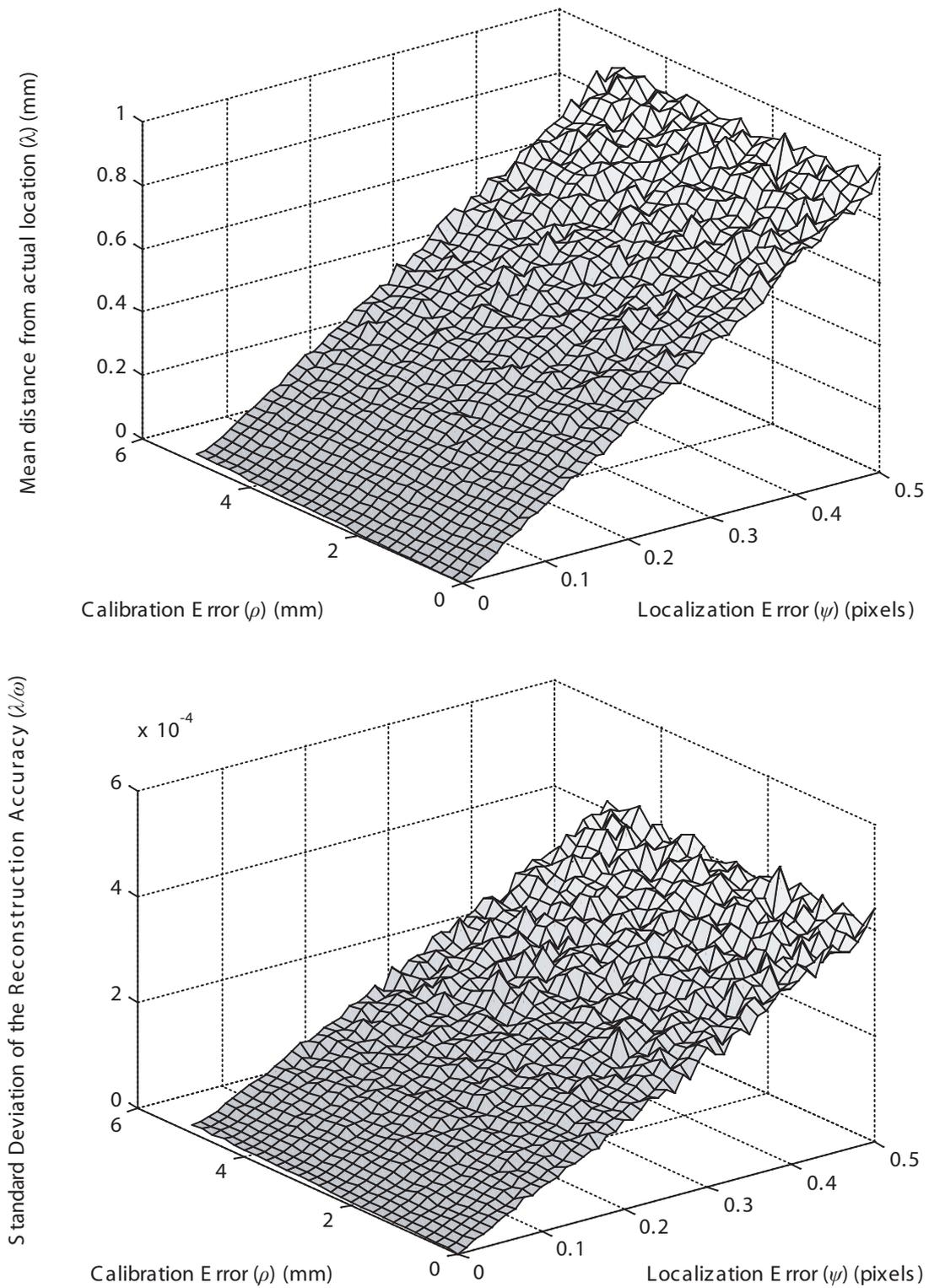


Figure 4.7: Simulation graphs showing the mean distance between reconstructed points and the actual points (top) and the standard deviation the depth reconstruction accuracy (bottom) for varying calibration and localization errors

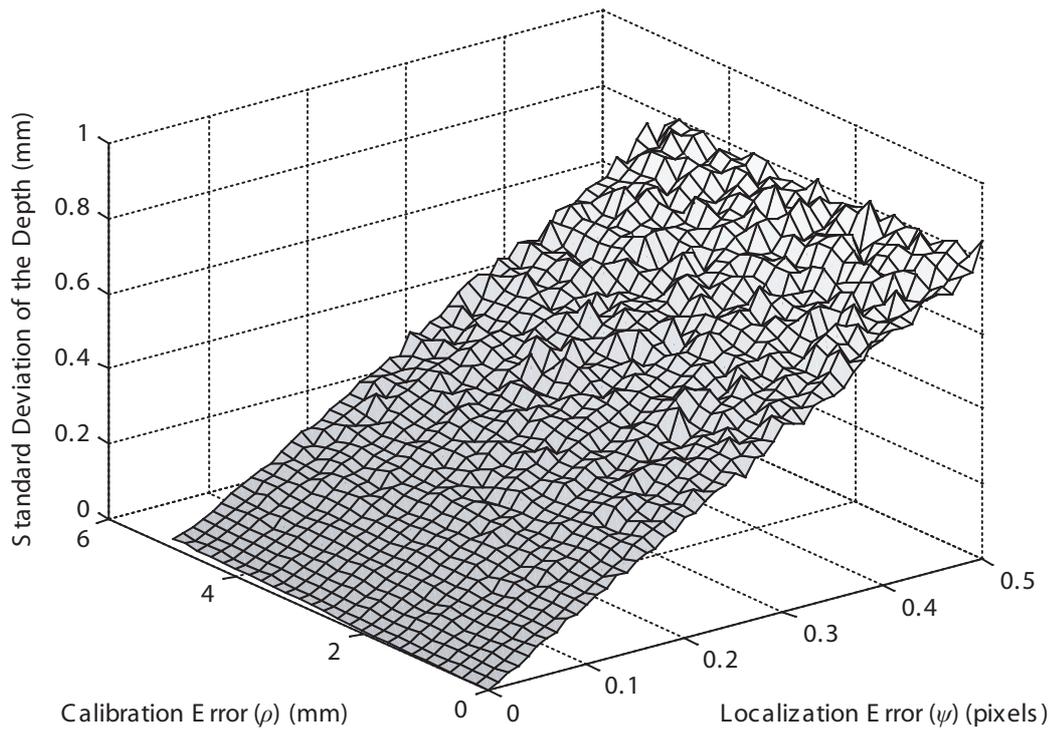
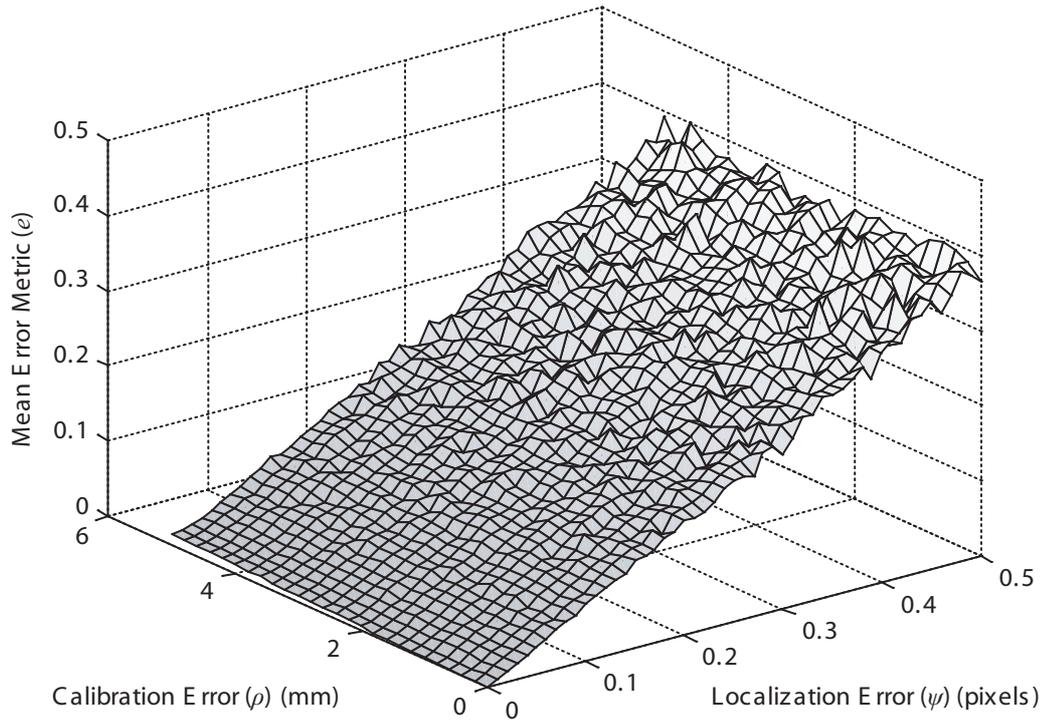


Figure 4.8: Simulation graphs showing the mean error metric (top) and the standard deviation of the reconstructed depths (bottom) for varying calibration and localization errors

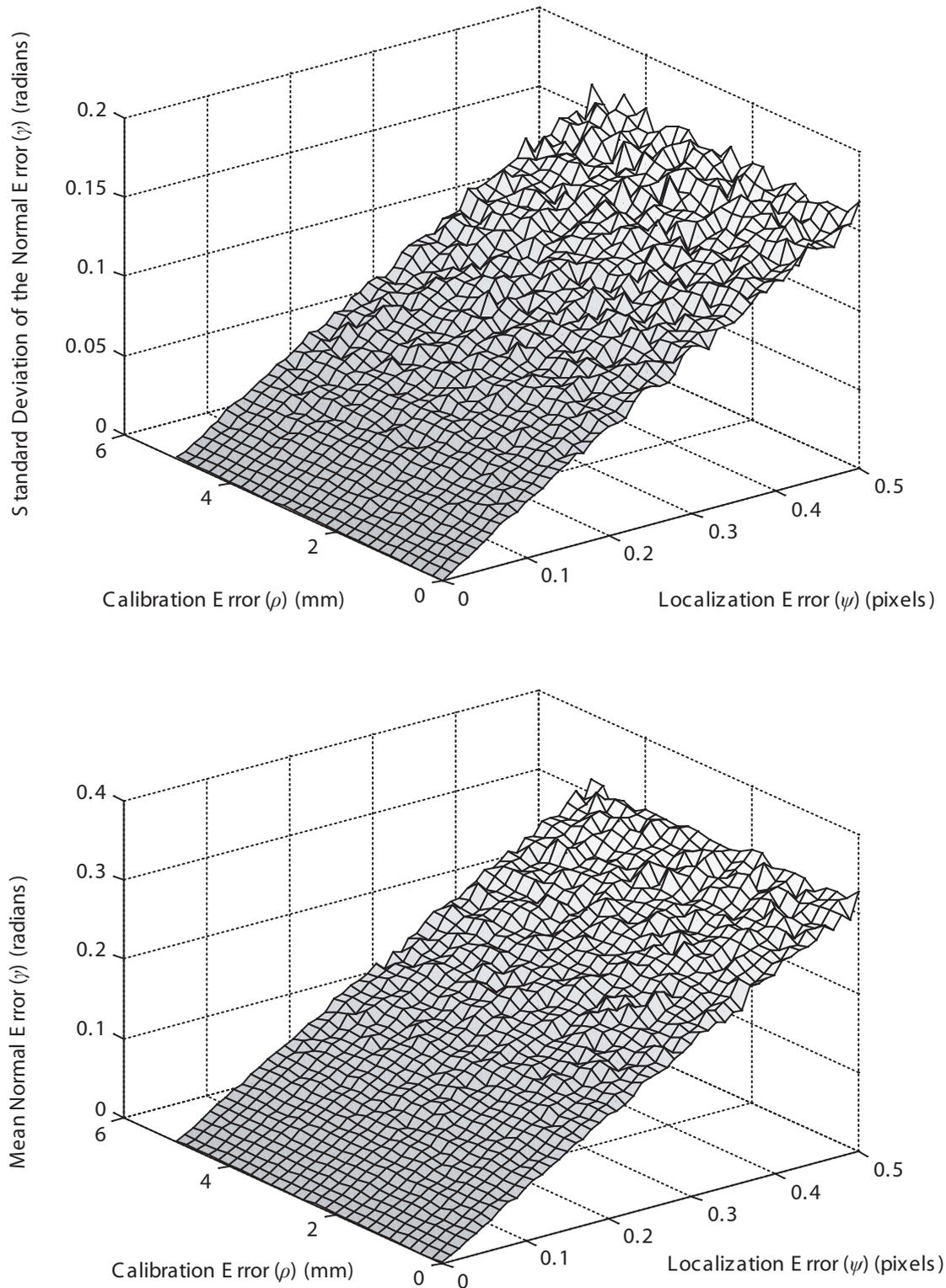


Figure 4.9: Simulation graphs showing the mean and standard deviation of the normal error for varying calibration and localization errors

our real world apparatus. Depth slightly affects the reconstruction error but to a much lesser extent than the localization error (Figures 4.10 and 4.11). These results suggest that our algorithm robustly reconstructs a range of depths. Figure 4.12 demonstrates the degeneration of the surface normal as the water depth approaches zero.

4.2.5 Simulation data compared to real world data

Next, we performed a set of experiments, reconstructing flat water surfaces at varying water heights. We attempted to gauge the results in a similar manner to our simulation gauges. The error metric gauge is directly comparable, but the true location of the surface is unknown so the other gauges must be approximated. Since we were dealing with flat water, we approximated the true surface by a best fit plane through all our data points. This was achieved with Single Value Decomposition on the point set to determine a planar basis.

We then measured the distance of each point from the plane for our distance gauge λ and we compared the point normals to the plane normal for the normal error γ . Plots of the results are shown in Figure 4.13.

In order to compare our empirical results with our simulation results, we needed to determine appropriate values for the calibration and localization simulation parameters.

The calibration error in our empirical system can be estimated by projecting the detected features from both cameras onto the tank bottom plane and measuring the mean correspondence error between the two homographies of feature points. We obtained a mean error of 0.55 mm and used this as our calibration error parameter in our simulation.

The localization error is not as readily available for measurement as the true location of the features cannot be accurately known. We ran tests on our system, where we localized corners for a sequence of twenty frames of our pattern without disturbance. We determined an average position for the corner from these samples and then found the mean perturbation of the samples around the average position. This test gauges the

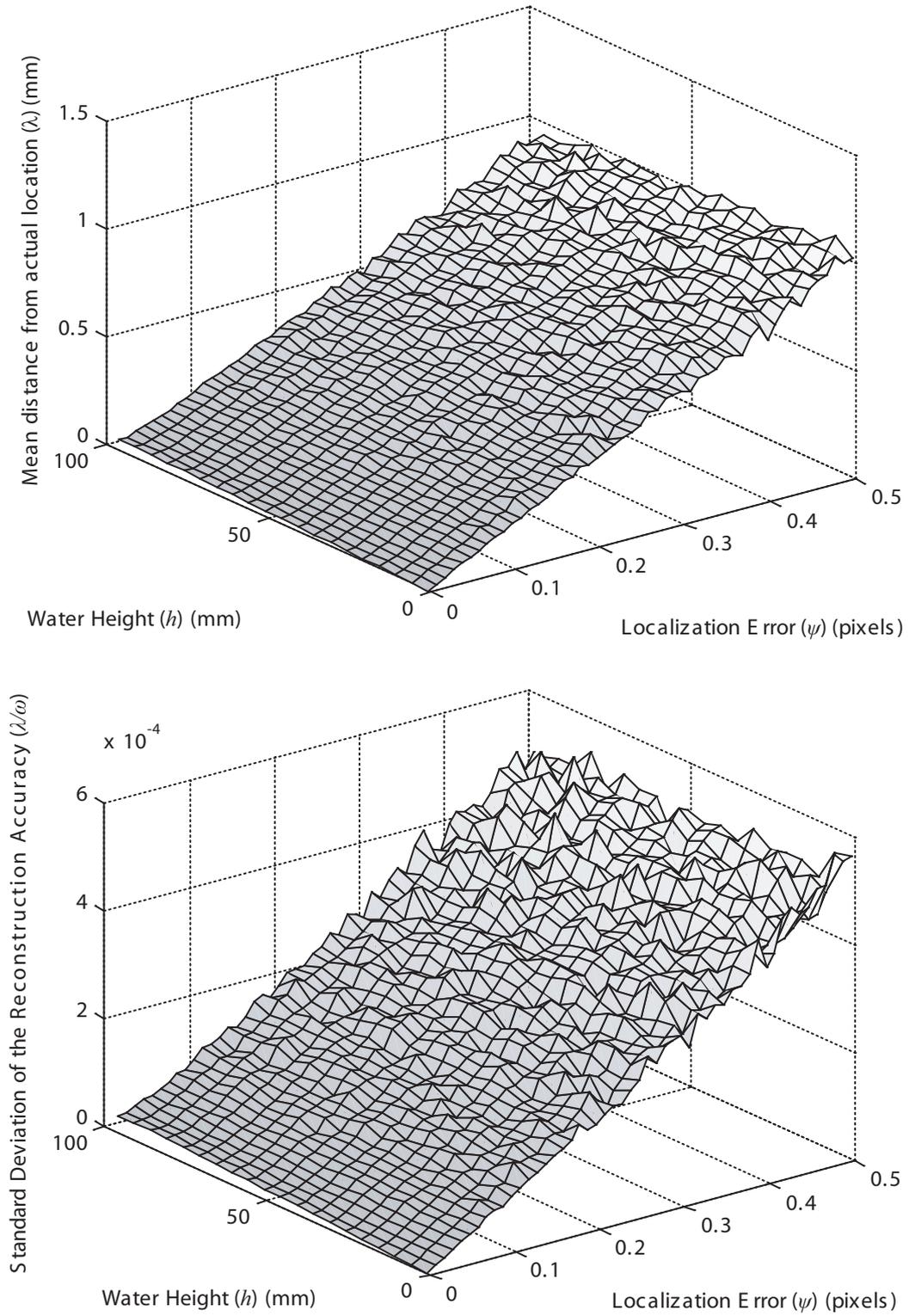


Figure 4.10: Simulation graphs showing the mean distance between reconstructed points and the actual points (top) and the standard deviation the depth reconstruction accuracy (bottom) for varying depths and localization errors

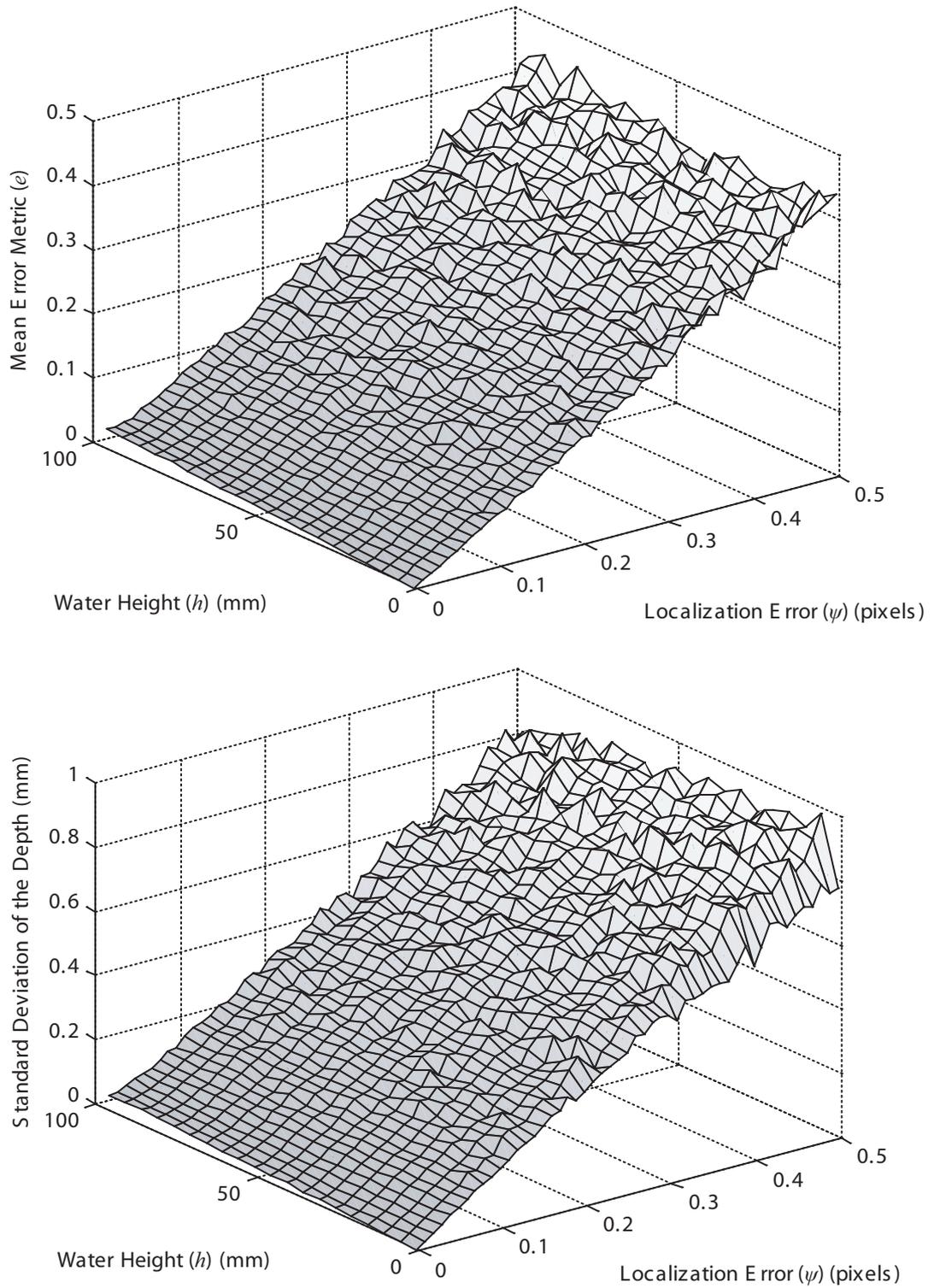


Figure 4.11: Simulation graphs showing the mean error metric (top) and the standard deviation of the reconstructed depths (bottom) for varying depths and localization errors

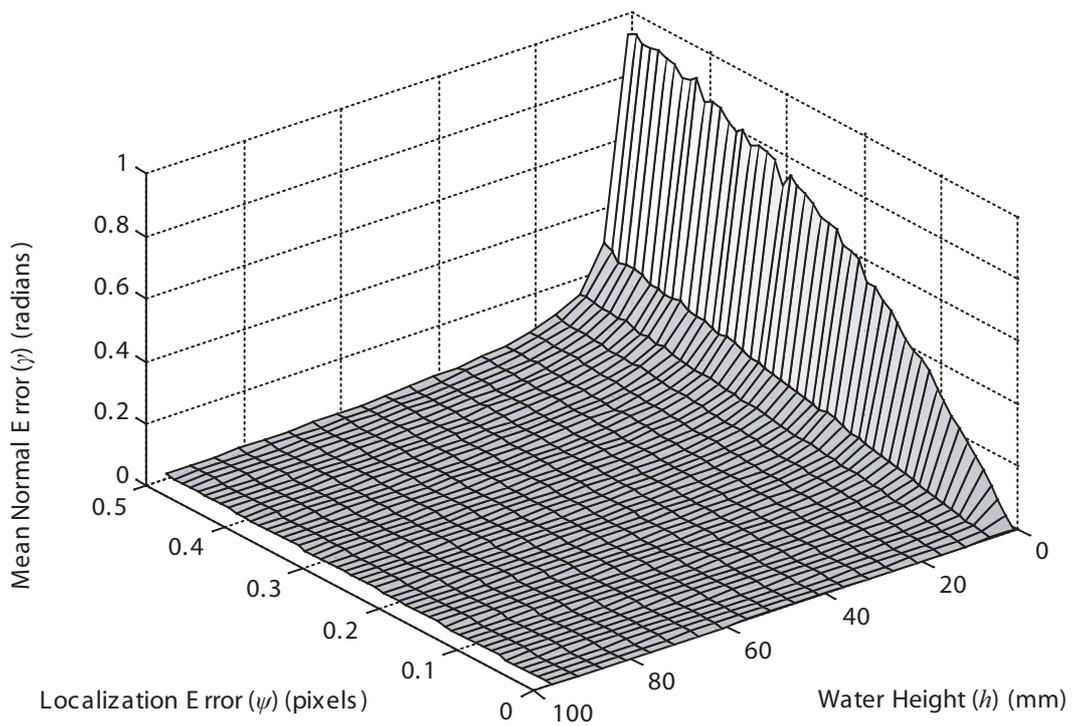
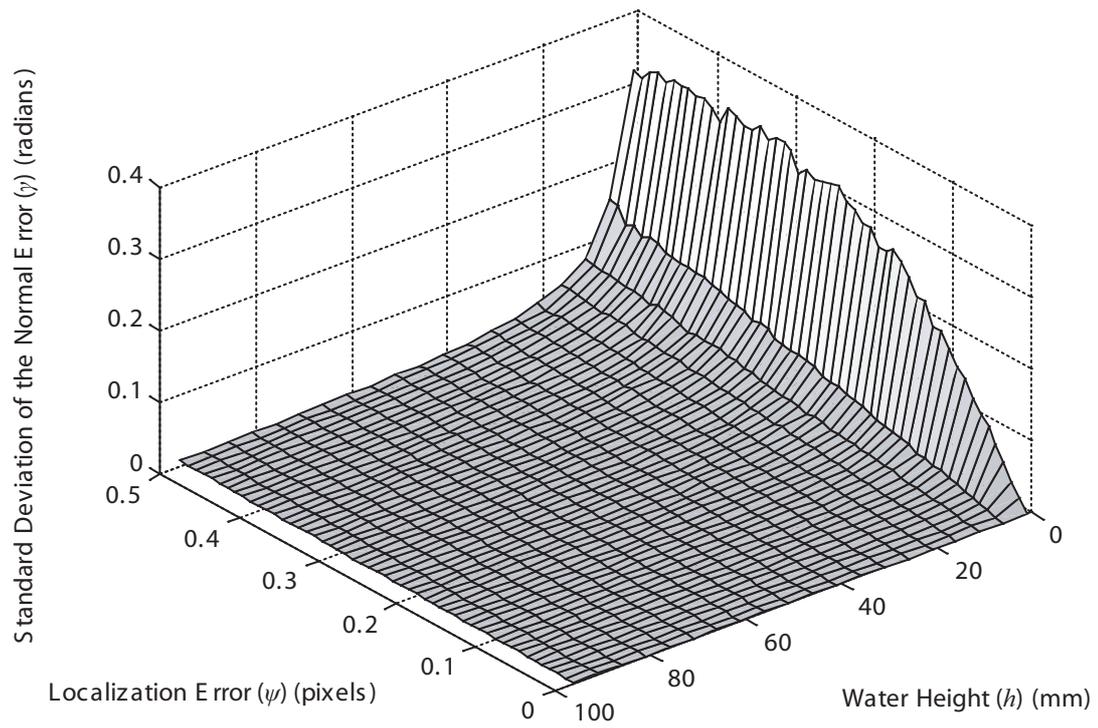


Figure 4.12: Simulation graphs showing the mean and standard deviation of the normal error for varying depths and localization errors

precision of our system, but is not able to determine the accuracy. Our experiments revealed a precision of ~ 0.1 pixels.

We ran our simulation using the computed calibration error parameter for several values of localization error. We varied the localization error from 0.6 pixels to 1.2 pixels in 0.2 pixel increments. Figure 4.13 shows the comparison between simulation results and our results from observation. Our empirical results closely match the simulation results in every category. However the localization error appears to be roughly 0.6-1.2 pixels larger than the precision of 0.1 pixels. The same characteristic increase in normal error is found as the depth decreases. The distance gauges show a similar robustness to water height and the error range to the corresponding simulation results.

4.3 Water surface sequences

We reconstructed several sequences of captured flowing water. For each of these sequences the input to our algorithm was a stereo view of a chequered pattern over which water was poured.

The first two sequences were captured during the actual pouring of the water onto the pattern area. In both cases the water depths were low, beginning at approximately 1-2mm deep and rising as more water was added. We label these sequences: POUR-A and POUR-B.

Figures 4.14 to 4.17 show four frames from sequence POUR-A, along with the corresponding input images of the pattern from both cameras. This sequence used a pattern checker size of approximately 4mm.

Figures 4.18 to 4.21 show four frames from the second sequence POUR-B, along with the corresponding input images of the pattern as before. This sequence was rendered with ray traced refraction and reflection, with a textured plane beneath the water so that the results can be compared more closely to input. This sequence used a pattern

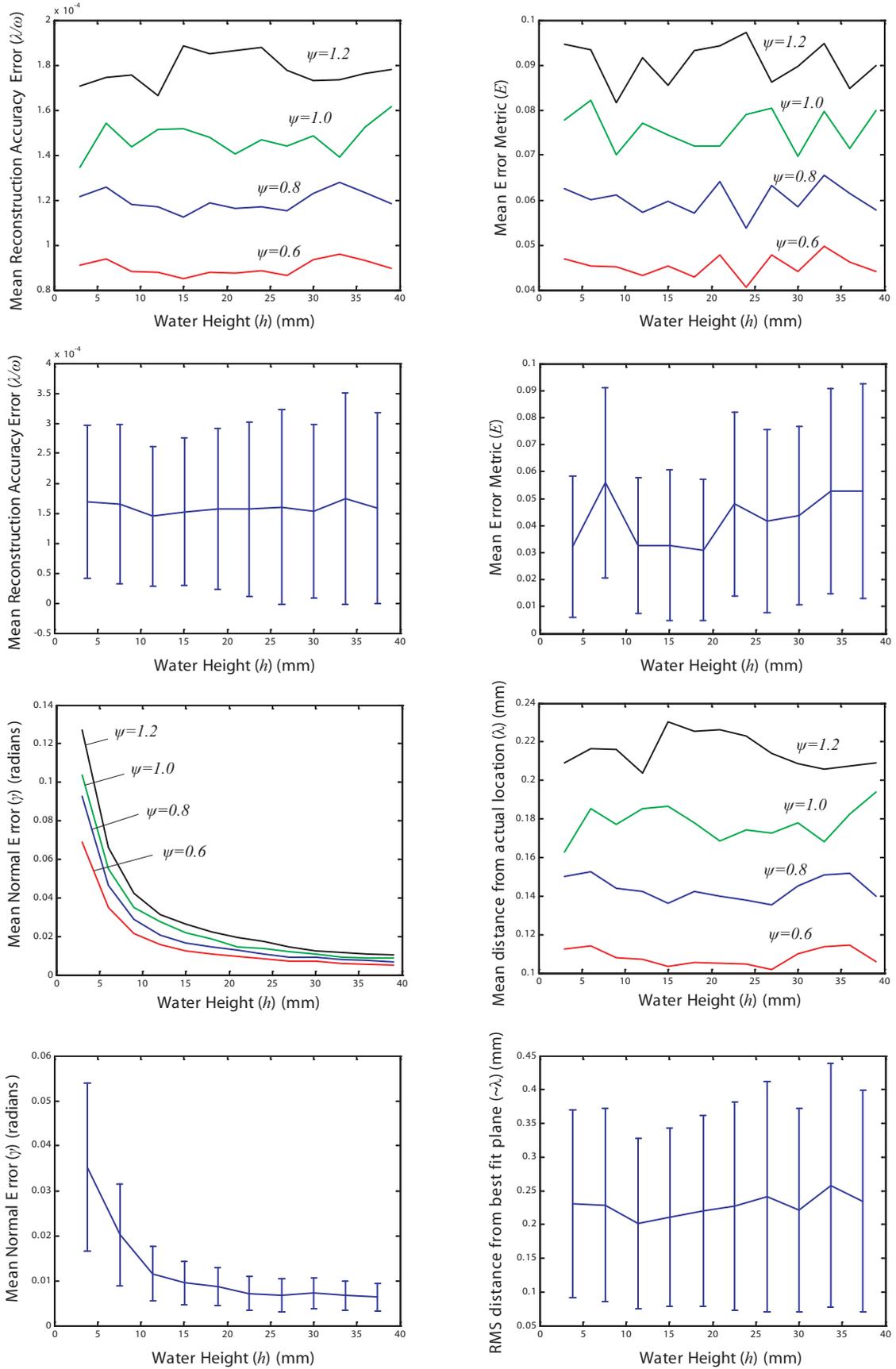


Figure 4.13: Simulation results are shown above the corresponding empirical results for four values of the localization error (ψ)

checker size of approximately 3mm. Notice that this sequence has some bubbles on the water surface (Figure 4.19). The bubbles cause indentations in the water surface and the reconstruction correctly models this.

Often the subtleties of the reconstruction cannot be seen without viewing animations of the resulting sequences. In some of our reconstructions, low amplitude waves are seen to propagate through the reconstructed surfaces that cannot be detected in single images².

Our next reconstructed sequence is labelled as RIPPLE. It consists of the reconstruction of the surface after a few drops are dripped into water several centimetres deep. We were unable to reconstruct the initial splash as the pattern was too distorted for the corners to be matched correctly (shown in Figure 4.22). Had the water depth been lower the initial splash would have been easier to reconstruct since less elimination would have occurred. The reconstruction checker size was 3mm for this sequence.

We present one frame of the RIPPLE sequence in Figure 4.23. This figure shows the set of reconstructed points as well as a rendered mesh of the frame. Notice the sparse areas on the left and right edges of the reconstructed point set. These areas are the results of overlapping as described in Section §4.1.1. Although these areas cannot be reconstructed as accurately, their locations can be estimated using the nearest verification features as shown.

²We refer the reader to the resulting animations that are available here: <http://www.dgp.toronto.ed/~nmorris/thesis/>

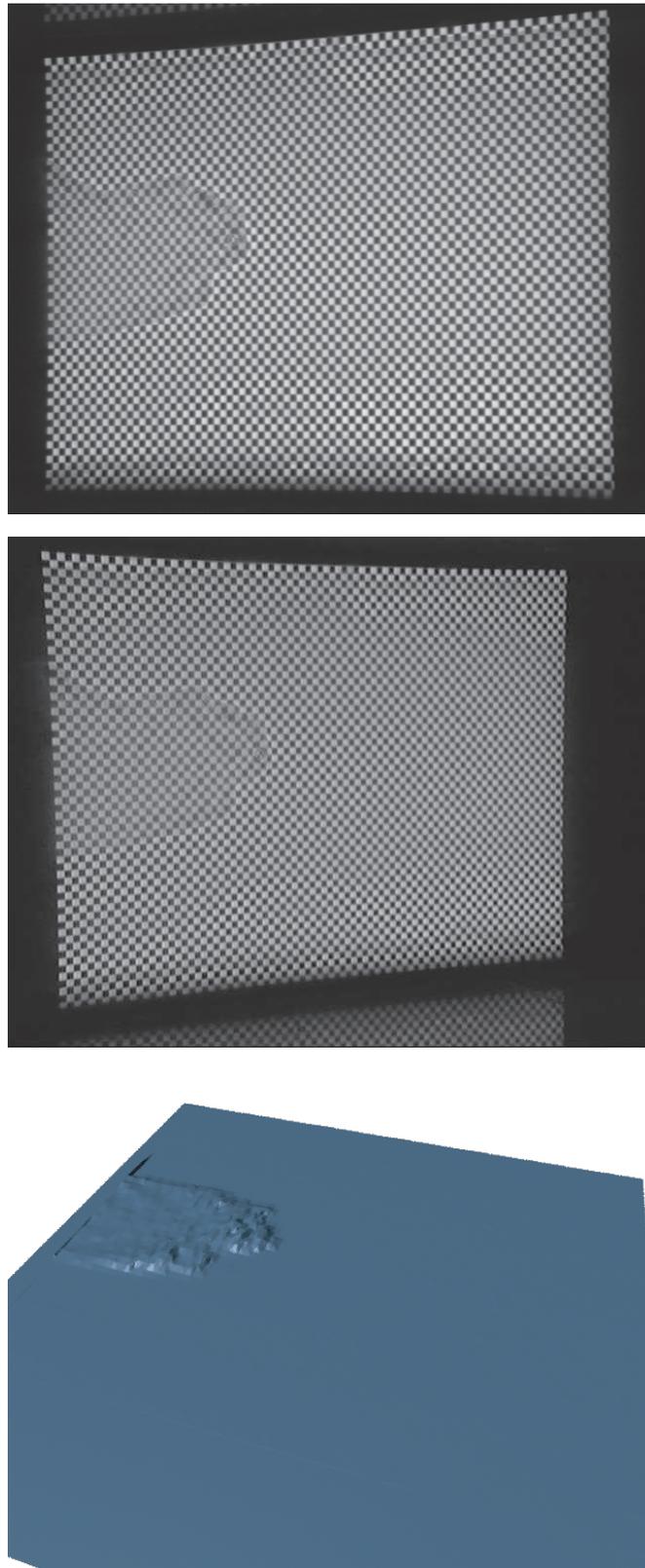


Figure 4.14: Frame of sequence POUR-A. The top two rows are the stereo views of the water. The bottom row is the reconstructed surface.

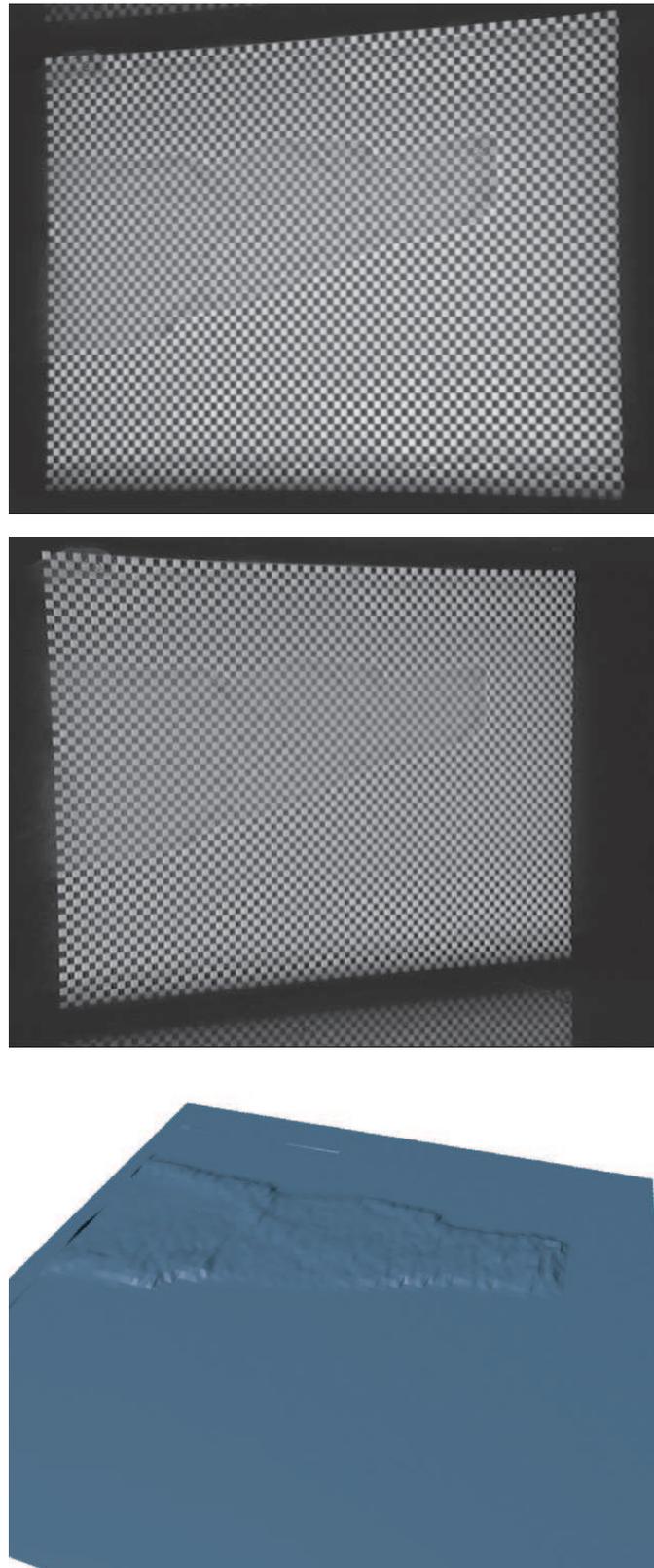


Figure 4.15: Frame of sequence POUR-A. The top two rows are the stereo views of the water. The bottom row is the reconstructed surface.

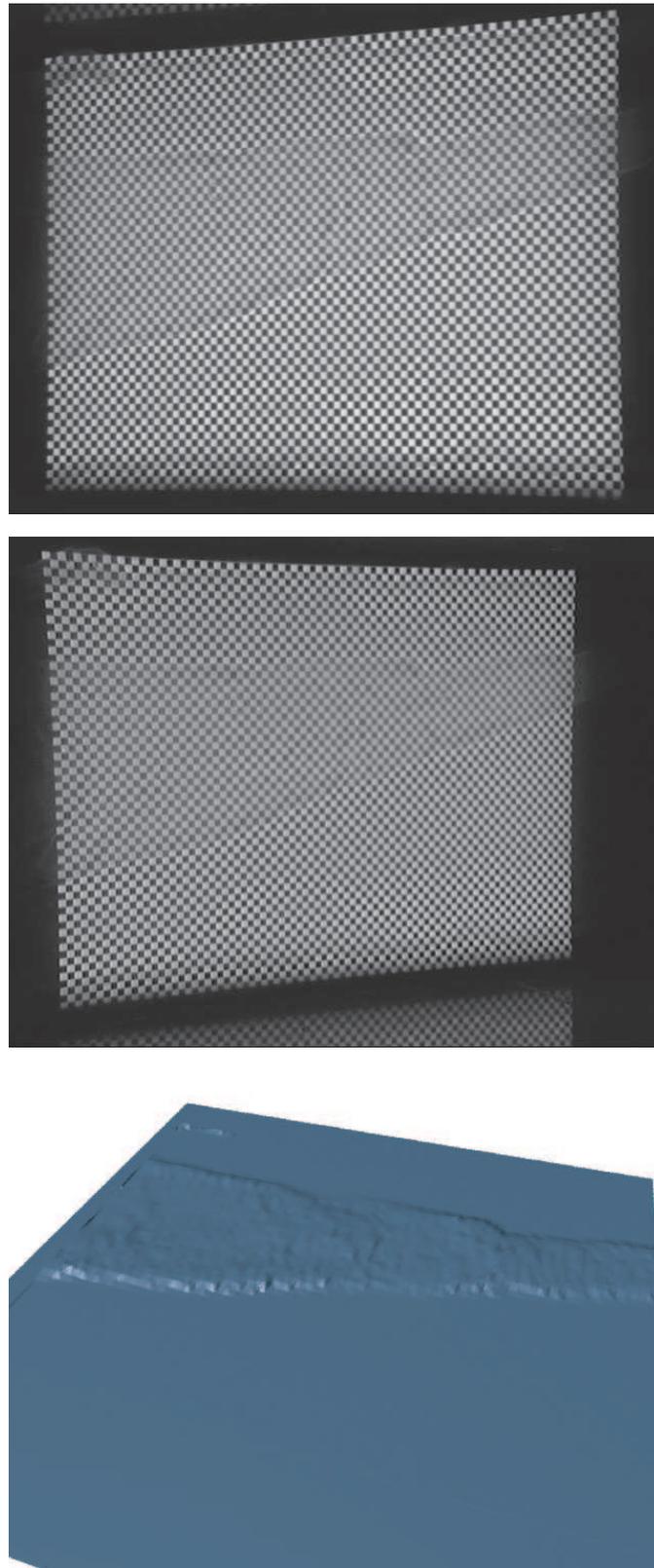


Figure 4.16: Frame of sequence POUR-A. The top two rows are the stereo views of the water. The bottom row is the reconstructed surface.

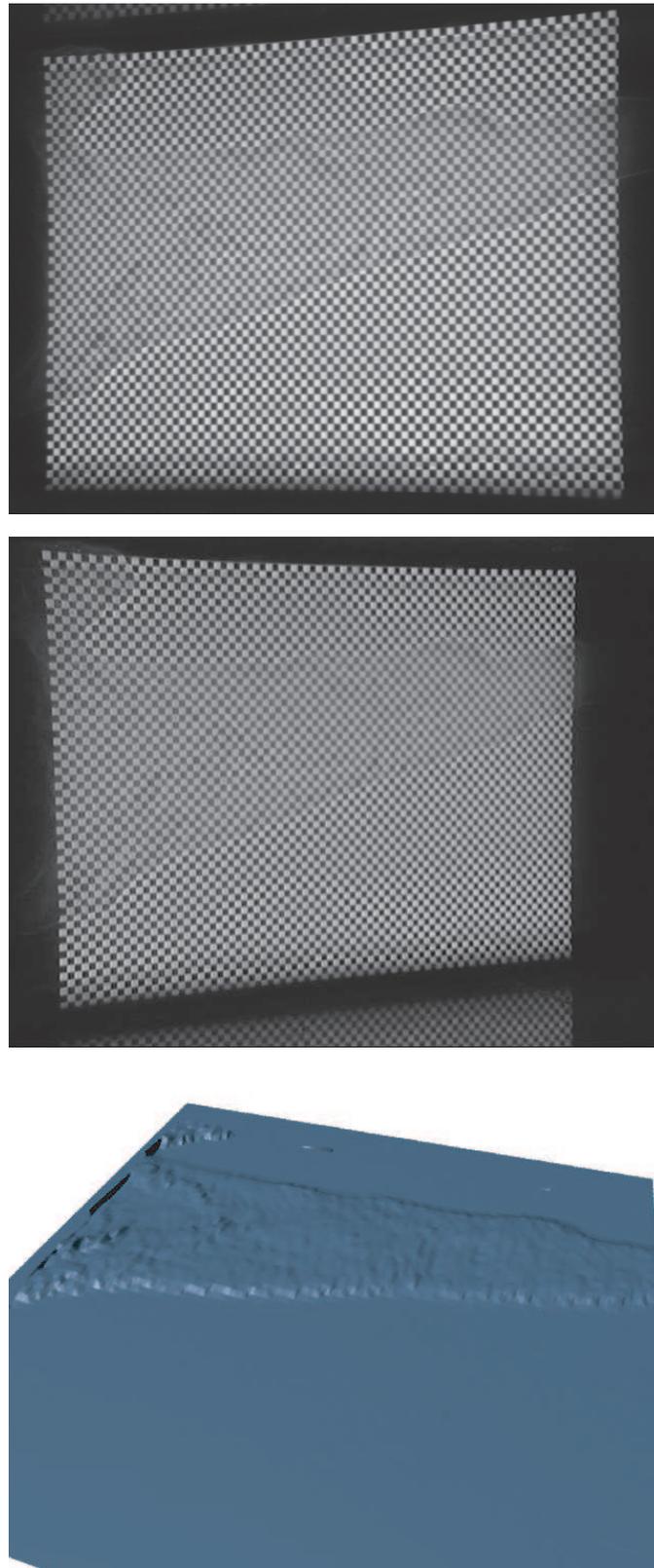


Figure 4.17: Frame of sequence POUR-A. The top two rows are the stereo views of the water. The bottom row is the reconstructed surface.

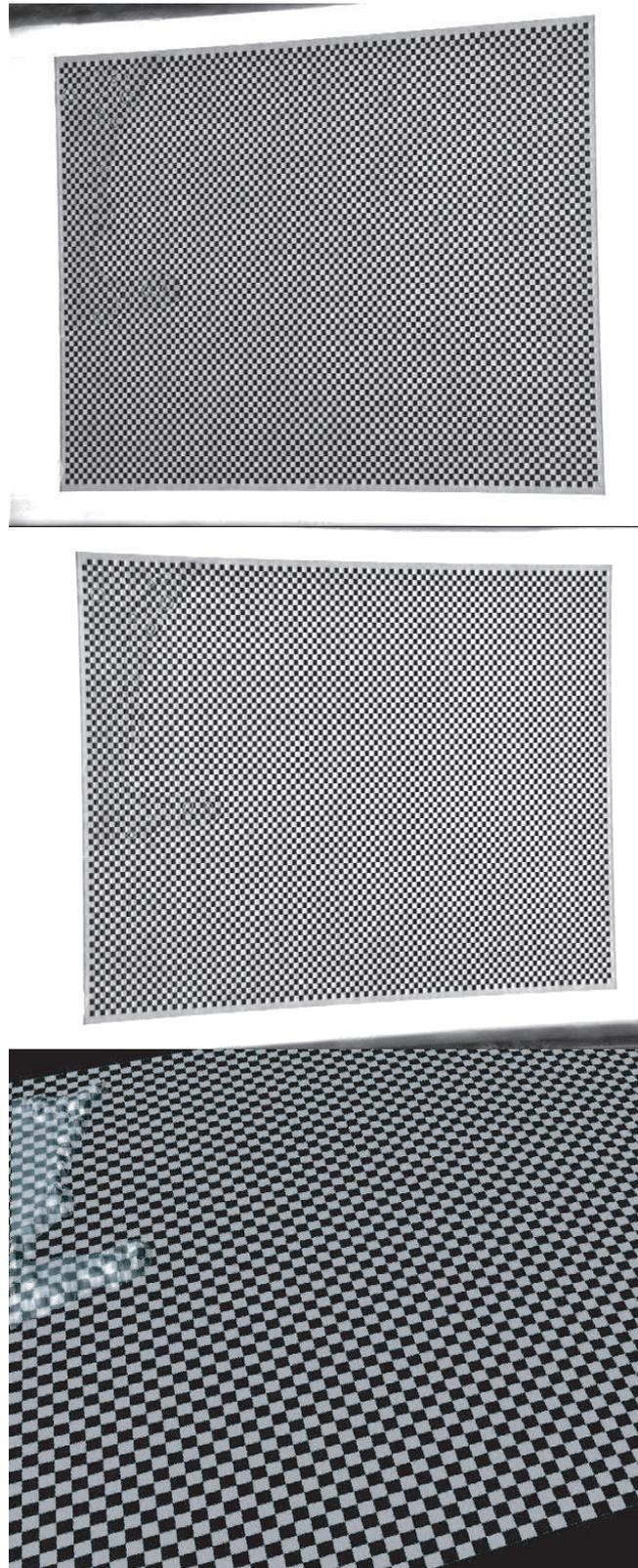


Figure 4.18: Frame of sequence POUR-B. The top two rows are the stereo views of the water. The bottom row is the reconstructed surface.

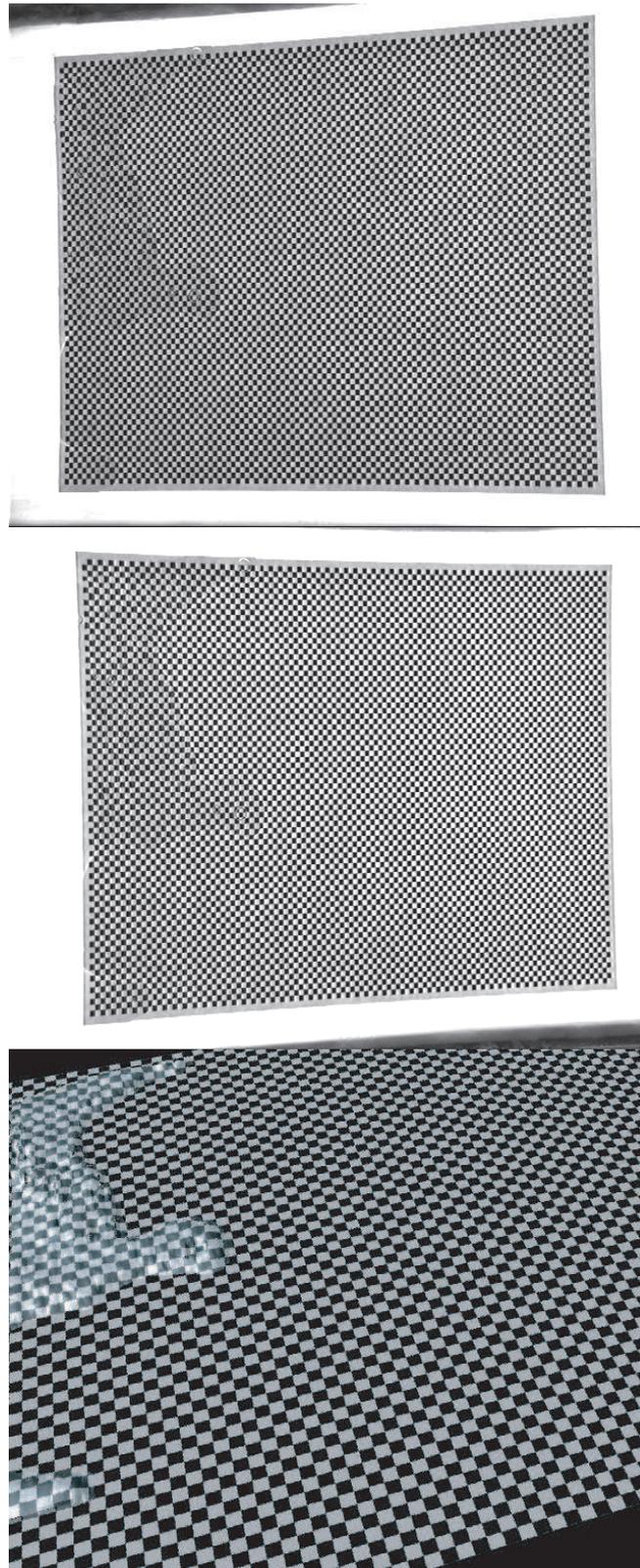


Figure 4.19: Frame of sequence POUR-B. The top two rows are the stereo views of the water. The bottom row is the reconstructed surface.

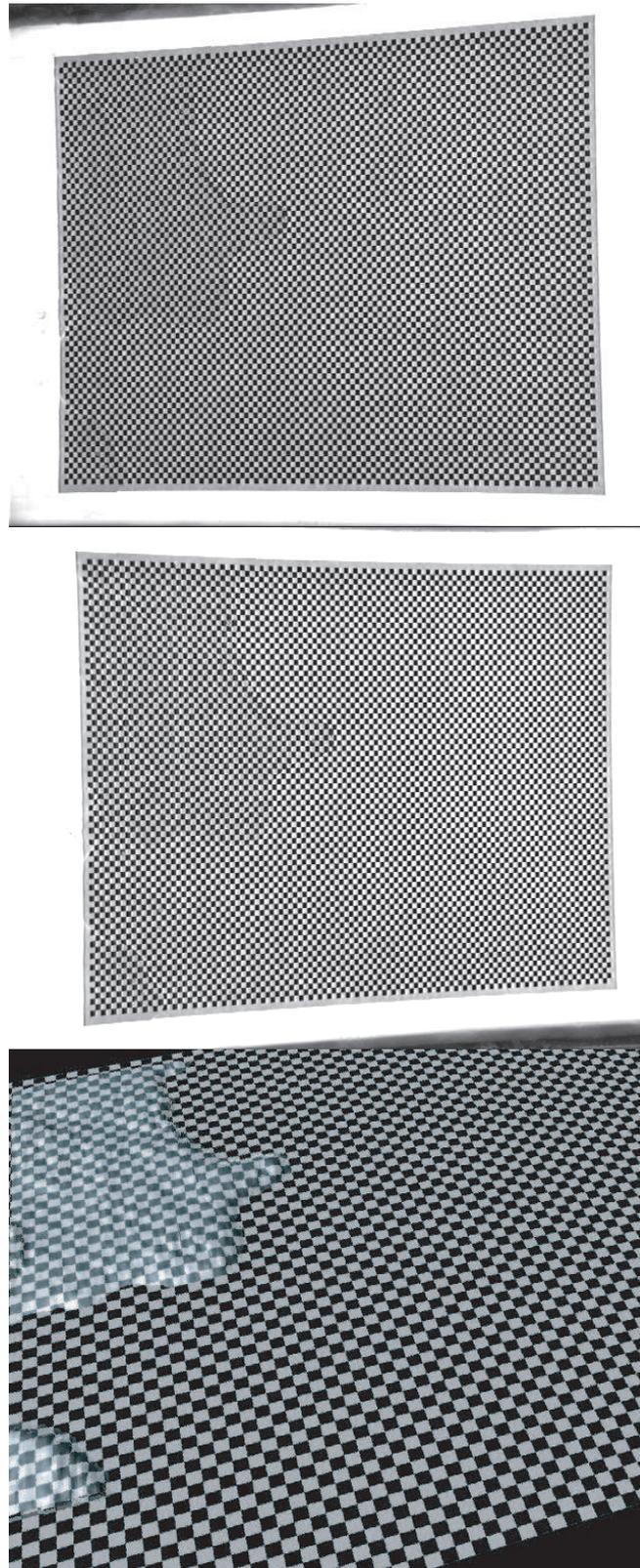


Figure 4.20: Frame of sequence POUR-B. The top two rows are the stereo views of the water. The bottom row is the reconstructed surface.

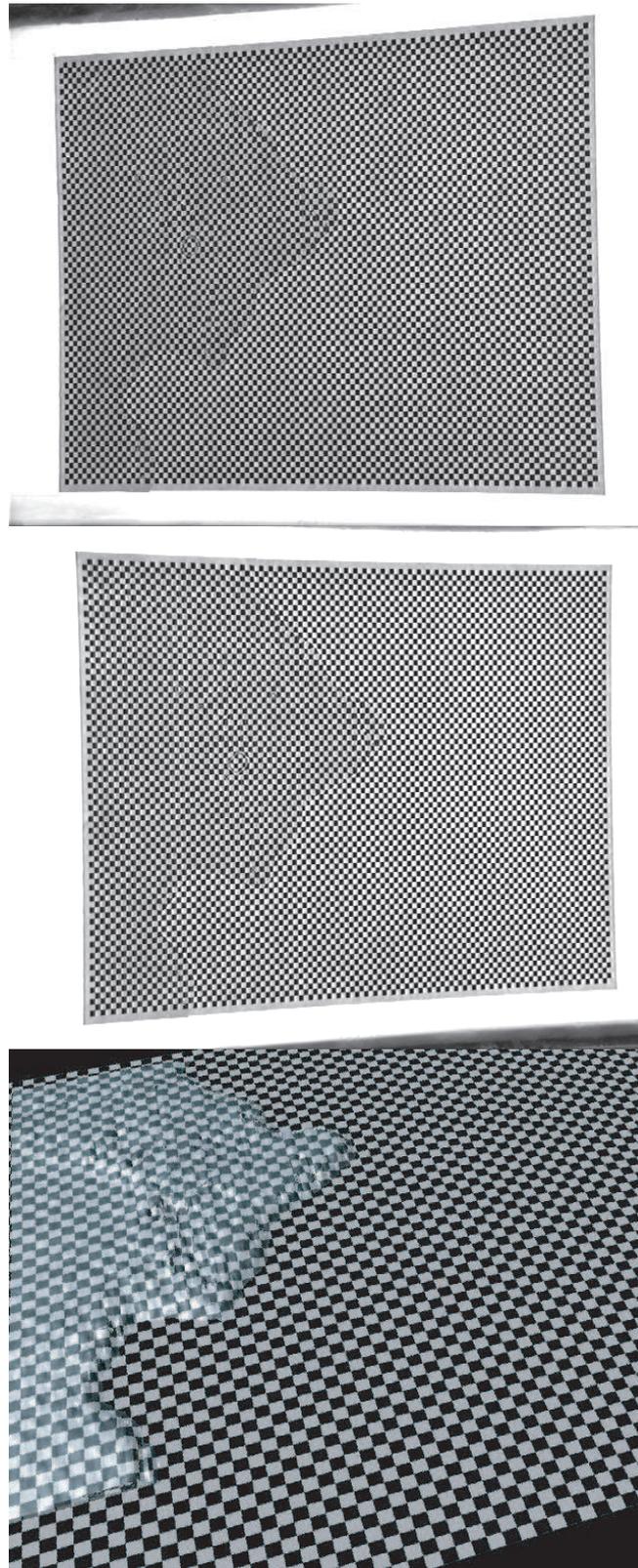


Figure 4.21: Frame of sequence POUR-B. The top two rows are the stereo views of the water. The bottom row is the reconstructed surface.

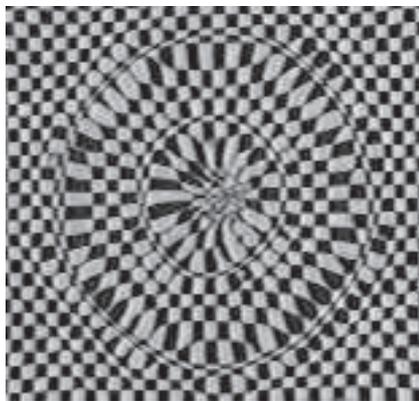


Figure 4.22: Image of the pattern distorted by a splash in the water. This pattern has too much elimination for our reconstruction algorithm to localize enough of the corners for a reasonable reconstruction.

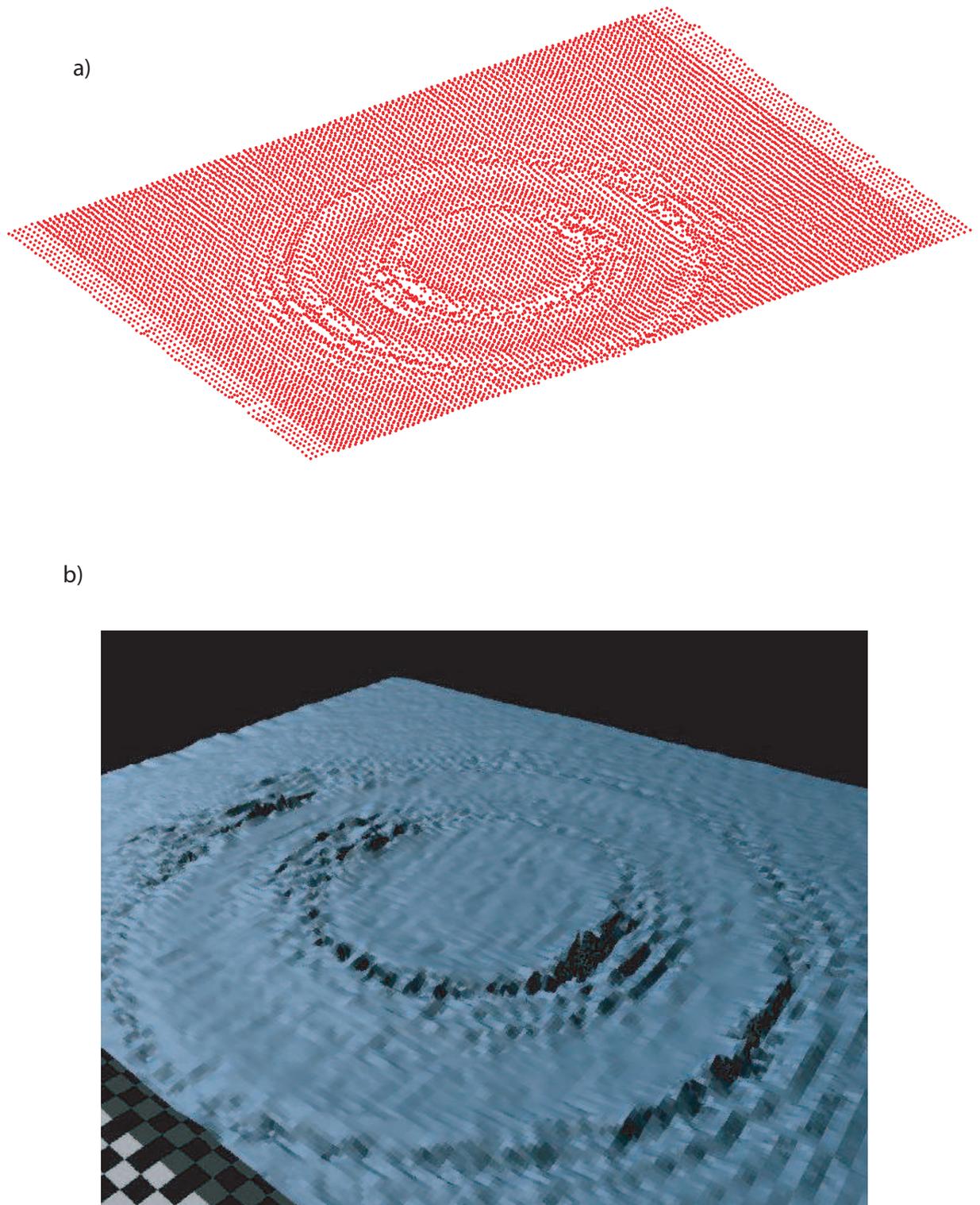


Figure 4.23: a) Shows the reconstructed set of points from one frame of the RIPPLE sequence. b) Shows the rendered mesh of the above point set.

Chapter 5

Conclusion

“The cure for anything is salt water - sweat, tears, or the sea.”

-Isak Dinesen

We have presented a new system for reconstructing the surface of water, utilizing stereo images of a pattern refracted through the water. Our system builds upon work that utilizes refractive distortion as well as stereo reconstruction research. We have provided a theoretical outline of the algorithm that combines these two methods. An implementation of our system was also presented. The implementation only requires a simple stereo camera setup with no additional equipment.

We generated input data from a simulation and showed that the simulation results were consistent with our empirical data. We also proposed two matching metrics for determining points of the water surface. We showed that our disparity difference metric outperformed the normal collinearity metric when the water depth approached the size of the localization error.

We discovered that the localization error of pattern feature points contributes the most to the error in water surface point determination, especially when the water is still. The calibration error is expected to affect reconstruction accuracy to a greater extent when the water is disturbed.

Our system is built to allow the reconstruction of sequences of flowing water and our results show that it is especially effective at reconstructing shallow flows. At greater water depths the trade-off between the pattern density and the surface roughness that can be captured is more noticeable.

While our system is described specifically for water, the technique described here can readily be applied to other liquids by specifying different refractive indices.

There are several avenues available for improving and extending our system. We outline them in the next section.

5.1 Future Work

Currently our system is based upon finding individual points on the water surface. In order to improve the overall smoothness we propose that a global method could be applied so that the surface is determined by global minimization of the whole set of points. It may also be feasible to attach a temporal smoothness term to our surface generation, to eliminate outliers that suddenly appear in a sequence.

Our system currently cannot handle splashing water. While it would be beneficial to enhance the robustness of our surface determination to splashes, it would also be interesting to capture such effects. We propose that a volume carving approach could be applied to the splashing water in order to incorporate it with the generated surface.

Another enhancement to our system would be to remove the constraint of a planar surface underneath the water. We believe that it would be possible to reconstruct the ground surface below the water as well as the water surface given sufficient views of the surfaces.

We foresee that this work may be used as a key piece in several larger bodies of work. First, the determination of internal fluid flow from images would certainly require precise knowledge of the surface topology, presenting a vital application for our work. Another

use for this thesis may be in the collection of a library of liquid flows that may be used as a tool to compose arbitrary flows.

Appendix A

Simulation algorithm

Here we present the details for our simulation algorithm. This algorithm takes in parameters for the calibration error range and localization error range and generates the appropriate inputs for the depth verification algorithm. It returns a result set for the input parameter ranges.

Algorithm 5: Simulation with constant height

Input: Reconstruction height (z), Calibration error range ($calibErrMin$, $calibErrStep$, $calibErrMax$), Localization error range ($localErrMin$, $localErrStep$, $localErrMax$), Virtual camera centres of projection \mathbf{c}_1 and \mathbf{c}_2 , Tank bottom plane T , $numIterations$

Result : Behaviour gauges

for $\rho \leftarrow calibErrMin$; $\rho < calibErrMax$; $\rho+ = calibErrStep$ **do**

for $\psi \leftarrow localErrMin$; $\psi < localErrMax$; $\psi+ = localErrStep$ **do**

 Pick image coordinates \mathbf{q}_1 of feature \mathbf{f}_1 from Camera \mathbf{c}_1 ;

 Determine actual surface location $\mathbf{p} = \mathbf{c}_1 + z\|\mathbf{q}_1 - \mathbf{c}_1\|$;

$i \leftarrow 0$

while $i < numIterations$ **do**

 Shift \mathbf{q}_1 by a random amount around a mean of ψ to get $\mathbf{q}_1 + \Delta\mathbf{q}_1$;

 Determine the adjusted surface point $\mathbf{p} + \Delta\mathbf{p}$;

 Intersect $\mathbf{p} + \Delta\mathbf{p} - \mathbf{c}_2$ with T to find the virtual feature \mathbf{f}_2 ;

 Shift \mathbf{f}_2 by a random amount around a mean of ρ to get $\mathbf{f}_2 + \Delta\mathbf{f}_2$;

 Project $\mathbf{f}_2 + \Delta\mathbf{f}_2$ to \mathbf{c}_2 to get image coordinates \mathbf{q}_2 ;

 Shift \mathbf{q}_2 by a random amount around a mean of ψ to get $\mathbf{q}_2 + \Delta\mathbf{q}_2$;

 Compute the shifted images of the features without water;

 Minimize DepthVerificationSimulation to find the expected best depth and error metric result;

$i = i + 1$;

 Average expected best depths and error metric results;

Return data structure of averaged expected best depths, error metrics, expected normals, actual depths and actual normals;

Bibliography

- [1] E. H. Adelson and J. R. Bergen. The plenoptic function and the elements of early vision. *M. Landy and J. A. Movshon, (eds) Computational Models of Visual Processing*, 1991.
- [2] Reginald D. Arnold. Automated stereo perception. Technical Report AIM-351, Artificial Intelligence Laboratory, Stanford University, 1983.
- [3] M. Baba, K. Ohtani, and M. Imai. New laser rangefinder for three-dimensional shape measurement of specular objects. *Optical Engineering*, 40(1):53–60, 2001.
- [4] H. H. Baker. *Image Understanding Workshop*. Science Applications International Corporation, 1980.
- [5] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. *IJCV*, 56(3):221–255, 2004.
- [6] S. T. Barnard. Stochastic stereo matching over scale. *IJCV*, 3(1):17–32, 1989.
- [7] Peter N. Belhumeur. A bayesian approach to binocular stereopsis. *IJCV*, 19(3):237–260, 1996.
- [8] Peter N. Belhumeur and D. Mumford. A bayesian treatment of the stereo correspondence problem using half-occluded regions. In *CVPR*, pages 506–512, 1992.
- [9] Moshe Ben-Ezra and Shree K. Nayar. What does motion reveal about transparency? In *Proceedings of ICCV 2003*, pages 1025–1032, 2003.

- [10] D.N. Bhat and S.K. Nayar. Stereo in the presence of specular reflection. In *ICCV95*, pages 1086–1092, 1995.
- [11] D.N. Bhat and S.K. Nayar. Stereo and specular reflection. *IJCV*, 26(2):91–106, February 1998.
- [12] Stan Birchfield and Carlo Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(4):401–406, 1998.
- [13] Michael J. Black and P. Anandan. A framework for the robust estimation of optical flow. In *ICCV*, pages 231–236, 1993.
- [14] Michael J. Black and A. Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *IJCV*, 19(1):57–91, 1996.
- [15] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, Cambridge, MA, 1987.
- [16] A. F. Bobick and S. S. Intille. Large occlusion stereo. *IJCV*, 33(3):181–200, 1999.
- [17] Thomas Bonfort and Peter Sturm. Voxel carving for specular surfaces. In *Proceedings of the 9th International Conference on Computer Vision*, volume 9, 2003.
- [18] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.
- [19] J. F. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):34–43, 1986.

- [20] Jin-Xiang Chai, Shing-Chow Chan, Heung-Yeung Shum, and Xin Tong. Plenoptic sampling. In *Proceedings of SIGGRAPH 2000, ACM Press / ACM SIGGRAPH*, pages 307–318, 2000.
- [21] Yung-Yu Chaung, Douglas E. Zongker, Joel Hindorrd, Brian Curless, David H. Salesin, and Richard Szeliski. Environment matting extensions: Towards higher accuracy and real-time capture. In *Proceedings of ACM SIGGRAPH 2000, Computer Graphics Proceedings, Annual Conference Series*, pages 121–130. ACM Press / ACM SIGGRAPH / Addison Wesley Logman, July 2000. ISBN 1-58113-208-5.
- [22] I. J. Cox, S. L. Hingorani, S. B. Rao, and B. M. Maggs. A maximum likelihood stereo algorithm. *CVUI*, 63(3):542–567, 1996.
- [23] J.M. Daida, D. Lund, C. Wolf, G.A. Meadows, K. Schroeder, J. Vesecky, D.R. Lyzenga, B.C. Hannan, and R.R. Bertram. Measuring topography of small-scale water surface waves. In *Geoscience and Remote Sensing Symposium. Conference Proceedings*, volume 3, pages 1881–1883, 1995.
- [24] P. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin, and M. Sagar. Acquiring the reflectance field of a human face. In *Computer Graphics, SIGGRAPH 2000 Proceedings*, pages 145–156, 2000.
- [25] Douglas Enright, Ronald Fedkiw, Joel Ferziger, and Ian Mitchell. A hybrid particle level set method for improved interface capturing. In *Proceedings of SIGGRAPH 2002, ACM Press / ACM SIGGRAPH*, 2002.
- [26] Douglas Enright, Stephen Marschner, and Ronald Fedkiw. Animation and rendering of complex water surfaces. In *Proc. Siggraph 2002*, pages 736–744, 2002.
- [27] O. Faugeras and R. Keriven. Complete dense stereovision using level set methods. In *Proceedings of the 5th ECCV*, pages 379–393, 1998.

- [28] Nick Foster and Ronald Fedkiw. Practical animation of liquids. In *Proceedings of SIGGRAPH 2001, ACM Press / ACM SIGGRAPH*, pages 23–30, 2001.
- [29] Nick Foster and Dimitri Metaxas. Realistic animation of liquids. In *Graphical models and image processing*, 1995.
- [30] P. Fua. Aparallel stereo algorithm that produces dense depth maps and preserves image features. *Machine Vision and Applications*, 6:35–49, 1993.
- [31] D. Geiger and F. Girosi. Parallel and deterministic algorithms for mrfs: Surface reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):401–412, 1991.
- [32] D. Geiger, B. Ladendorf, and A. Yuille. Occlusions and binocular stereo. In *ECCV*, pages 425–433, 1992.
- [33] S. Geman and D. Geman. Stochastic relaxation, gibbs distribution, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.
- [34] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. *Computer Graphics*, 30(Annual Conference Series):43–54, 1996.
- [35] W. E. L. Grimson. Computational experiments with a feature based stereo algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(1):17–34, 1985.
- [36] Mark A. Halstead, Brian A. Barsky, Stanley A. Klein, and Robert B. Mandell. Reconstructing curved surfaces from specular reflection patterns. In *Proceedings of SIGGRAPH 1996*, pages 335–342, 1996.
- [37] M. J. Hannah. *Computer Matching of Areas in Stereo Images*. PhD thesis, Standord University, 1974.

- [38] F.H. Harlow and J.E. Welch. Numerical calculation of time-dependent viscous incompressible flow. *Phys. Fluids*, 8:2182–2189, 1965.
- [39] Janne Heikkila and Olli Silven. A four-step camera calibration procedure with implicit image correction. In *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, page 1106. IEEE Computer Society, 1997.
- [40] B. K. P. Horn. Shape from shading: a method for obtaining the shape of a smooth opaque object from one view. Technical Report 232, MIT Artificial Intelligence Laboratory, Cambridge, MA, 1970.
- [41] Bernd Jähne, Jochen Klinke, Peter Geissler, and Frank Hering. Image sequence analysis of ocean wind waves. In *Proceedings of the International Seminar on Imaging in Transport Processes*, 1992.
- [42] Bernd Jähne, Jochen Klinke, and Stefan Waas. Imaging of short ocean wind waves: a critical review. *Journal of Optical Society of America*, 11(8):2197–2209, 1994.
- [43] A. C. Kak and M. Slaney. *Principles of Copmputerized Tomographics Imaging*. IEEE Press, New York, 1988.
- [44] Takeo Kanade. Development of a video-rate stereo machine. In *Image Understanding Workshop*, pages 549–557, 1994.
- [45] S. Kang, R. Szeliski, and J. Chai. Handling occlusions in dense multiview stereo. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 103–110.
- [46] Michael Kass and Gavin Miller. Rapid, stable fluid dynamics for computer graphics. In *Proceedings of SIGGRAPH 1990, ACM Press / ACM SIGGRAPH*, pages 49–55, 1990.
- [47] W. C. Keller and B. L. Gotwols. Two-dimensional optical measurement of wave slope. *Applied Optics*, 22(22):3476–3478, 1983.

- [48] Marc Levoy and Pat Hanrahan. Light field rendering. *Computer Graphics*, 30(Annual Conference Series):31–42, 1996.
- [49] A. Manessis, A. Hilton, P. Palmer, P. McLauchlan, and X. Shen. Reconstruction of scene models from sparse 3d structure. In *CVPR*, volume 2, pages 666–671, 2000.
- [50] J. Marroquin, S. Mitter, and T. Poggio. Two-dimensional optical measurement of wave slope. *Journal of the American Statistical Association*, 82(397):76–89, 1987.
- [51] G. Masten, P. Watterberg, and I. Mareda. Fourier synthesis of ocean scenes. *IEEE Computer Graphics and Application*, 7:16–23, 1987.
- [52] Wojciech Matusik, Hanspeter Pfister, Remo Ziegler, Addy Ngan, and Leonard McMillan. Acquisition and rendering of transparent and refractive objects. In *Eurographics '02 Proceedings*, 2002.
- [53] Daisuke Miyazaki, Masataka Kagesawa, and Katsushi Ikeuchi. Transparent surface modeling from a pair of polarization images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):73–82, 2003.
- [54] Daniel D. Morris and Takeo Kanade. Image-consistent surface triangulation. In *CVPR*, volume 1, pages 332–338, 2000.
- [55] Matthias Müller, David Charypar, and Markus Gross. Particle-based fluid simulation for interactive applications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 154–159, 2003.
- [56] Hiroshi Murase. Shape reconstruction of an undulating transparent object. In *Proc. IEEE Intl. Conf. Computer Vision*, pages 313–317, 1990.
- [57] S.K. Nayar, X.S. Fang, and T.E. Boult. Separation of reflection components using color and polarization. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 585–590, 1993.

- [58] Masatoshi Okutomi and Takeo Kanade. A locally adaptive window for signal matching. In *ICCV*, pages 190–199, 1990.
- [59] Masatoshi Okutomi and Takeo Kanade. A multi-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):355–363, 1993.
- [60] D. Peachy. Modeling waves and surf. In *Proceedings of SIGGRAPH 1986, ACM Press / ACM SIGGRAPH*, pages 65–74, 1986.
- [61] S. B. Pollard, J. E. W. Mayhew, and J. P. Frisby. Pmf: A stereo correspondence algorithm using a disparity gradient limit. *Perception*, 14:449–470, 1985.
- [62] K. Prazdny. Detection of binocular disparities. *Biological Cybernetics*, 52(2):93–99, 1985.
- [63] Stefan Rahmann and Nikos Canterakis. Reconstruction of specular surfaces using polarization. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 149–155, 2001.
- [64] Silvio Savarese and Pietro Perona. Local analysis for 3d reconstruction of specular surfaces - part ii. In *Proceedings of the 7th ECCV*, pages 759–774, 2002.
- [65] Daniel Scharstein and Richard Szeliski. Stereo matching with nonlinear diffusion. *International Journal of Computer Vision*, 28(2):155–174, 1998.
- [66] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1/2/3):7–42, 2001.
- [67] Howard Schultz. Retrieving shape information from multiple images of a specular surface. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):195–201, 1994.

- [68] J. Shah. A nonlinear diffusion model for discontinuous disparity and half-occlusion in stereo. In *CVPR*, pages 34–40, 1993.
- [69] O.H. Shemdin. Measurement of short surface waves with stereophotography. In *Engineering in the Ocean Environment. Conference Proceedings*, pages 568–571, 1990.
- [70] Karl Sims. Particle animation and renderings using data parallel computation. *ACM Transactions on Graphics*, 24(4):405–413, 1990.
- [71] Jos Stam. Stable fluids. In *Proceedings of SIGGRAPH 1999, ACM Press / ACM SIGGRAPH*, pages 121–128, 1999.
- [72] S. A. Sullivan. Experimental study of the absorption in distilled water, artificial sea water, and heavy water in the visible region of the spectrum. *Opt. Soc. Am. J.*, 53:962–968, 1963.
- [73] Demetri Terzopoulos. Regularization of inverse visual problems involving discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(4):413–424, 1986.
- [74] Adrien Treuille, Aaron Hertzmann, and Steven M. Seitz. Example-based stereo with general brdfs. In *Proceedings of European Conference on Computer Vision (In Press)*, 2004.
- [75] Yanghai Tsin, Sing Bing Kang, and Richard Szeliski. Stereo matching with reflections and translucency. In *IEEE Computer Vision and Pattern Recognition (CVPR'03)*, June 2003.
- [76] Pauline Y. Ts'o and Brian A. Barsky. Modeling and rendering waves: wave-tracing using beta-splines and reflective and refractive texture mapping. *ACM Transactions on Graphics*, 6(3):191–214, 1987.

- [77] Zhijian Wu and Guy A. Meadows. 2-d surface reconstruction of water waves. In *Engineering in the Ocean Environment. Conference Proceedings*, pages 416–421, 1990.
- [78] Xin Zhang and Charles Cox. Measuring the two-dimensional structure of a wavy water surface optically: A surface gradient detector. *Experiments in Fluids, Springer Verlag*, 17:225–237, 1994.
- [79] Zhengyou Zhang and Olivier Faugeras. Estimation of displacements from two 3d frames obtained from stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(12):1141–1156, 1992.
- [80] Todd E. Zickler, Jeffrey Ho, David J. Kriegman, Jean Ponce, and Peter N. Belhumeur. Binocular helmholtz stereopsis. In *Proceedings. Ninth IEEE International Conference on Computer Vision*, pages 1411–1417, 2003.
- [81] Douglas E. Zongker, Dawn M. Werner, Brian Curless, and David H. Salesin. Environment matting and compositing. In Alyn Rockwood, editor, *Siggraph 1999, Computer Graphics Proceedings*, pages 205–214, Los Angeles, 1999. Addison Wesley Longman.