Spatio-temporal image-based water surface reconstruction: Application to sloshing in navigation lock filling

Lukas Engelen

Supervisor: Prof. dr. ir. Tom De Mulder
Counsellors: Ir. Stéphan Creëlle, Ir. Laurent Schindfessel

Master's dissertation submitted in order to obtain the academic degree of
Master of Science in Civil Engineering

Department of Civil Engineering
Chair: Prof. dr. ir. Peter Troch
Faculty of Engineering and Architecture
Academic year 2015-2016

UNIVERSITEIT
GENT

Foreword

Firstly, I wanted my master dissertation to be experimental-based because it allows one to study the world of physics in a practical way. Moreover, my interest in hydraulic & hydrodynamic engineering grew during the first years of my study in Civil Engineering. The choice for this master dissertation, in which experimental fundamental research is combined with the field of hydrodynamics, was as such evident.

I enjoyed working on this thesis because it allowed me to come up with out of the box ideas, in which I explored a whole new world of experimental, image-based techniques. Although the numerous difficulties I encountered made it sometimes frustrating, the problem solving nature of this thesis made it also very challenging and satisfying. I developed a novel methodology that can be be used to study various hydrodynamic phenomena. For this reason, this work offers a clear contribution to further research.

I would like to thank all the people who supported me during the past year. First of all, my gratitude goes to prof. dr. ir. Tom De Mulder of the Hydraulics Laboratory of Ghent University, who gave me the opportunity and freedom to follow my own path during this thesis. Secondly, I would like to thank ir. Stéphan Creëlle and ir. Laurent Schindfessel for their close involvement, guidance and feedback during this past year. Their expertise and experience with image-based techniques helped a lot in the practical development of the algorithm and their support made my work definitely more pleasing. Also the technicians at the laboratory, Stefaan Bliki and Jens De Crop, have contributed significantly in realizing the experimental setup that was needed. For that, I am very grateful.

Finally, I wish to thank my family for their supporting words, not only during this master thesis but also during my entire study of Civil Engineering.

Lukas Engelen
Ghent, June 2016

Spatio-temporal image-based water surface reconstruction: Application to sloshing in navigation lock filling

by Lukas Engelen
Supervisor: Prof. dr. ir. Tom De Mulder
Counsellors: ir. Stéphan Creëlle, ir. Laurent Schindfessel

Master's dissertation submitted in order to obtain the academic degree of
Master of Science in Civil Engineering

Department of Civil Engineering
Chair: Prof. dr. ir. Peter Troch
Faculty of Engineering and Architecture
Academic year 2015-2016

Summary

During this master dissertation, we present a novel, image-based methodology to obtain a spatio-temporal description of a dynamically changing water surface. The method combines a shape from refractive distortion approach with prior knowledge about the expected surface shape. To that end, we derive a parameterized model that for each time instance is fitted to describe the instantaneous surface shape.

The developed algorithm is validated by means of experimental tests as well as with numerical simulations. The results prove that we are able to reconstruct the three-dimensional shape of the water surface with sub-millimeter accuracy. Additionally, we conducted an extensive error assessment to obtain insight in how the accuracy and robustness of the algorithm can be maximized.

The method is finally extended with an optical flow method to track distinctive features across images, which allows highly time-efficient processing of image sequences or videos. A first test case is presented, consisting of the filling process of a rectangular basin, which shows that the time-dependent three-dimensional shape of a fluctuating water surface can be reconstructed. The temporal resolution was however too small to obtain accurate quantitative results. Provided that the experimental setup is improved, it can be assumed that the developed algorithm can be used to study numerous hydrodynamic phenomena.

Keywords

Image-based reconstruction, shape from refractive distortion, 3D shape acquisition of water, surface parameterization, feature tracking

Spatio-temporal image-based parametric water surface reconstruction: Application to sloshing in navigation lock filling

Lukas Engelen

Supervisor: prof. dr. ir. Tom De Mulder
Counsellors: ir. Stéphan Creëlle, ir. Laurent Schindfessel

Abstract

In this paper, an image-based reconstruction technique is presented in which the refractive distortion of a known pattern seen through the water is used. Based on prior knowledge about the expected surface shape, a low parameter surface model is proposed which is fitted to the refractive disparities (Morris (2004)) of a feature pattern located below the water. The developed algorithm is validated in experimental tests as well as with numerical simulations. Additionally, an extensive error assessment with respect to the most important aspects of the reconstruction procedure is elaborated based on single images of still water. The optimal reconstruction conditions are determined, which allow robust and highly accurate 3D reconstruction of the surface shape. In order to reconstruct sequences of dynamically changing surface shapes, the developed method is extended with optical flow methods that allow to track the individual feature points across multiple images. The global temporal reconstruction algorithm is finally tested in a first experiment in which the surface oscillations during filling of a rectangular basin are reconstructed.

Keywords: Image-based reconstruction, shape from refractive distortion, 3D shape acquisition of water, surface parameterization, feature tracking

1 Introduction

Reconstruction of a three-dimensional dynamically changing water surface is difficult using traditional measurement equipment. To obtain an accurate description of the 3D shape of the water surface, image-based are therefore preferred. Moreover, they allow highly accurate results with easily available imaging equipment. Optical spatio-temporal reconstruction of water surfaces is however a difficult task due to the optical properties of water and its dynamic behavior. From the available methods in literature, direct rays measurements and shape from refractive distortion seem the most suitable methodologies to extract shape information in which the refractive properties of water are exploited. Techniques that only require a single image to reconstruct the instantaneous surface shape are preferred due to the rapid evolution of the dynamic water surface. This proves however difficult due to the ill-posed nature of the refractive-based geometric problem. Even if the refractive index of the reconstructed object is known, image assumptions are required to solve this inherent ambiguity.

Ben-Ezra and Nayar (2003) adopt a general parametric form (model) of a static transparent object with known refractive index to estimate its position and shape. They assume that the background pattern is located far from the refractive transparent object, which allows to compute the best fitting shape parameter ξ, rotation matrix R and translation vector T.

Murase (1992) was the first to apply shape from distortion to reconstruct water surfaces. Due to the imaging assumptions made and the limited applicability of this method, other alternatives were suggested which also use the refractive distortion of a pattern seen through the water. Kutulakos and Steger (2008) give an extensive overview of possible solutions for the ambiguity that arises in case direct ray measurements are combined with refraction. Most of these solutions seem however inappropriate for the temporal reconstruction of an oscillating water mass.

Morris (2004) proposed a multi-view stereo approach which combines refractive reconstruction approaches with the traditional stereo techniques that are usually used to extract 3D information from a scene. This allows to avoid the imaging assumptions and the related inaccuracies made in other methods. By adopting a 'verification camera', they are able to solve the depth-normal ambiguity that arises due to the refraction of light at the air-water interface. The method has already successfully been applied to reconstruct water surface sequences with mm accuracy (Morris and Kutulakos (2005)).

The reconstruction algorithm presented in this paper is originally developed as a research tool to study the self-induced sloshing phenomenon on a small-scale model of a navigation lock. Self-induced sloshing occurs when the free surface of a water mass within a container oscillates periodically due to an imposed flow without the presence of other external forces. The surface shape remains in that case smooth -at least when no ship is present in the lock chamber- and can be described by a limited amount of parameters. For this reason, this work intends to combine the shape from refractive distortion approach of Morris (2004) with a low parameter model based on prior knowledge about the expected surface shape. This finally needs to be incorporated in a global, image-based reconstruction algorithm to obtain a temporal description of the surface shape.

2 Methodology

The presented methodology is based on refraction of light rays at the air-water interface, where the water is assumed to have a known and constant refractive index for each individual light ray. A regular feature pattern (**f**) on a plain surface F is positioned below the water and the projected, deformed pattern on the surface is viewed by the camera. At the intersection of these viewing rays with the water surface, further denoted as surface points **p**, the light rays change in direction (Figure 1). This causes that the viewing rays from the camera center **c** intersect the feature plane F at **f** instead of points **f'** (in case no refraction would occur). The distance between **f** and **f'** determines the 'refractive disparity' of the feature point, as defined by Morris (2004).

The refraction of light rays is governed by Snell's law. It expresses the relationship between the incident and refracted angle w.r.t. the surface normal at the boundary between two refractive media with a different refraction index r. At the air-water interface, Snell's law can be formulated as:

$$sin(\theta_{air}) = r_w \, sin\,(\theta_{water}) \tag{1}$$

where the refractive index of water r_w typically equals 1.33. Additionally, it is known that the surface normal, incident and refracted light ray lie in the same plane. This allows to simplify the 3D reconstruction to a two-dimensional geometric problem in function of the unknown location of surface point **p** and the direction of the surface normal $\overrightarrow{\boldsymbol{n}}$. An ambiguity however arises because of this underconstrained expression in two unknowns ($\boldsymbol{p}, \overrightarrow{\boldsymbol{n}}$).

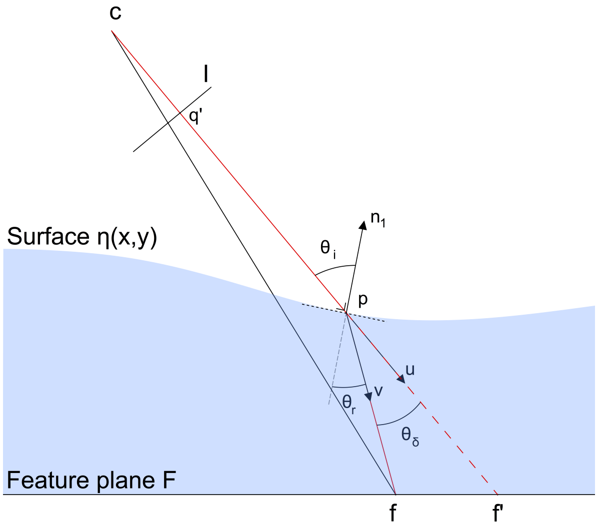


Figure 1: Illustration of derivation of surface normal $\overrightarrow{\boldsymbol{n}}_{\boldsymbol{1}}$ based on refractive disparity. Based on Morris (2004).

In case the location of the surface point **p** is assumed to be known, the direction of the incident and refracted rays at **p** can however be computed using the known 3D coordinates of feature point **f**, image point **q'** and camera center **c**. As shown in Figure 1, the incident light rays are defined as $\overrightarrow{\boldsymbol{u}} = \overrightarrow{\boldsymbol{cq'}}$ and the refracted light rays as $\overrightarrow{\boldsymbol{v}} = \overrightarrow{\boldsymbol{pf}}$. It can be proven that the surface normal $\overrightarrow{\boldsymbol{n_1}}$ which accounts for the refractive change in direction between $\overrightarrow{\boldsymbol{u}}$ and $\overrightarrow{\boldsymbol{v}}$ is given by:

$$\overrightarrow{\boldsymbol{n_1}} = R(\theta_i, \hat{\boldsymbol{u}} \times \hat{\boldsymbol{v}})(-\hat{\boldsymbol{u}}) \tag{2}$$

where R(θ, $\hat{\boldsymbol{X}}$) represents the rotation matrix of an angle θ about an (normalized) axis $\hat{\boldsymbol{X}}$. Based on the 2D refraction of the light rays, the rotation axis in eq. (2) is expressed as the vector perpendicular to the plane defined by the normalized incident $\hat{\boldsymbol{u}}$ and refracted $\hat{\boldsymbol{v}}$ light rays. The rotation angle is finally computed by applying Snell's law and basic trigonometric identities:

$$\theta_i = tan^{-1}\left(\frac{r_w sin(\theta_\delta)}{r_w cos(\theta_\delta) - 1}\right) \tag{3}$$

where θ_δ represents the angle between the incident and refracted ray.

As already mentioned, the novel approach described in this paper combines the observed disparities with a parameterized surface model describing the surface shape. The theory of Lamb (1932) is used to derive a general description of the surface undulations for an oscillating water mass in a rectangular tank with dimensions L_x and L_y:

$$\eta(x,y) = \sum_m \sum_n A_{mn}\, cos(\frac{m\pi x}{L_x})\, cos(\frac{n\pi y}{L_y}) \quad \text{with } m,n = 0,1,2,\ldots \tag{4}$$

This infinite sum is limited to second order terms and additionally extended with a linear term in the x- and y-direction $(B\frac{x}{L_x} + C\frac{y}{L_y})$ to cope with unevenness of the flat but not perfectly horizontal plane F. This results in a final, parameterized surface model $\eta(x,y)$ containing 8 coefficients (parameters), which describes the surface height z at every location (x,y). In case a certain set of hypothesized coefficients A_{mn}, B and C is assumed, the surface points **p** can be computed as the intersection points of the incident rays $\overrightarrow{\boldsymbol{u}} = \overrightarrow{\boldsymbol{cq'}}$ with the water surface corresponding to these coefficients. Additionally, this model can be used to obtain an alternative set of surface normals $\overrightarrow{\boldsymbol{n_2}}$:

$$\overrightarrow{\boldsymbol{n_2}} = (-\frac{\partial\eta(x,y)}{\partial x}, -\frac{\partial\eta(x,y)}{\partial y}, 1) \tag{5}$$

$$\text{in which } n_{2,z} = \frac{\partial\eta(x,y)}{\partial z} = 1$$

After normalization of $\overrightarrow{\boldsymbol{n_2}}$, this normal set can be compared with the set of normals $\overrightarrow{\boldsymbol{n_1}}$ computed with eq. (2) to obtain the optimal set of coefficients in eq. 4. To this end, two possible error metrics are proposed that express the dissimilarity between the two surface normals corresponding with the same surface point. The first error metric (M1) is denoted as the 'normal collinearity metric' E_{col}, which directly compares the direction of $\overrightarrow{\boldsymbol{n_1}}$ and $\overrightarrow{\boldsymbol{n_2}}$:

$$E_{col} = cos^{-1}(\overrightarrow{\boldsymbol{n_1}} \cdot \overrightarrow{\boldsymbol{n_2}}) \tag{6}$$

A second suggestion (denoted M2) by Morris (2004) is to use the refractive displacements of the feature points in case the two normal sets are swapped. In case $\overrightarrow{\boldsymbol{n_1}}$ and $\overrightarrow{\boldsymbol{n_2}}$ are not the same, this causes a refractive displacement of the intersection of the refracted ray with plane F. The distance between the intersection point $\boldsymbol{f_2}$ and the corresponding actual feature location **f** allows to compute the 'disparity difference metric' E_{disp}, which is also shown in Figure 2.

$$E_{disp} = |\boldsymbol{f} - \boldsymbol{f_2}| \tag{7}$$

One of both metrics is computed for every feature point by adopting a hypothesized set of coefficients in eq. (4). These errors are finally combined to obtain a global error function that needs to be minimized:

$$E_{tot} = \sum_f E_f^2 \tag{8}$$

The entire surface reconstruction for one particular time instance is as such transposed to a multivariate optimization of the coefficients in eq. (4) which minimize the global error function. This optimization is done by applying the Levenberg-Marquardt optimization algorithm, implemented in the open-source library ALGLIB (ALGLIB (2016)).

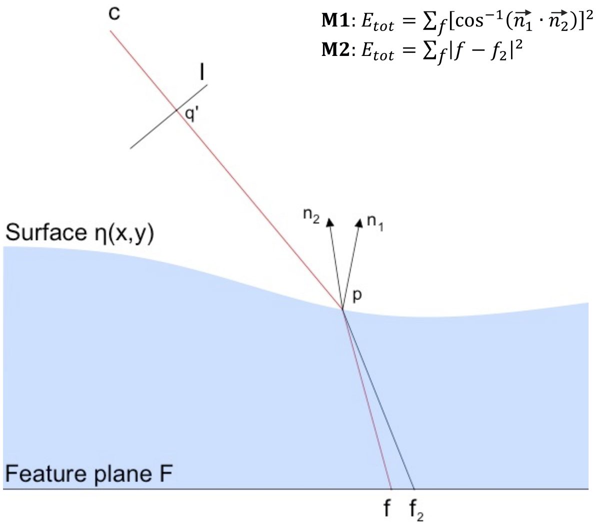


Figure 2: Illustration of the two possible error metrics that measure the dissimilarity between the refractive-based surface and the parameterized surface model. Based on Morris (2004).

In order to reconstruct the water surface at any time instance, the pixel locations **q'** in the corresponding image need to be known. The points in the first processed image are located with an OpenCV feature detector (Bradski (2000)), after which the image points **q'** are tracked across the image sequence using the pyramidal LK optical flow method (Bouguet (2001)). Although the developed reconstruction algorithm considers each time frame separately, tracking of feature points allows much faster processing than in case no prior knowledge about their location is used. Each set of image points **q'** allows finally to derive the most suitable set of coefficients that describe the instantaneous surface shape.

3 Validation and error assessment

3.1 Experimental setup

Measurements of actual water surfaces were performed in a small-scale test tank, with a cross-section of 8 cm wide and 10 cm high. Three Basler ace GigE cameras were positioned above the test tank. Additionally, two possible lighting setups were installed: regular light spots for images with visible light but also a UV light spot in combination with fluorescent paper was employed. In the latter case, the illuminated fluorescent feature plane served as apparent visible light source for the camera. The feature pattern on the bottom of the tank consisted of dots with a diameter of 0.1 mm with a spacing of 1 cm.

3.2 Experimental validation based on still water

Because the presented method is image-based, a validation with a still and horizontal water surface is only a small simplification compared to a dynamic water surface. It allows on the other hand to ground-truth the obtained water level differences with measurements using a level gauge. For this purpose, the error made on the water level differences over the entire reconstructed area is computed and subsequently spatially averaged to obtain an error measure $|E_{mean}|$. In case strong and uniform illumination of the feature plane is realized, the experimental tests show that both UV and visible light allow to obtain results within the accuracy range of the level gauge (0.2 mm). This is demonstrated in Figure 3 for UV light at different water level differences, with initial and final (average) water level indicated as h_i and h_f respectively.

Secondly, a study on the influence of several important parameters in the reconstruction was conducted (results are not shown). The variation in the refractive index of light has a negligible effect on the surface reconstruction, as well as the distortion model used and the estimation of the camera position. It must be mentioned that these results are only valid in the range of combinations of camera pose and surface location that were tested.

An important parameter in the surface reconstruction is however the inclination of the viewing rays w.r.t the surface. More specifically, grazing angles allow more accurate results because the refractive disparities in the image plane then become larger for the same movement of the surface points $\mathbf{p}$. Initial knowledge about the expected surface that is reconstructed proves also important: redundant terms in the surface model make the algorithm more susceptible to inaccurate feature localization. When combined with the disparity difference metric, the optimization seems in that case more sensitive. Finally, combining multiple cameras proves highly effective to improve the accuracy and the robustness with respect to inaccurate input for the optimization.

3.3 Numerical validation

Inaccuracy in feature localization was modeled by adopting a Gaussian $N(0, \sigma)$ distribution for the localization errors made within the image plane. Random noise was added to the pixel coordinates of the correct image points corresponding with a known surface location and the deviation of the resulting sets of optimized coefficients was evaluated.

Firstly, the numerical simulations showed that very small water depths make it more difficult to obtain accurate results. This is caused by the smaller movements of the surface points corresponding to a varying surface shape. Additionally, movement of points located far from the camera (i.e. at low water depths) results in smaller movements in the image plane. For larger water depths, the accuracy increases in case the amount of adopted coefficients remains limited. For more parameter-redundant models, the error of the reconstructed surfaces however increases again with increasing water depth after reaching a minimum which is dependent on the test configuration. The normal collinearity metric (M1) seems in that case more robust in case the localization error remains small. In contrast, the disparity difference metric (M2) proves more robust to large localization errors. It remains however difficult to explain this on a theoretical basis. It can be concluded that prior knowledge about the expected surface shape can significantly increase the performance of the algorithm.

4 Reconstruction of a dynamically changing water surface

The developed reconstruction method was finally tested in a first experiment in which the filling process of a navigation lock was simulated. Figure 4 shows that the developed method allows to obtain a good approximate description of the variability of the water surface in the tank. The experimental setup proved however insufficient due to the limited frame rate compared to the celerity of the reconstructed surface wave phenomena. This made high temporal resolution images with sufficient quality challenging, making it difficult to relate the test results with the theory of self-induced sloshing. The results matched however qualitatively with theoretical expectations for shallow water waves.

5 Conclusions

In this paper, a novel methodology is presented to describe the spatio-temporal variability of the surface of an oscillating water mass. The developed algorithm combines the refractive distortion of a known feature pattern with a low parameter surface model that is fitted in a global optimization algorithm. The proposed technique was validated

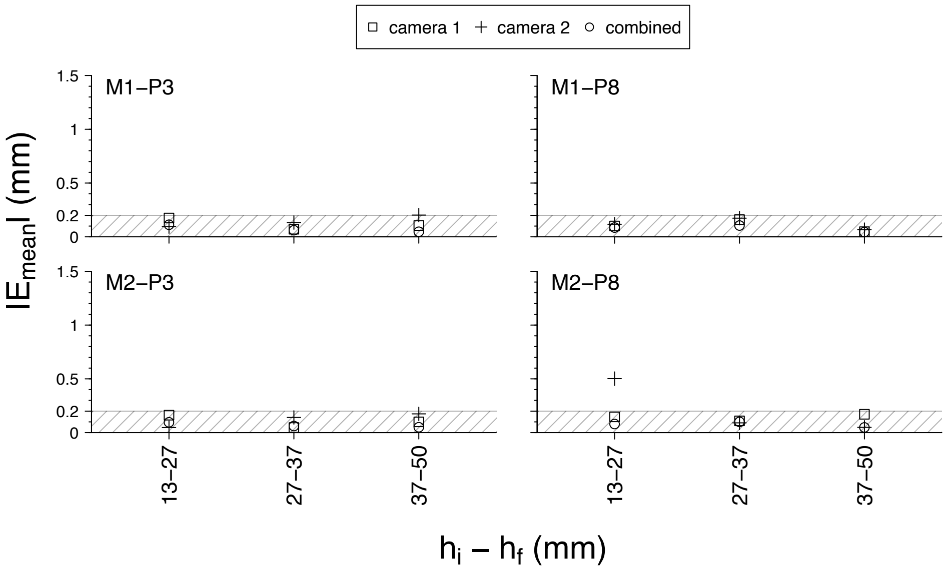


Figure 3: Spatially averaged error $|E_{mean}|$ for the test case with UV light at different water levels. For each reconstruction, metrics M1 (normal collinearity metric) or metric M2 (disparity difference metric) and a surface model with 3 (P3) or 8 (P8) parameters was used.

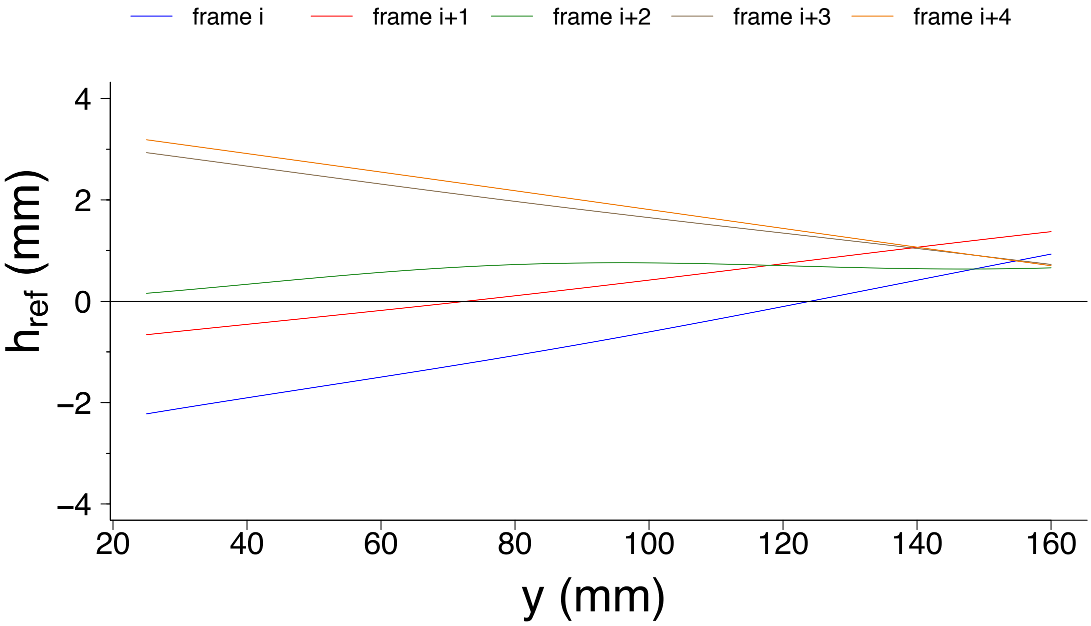


Figure 4: Reconstructed water surfaces using the normal collinearity metric (M1), averaged over the width of the tank and relatively to the final still water level. The frame rate (and the corresponding temporal resolution of the reconstruction) was 6.8 fps or ±0.15 s between successive frames.

based on flat water surfaces which showed that accurate results can be obtained. Several improvements can however be made to ameliorate its performance in case of a dynamically changing water surface. Because the developed C++ algorithm is modular based, adaptations to the algorithm should be easy to implement.

Bibliography

ALGLIB. Project: ALGLIB: a cross-platform open source numerical analysis and data processing library. `http://www.alglib.net/`, 2016.

M. Ben-Ezra and S. K. Nayar. What does motion reveal about transparency? In *Proc. IEEE 9th Int. Conf. Computer Vision (ICCV), 2003*, pages 1025–1032, 2003.

J.-Y. Bouguet. Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. pages 1–10, 2001.

G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. [Currently available at: `http://opencv.org/`].

K. N. Kutulakos and E. Steger. A theory of refractive and specular 3d shape by light-path triangulation. *Int. Journal of Computer Vision (IJCV)*, 76(1):13–29, 2008.

H. Lamb. *Hydrodynamics*. Cambridge University Press, 1932.

N. J. W. Morris. *Image-based water surface reconstruction with refractive stereo*. PhD thesis, University of Toronto, 2004.

N. J. W. Morris and K. N. Kutulakos. Dynamic refraction stereo. In *Proc. 10th IEEE Int. Conf. Computer Vision (ICCV), 2005*, volume 2, pages 1573–1580, 2005.

H. Murase. Surface shape reconstruction of a nonrigid transport object using refraction and motion. *IEEE Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, 14(10):1045–1052, 1992.

Contents

List of symbols

Symbol	Unit	Explanation
		New symbols in Chapter 2
W	$[m]$	tank width
D	$[m]$	tank depth from the front
B	$[m]$	inlet height from the bottom
S	$[m]$	outlet location from the left side wall
b	$[m]$	inlet width
s	$[m]$	outlet width
U_0	$[\frac{m}{s}]$	jet inlet velocity
H	$[m]$	mean water level
h	$[m]$	surface height from inlet center
n	$[-]$	mode of sloshing
m	$[-]$	jet mode (stage)
f_s^n	$[\frac{1}{s}]$	sloshing frequency of water in a rectangular tank
ω_s	$[\frac{rad}{s}]$	angular sloshing frequency
ϕ_s	$[\frac{m^2}{s}]$	sloshing potential
a	$[m]$	average value of surface fluctuations
δ	$[rad]$	phase delay
ρ	$[\frac{kg}{m^3}]$	density
p	$[Pa]$	pressure
ν	$[\frac{m^2}{s}]$	kinematic viscosity
g	$[\frac{m}{s^2}]$	acceleration of gravity
ΔE_n	$[\frac{Nm}{s}]$	sloshing energy in a control volume per unit of time
T_s	$[s]$	sloshing period
$\overline{\Delta E_n}$	$[J]$	sloshing energy in a control volume over T_s
E_{con}	$[\frac{J}{m^2 m^2}]$	convection contribution to oscillation energy
E_{press}	$[\frac{J}{m^2 m^2}]$	pressure contribution to oscillation energy
E_{diss}	$[\frac{J}{m^2 m^2}]$	dissipation contribution to oscillation energy
E_{unst}	$[\frac{J}{m^2 m^2}]$	unsteady oscillation energy
St	$[-]$	Strouhal number
f_r	$[\frac{1}{s}]$	representative frequency in expression of St
l_r	$[m]$	representative length in expression of St
U_r	$[\frac{m}{s}]$	representative velocity in expression of St
u_{con}	$[\frac{m}{s}]$	dominant phase velocity of jet fluctuation
St_s	$[-]$	modified Strouhal number
Re	$[-]$	Reynolds number
μ	$[\frac{kg}{m\,s}]$	dynamic viscosity

Symbol	Unit	Explanation
		New symbols in Chapter 3
P		plenoptic function
z	$[m]$	stereo depth from point to stereo baseline
d	[pixels]	stereo disparity
f	[pixels]	focal length
		New symbols in Chapter 4
r	$[-]$	refractive index
r_w	$[-]$	refractive index of water
θ_i	$[rad]$	angle between incident light ray and surface normal
θ_i	$[rad]$	angle between refracted light ray and surface normal
θ_{cr}	$[rad]$	critical angle
$\mathbf{p}$		surface point = (p_x,p_y,p_z)
$\mathbf{c}$		camera center = (c_x,c_y,c_z)
$\mathbf{f}$		feature point on feature plane F = (f_x,f_y,f_z)
f’		apparent feature location without refraction = (f'_x,f'_y,f'_z)
$\mathbf{q}$		image point without water present = (q_x,q_y)
q’		image point with water present = (q'_x,q'_y)
$\overrightarrow{\boldsymbol{n}}$		surface normal vector = (n_x,n_y,n_z)
h	$[m]$	ordinate of undisturbed free surface
ζ	$[m]$	surface elevation above undisturbed free surface
p	$[Pa]$	pressure term in Bernouilli equation
p_0	$[Pa]$	atmospheric pressure
c_{wave}	$[\frac{m}{s}]$	wave celerity
η	$[m]$	surface function describing water surface shape
A_{00},A_{10},A_{01},A_{11} A_{20},A_{02},B,C	$[m]$	surface coefficients in surface model η
θ_δ	$[rad]$	angle between incident light ray and refracted ray
$\overrightarrow{\boldsymbol{u}}$		direction of incident light ray = (u_x,u_y,u_z)
$\overrightarrow{\boldsymbol{v}}$		direction of refracted ray = (v_x,v_y,v_z)
F	$[m]$	implicit form of surface function
$\overrightarrow{\boldsymbol{n_1}}$		surface normal vector derived with refractive disparity = (n_{1x},n_{1y},n_{1z})
$\overrightarrow{\boldsymbol{n_2}}$		surface normal vector derived with surface model = (n_{2x},n_{2y},n_{2z})
E_{col}	$[rad]$	normal collinearity metric
$\boldsymbol{f_2}$		intersection point of ray with plane F due to $\overrightarrow{\boldsymbol{n_2}}$ = (f_{2x},f_{2y},f_{2z})
E_{disp}	$[m]$	disparity difference metric
E_{tot}	$[m]$	global error function
		New symbols in Chapter 5
I	$[-]$	(graylevel) intensity
w		window function
I_x		image derivative in x-direction
I_y		image derivative in y-direction
M		structure tensor of image $\in \mathbb{R}^{2\times 2}$
λ	$[-]$	eigenvalue of structure tensor

Symbol	Unit	Explanation
∇		gradient operator
		New symbols in Chapter 6
(x_d,y_d)		distorted image point position
(x_u,y_u)		undistorted image point position
(c_x,c_y)		position of principal point
k_n	$[-]$	n^{th} radial distortion coefficient
p_n	$[-]$	n^{th} tangential distortion coefficient
s_n	$[-]$	n^{th} thin prism distortion coefficient
r	[pixels]	distance to principal point
τ_x, τ_y	$[-]$	angular parameters of Scheimpflug model
s_x	$[-]$	scale factor in horizontal direction
s_y	$[-]$	scale factor in vertical direction
s_{xy}	$[-]$	skew coefficient
f_x	[pixels]	focal length for scaling in horizontal direction
f_y	[pixels]	focal length for scaling in vertical direction
K		Intrinsic camera matrix $\in \mathbb{R}^{3\times3}$
T		Translation vector $\in \mathbb{R}^{3\times1}$
R		Rotation matrix $\in \mathbb{R}^{3\times3}$
P		Projection matrix $\in \mathbb{R}^{3\times4}$
		New symbols in Chapter 7
$M1$		normal collinearity metric
$M2$		disparity difference metric
σ		standard deviation
$N(0,\sigma)$		Gaussian or normal distribution with mean 0
δq		simulated localization error with $\delta\boldsymbol{q} \sim N(0,\sigma)$ $= (\delta q_x,\delta q_y)$
W		spatial domain in which reconstruction is validated
Δh	$[m]$	reconstructed water level change at (x,y) within W
Δh_{gauge}	$[m]$	measured water level change with level gauge
E_q	$[m]$	error of reconstructed surface at (x,y) within W
N	$[-]$	number of discrete surface points in domain W
E_{mean}	$[m]$	spatially averaged error within W
Δh_{mean}	$[m]$	error measure for initialization-comparison
h_i	$[m]$	initial water height for water level difference computation
h_f	$[m]$	final water height for water level difference computation
Δq	[pixels]	image disparity shift due to varying surface
Δp	$[m]$	change in position of $\mathbf{p}$ due to varying surface
		New symbols in Chapter 8
I_t		time derivative of intensity I
$\boldsymbol{d}$	$[\frac{\text{pixels}}{frame}]$	optical flow velocity between two successive frames
ω_x	[pixels]	search interval for window in x-direction
ω_y	[pixels]	search interval for window in y-direction
G		gradient matrix $\in \mathbb{R}^{2\times2}$
I		previous frame
J		current frame

Symbol	Unit	Explanation
q_1		image point of feature in image I = (q_{1x}, q_{1y})
q_2		image point of feature in image J = (q_{2x}, q_{2y})
ϵ		matching error or residual function
I^L		pyramid level L of image I
n_x^L	[pixels]	size of image level L in horizontal direction
n_y^L	[pixels]	size of image level L in vertical direction
$\bar{b}$		image mismatch vector $\in \mathbb{R}^{2\times1}$
g^L		current estimate of optical flow at level L = (g_x^L, g_y^L)
ω_i		solution for optical flow of image point i ($\in$ 4 elements)
α_i	$[rad]$	motion direction of image point i
v_i	$[\frac{\text{pixels}}{frame}]$	velocity of image point i
L_1	[pixels]	L_1-distance between patches
L	$[m]$	length of the lock chamber
T	$[s]$	period of the translatory wave
$t_{pattern}$	$[s]$	time needed for the translatory wave to travel over pattern

List of abbreviations

Abbreviation	Full explanation
PIV	particle image velocimetry
PTV	particle tracking velocimetry
FFT	Fast Fourier transformation
1D	one-dimensional
2D	two-dimensional
3D	three-dimensional
CCD	charge-coupled device
RGB	red, green and blue (color channels)
BRDF	bidirectional reflectance distribution function
LED	light emitting diode
LCD	liquid-crystal display
Roxel	Responsibility Weighted 3D Volume Reconstruction
UV	ultraviolet
IR	infrared
B.C.	boundary condition
PDE	partial differential equation
ODE	ordinary differential equation
SSD	sum of squared differences
USAN	Univalue Segment Assimilating Nucleus
CRF	corner response function
NMS	non-maximum suppression
AST	accelerated segment test
GigE	Gigabit Ethernet
PoE	Power over Ethernet
SNR	signal-to-noise ratio
PnP	Perspective-n-Point
LMA	Levenberg-Marquardt algorithm
ISO	International Organization for Standardisation
LK	Lucas-Kanade
NaN	Not a Number

Chapter 1

Introduction

1.1 Description of the research topic

The motivation behind this master thesis is experimental research about navigation lock filling to derive physical models that describe the corresponding fluid motions. This requires accurate and three-dimensional reconstruction of the oscillating water surface. Although sloshing in rectangular tanks is usually considered as a two-dimensional phenomenon, spatio-temporal reconstruction of the surface shape could give new insights in the physical mechanism of the observed flow patterns.

The reconstruction of fluctuating water surfaces has been studied extensively in previous research. The presented methodology is image-based, in which images of the water surface are used to extract 3D shape information. Several approaches have been developed in the field of computer vision in which one of the optical properties of water is used to reconstruct the water surface shape.

Although multiple techniques have focused on the reflection of light at the water surface, their results proved disappointing. Another approach is to exploit the refraction at the air-water interface. Most refractive-based approaches however make imaging assumptions to solve the ill-posed nature of light refraction at the boundary between two different media.

In this thesis, assumptions such as a distant, orthographic camera view are avoided which increases the accuracy of the developed reconstruction algorithm. For this purpose, prior knowledge about the expected surface shape is used to derive a theoretical surface function. This parameterized model can then be fitted to the information contained within the images of the dynamically changing water surface. The presented methodology, which combines a shape from refractive distortion approach with a parameterization of the surface shape, allows to extract high accurate 3D information in which the temporal resolution is only dependent on the frame rate of the image sequence.

1.2 Motivation for the developed technique

Navigation locks are used to connect two adjacent water masses with a different water level. In practice, three particular situations require the use of navigation locks. Firstly, they allow ships to overcome a water level difference between two reaches of water, which can be canalized rivers or man-made canals. Secondly, locks are used to separate salt and fresh water. Fresh water is precious and contamination by saline water intrusion should therefore be minimized. Navigation locks in that case form a connection between areas of salt water and inland navigation channels with fresh water. For this application, ingenious systems have been developed to minimize the salt water intrusion during locking of vessels. Finally, locks can be used to create a constant water level inside a sea port and in this way provide docks that are not influenced by the tidal regime. This makes loading and unloading operations significantly easier. Consequently, navigation locks are important elements of navigable waterways and tidal ports. As shipping has become an important industry in today's global economy, considerable benefit gain be gained by improving the performance of navigation locks.

The navigation locks are constructed as a short canal, the lock chamber, which is closed off by an upper and lower lock gate. Within this lock chamber, the water height can be adjusted by using a filling and emptying system. Water is taken from the upper reach (filling) or discharged in the lower reach (emptying) in order to respectively elevate or lower the ship in the lock chamber.

Navigation locks in areas with limited differences in topographic elevations only need to overcome a small water level difference. For low head locks, leveling through openings in the gates, sealed by valves, can be relied upon. The lock chamber can subsequently be filled or emptied by lifting the valves in respectively the upper and lower lock gate as shown in Figure 1.1.

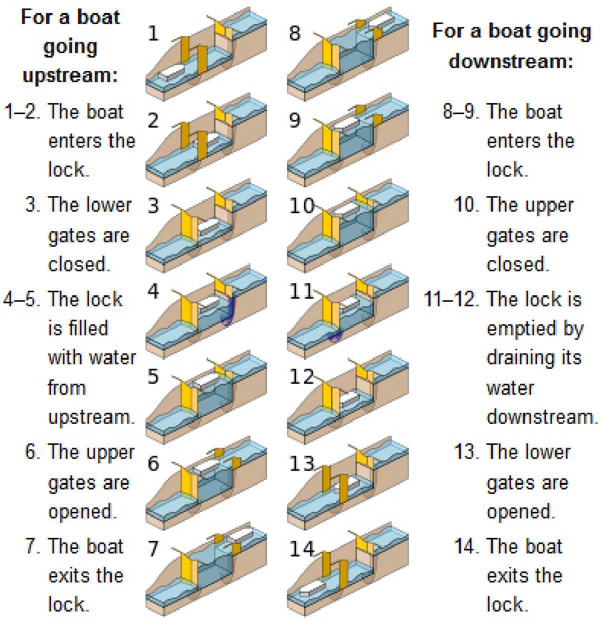


Figure 1.1: Functioning of a navigation lock. From Wikipedia (2016).

The adjustment of the fluid level in the lock gives rise to several hydrodynamic phenomena, which result in large hydrodynamic forces on the ship within the lock chamber (Figure 1.2). These forces can be grouped in three main categories as follows:

1. Forces due to the strong concentrated horizontal jets that enter the lock chamber during the filling process. These submerged jets cause both a direct stagnation force on the ship as well as a fluctuation force due to the produced turbulence.
2. Forces created by the slope of the water surface due to the lowering of the water surface above the horizontal jets. This local drop in water-level is caused by the large momentum generated by the jet and results in a longitudinal force on the ship by gravity.
3. Forces caused by a translatory wave on the water surface during filling and emptying. The waves have an oscillatory character which can result in self-induced sloshing. In the latter

case, the free-surface oscillations are the result of the submerged water jet (no external forces) and can become very high due to the effect of the submerged jet.

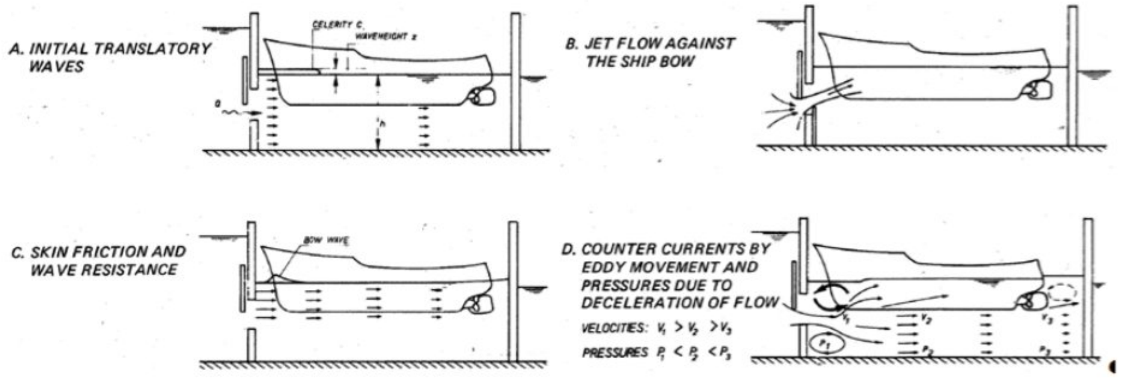


Figure 1.2: Hydrodynamic forces on a ship during filling or emptying of a navigation lock. From Kolkman (1973).

This thesis will concentrate on the last phenomenon, namely the self-induced sloshing caused by the submerged filling jets. A number of models has been developed to describe this phenomenon. Although in reality a ship will be moored at the side of the lock chamber, they incorporate the presence of the ship by modeling it as a rectangular object, placed centrally inside the lock chamber. It has been observed that the existing models are insufficient to accurately describe the sloshing motion. Secondly, predictions that have been made based on these models do not always agree well with lab or field experiments. Therefore, more fundamental research about the mechanism behind self-induced sloshing during the filling of a rectangular tank is needed.

This knowledge can then be used to obtain a more accurate and reliable parameterization of the forces on the ship. As such, simple numerical models can be developed, applicable during the (conceptual) design of navigation locks and through-the-gate filling and emptying systems.

1.3 Objectives of this master thesis

The main goal of this master thesis is to develop a methodology to determine the spatio-temporal variability of a dynamically changing water surface. The obtained reconstruction algorithm serves as research tool to study the water fluctuations in small scale models for various hydrodynamic phenomena. The motivation and first application area of the developed method is the study of self-induced sloshing in a scale model of a navigation lock. This experimental study can then be used to obtain physical insight in the occurrence of self-induced sloshing.

A list of requirements for the reconstruction methodology is given below, which are needed to obtain a technique that is suitable in practical research applications.

1. The technique needs to be sufficiently accurate to capture the small water surface oscillations in the scale model of the navigation lock. In case of self-induced sloshing, typical observed surface slopes inside navigation locks during the filling and emptying-process are in the order of 1 ‰. Given the dimensions of the experimental setup (discussed in Chapter 6, Section 6.2), this requires a measurement accuracy in the order of 0.1 mm in order to obtain an accurate reconstruction over the entire surface area.
2. The developed method is preferably modular in order to extend and improve it in later stages of the development of the methodology. This allows a rapid implementation, possible

in the time frame of this thesis, whereafter stepwise improvements can easily be implemented to address critical parts in the algorithm. Additionally, the reconstruction method can be fitted for the specific test case under consideration. By fine-tuning the technique to the scale of the water surface changes that is expected, a better overall performance of the algorithm can be obtained.

3. The technique should be compatible with velocity measurements of the fluid motion. For this purpose, techniques such as Particle Image Velocimetry (PIV) or Particle Tracking Velocimetry (PTV) are often adopted. These consist of tracking seeding particles in the water, after which their motion is considered as a good estimate of the water velocity. The developed technique to reconstruct the 3D water surface shape should therefore not come into conflict with these measurements.

4. Preferably, the developed method should be able to reconstruct a three-dimensional water surface. Although in case of self-induced sloshing the water motion is assumed to be mainly two-dimensional, an accurate spatio-temporal reconstruction of the surfaces allows to quantify possible three-dimensional effects. These can be used to obtain further insight in the flow-patterns that occur. Additionally, it offers the possibility to apply the technique to cases where 3D effects are important.

5. The technique should be scalable to other experimental setups. The developed methodology can then be applied to larger scale models of navigation locks or other fluid-dynamic phenomena.

6. Since this thesis is a first step in the development of the methodology, the adopted technique should be low-cost but offer the possibility to be improved in later stages of the research. To that end, image-based techniques have the advantage that higher accuracy can later be obtained by investing in better imaging equipment. In that regard, this thesis aims to prove that the chosen method allows to obtain accurate and reliable results.

The developed methodology should then be tested in a small scale model to confirm that it is possible to study hydrodynamic phenomena on a reduced scale. The case of self-induced sloshing is for this purpose used as a first application of the reconstruction algorithm.

1.4 Methodology

As mentioned earlier, the main goal of this master thesis is to develop a reconstruction technique to derive a spatio-temporal description of the water surface in small-scale tests. To that end, different steps are taken which are briefly outlined below.

1. Literature survey about the existing methods to determine a three-dimensional water surface. The different requirements that are mentioned in Section 1.3 are then used to choose the most promising method for further implementation during this thesis.

2. Developing a theoretical framework that allows to reconstruct a three-dimensional and fluctuating water surface.

3. Developing of a modular processing algorithm, in which the theoretical framework that was developed in the previous step is implemented in practice. Important aspects are the ability to improve the algorithm in a later stage of the development and reducing the computational time to an acceptable level. Additionally, the algorithm should be user-friendly so it can be applied widely.

4. Physical construction of the test setup. More specifically, this involves the choice of cameras and lighting setup, as well as the area and type of feature pattern that is used.

5. Validation of the reconstructed water surfaces based on several bench-mark cases. Additionally, an error assessment w.r.t. the most important parameters in the reconstruction needs to be done. This allows to determine the ideal circumstances under which the accuracy is maximized.
6. Application of the developed technique to a dynamically changing water surface. Because the motivation for the development of the reconstruction algorithm is self-induced sloshing, this first test case involves the filling process of a scale model of a navigation lock.

1.5 Thesis outline

This thesis consists of 9 chapters. After this introduction, Chapter 2 gives an overview of the existing literature on self-induced sloshing in rectangular tanks. Based on the most recent studies, the second part of this chapters comprises a description of the currently assumed excitation mechanism of self-induced sloshing in rectangular tanks. This gives insight in the magnitude of the hydrodynamic phenomena that need to be reconstructed. Chapter 3 provides a review of the existing techniques that are developed to determine the shape of transparent, reflective and refractive objects such as water. Based on this extensive overview, an image-based reconstruction method is proposed that will be implemented in this thesis.

Chapter 4 explains the theoretical framework of the method that is chosen based on the comparative study in Chapter 3. Firstly, the refractive properties of water are elaborated as well as the influence of refraction on the appearance of a pattern (consisting of so-called feature points) located below the water surface. A novel methodology is then presented in which a low parameter surface model is combined with a shape from refractive distortion approach. This allows to reduce the surface reconstruction to a multivariate optimization in which the surface model is fitted to the distortion of the feature pattern. How the individual features are recognized in the images is presented in Chapter 5. It also gives an overview of the most commonly used corner and feature detection algorithms.

In Chapter 6, a detailed description of the practical implementation of the developed method is given. In the first part of this chapter, the experimental test setup is described and some aspects which are inherent to image-based techniques are elaborated. Next, the practical implementation of the algorithm is explained in which several modules are combined to obtain an accurate and robust reconstruction of the water surface. Chapter 7 comprises a validation of the methodology by assessing the error made on the reconstruction of still water. Additionally, numerical simulations are conducted to investigate the influence of inaccurate feature localization on the reconstruction result. Both types of tests allow to quantify the influence of the most crucial parameters in the reconstruction on the accuracy and robustness of the methodology.

In Chapter 8, the reconstruction algorithm is extended with feature tracking based on optical flow. First, a theoretical background of optical flow methods is given, after which the practical implementation is elaborated. The final part of this chapter concerns a first application to dynamically changing water surfaces, in which the spatio-temporal variability of the water surface is reconstructed during the filling process of a rectangular basin. Chapter 9 gives a general overview of what is accomplished during this thesis, as well as suggestions for further development of the reconstruction algorithm.

1.6 Contributions of this work to hydraulic research

Firstly, this thesis provides an extensive overview of possible (optical) approaches that can be used to reconstruct specular reflective, transparent and refractive objects. This review focuses on techniques that seem suitable to reconstruct a dynamically changing water surface. It serves as background to obtain more insight in the possibilities of optical techniques and as source of possible ideas for further research.

Most importantly, a novel methodology is presented in which a shape from refractive distortion approach is combined with a low parameter surface model to describe the instantaneous surface shape. Because the refractive ray measurements are combined in a single multivariate optimization procedure, the presented method becomes more robust and time-efficient compared to other techniques. The algorithm is implemented in a single C++ program, in which each aspect of the reconstruction is incorporated in a different module of the program. This makes the algorithm user-friendly and allows to adapt it depending on the phenomenon that is reconstructed to obtain the best possible performance.

Furthermore, the reconstruction algorithm is validated and an extensive error assessment is conducted based on experimental tests and numerical simulations. The tests show that the surface shape can be determined with mm accuracy for a large range of water depths. The algorithm is additionally tested in a first, small-scale experiment of rapidly changing water surface. The experimental setup proved to be the limiting factor in the performance of the reconstruction. In case these would be improved, it can be assumed that the method allows an accurate 3D reconstruction of the temporal variation of a water surface. Based on the findings in this thesis, the most crucial parameters to obtain accurate and robust results are discerned. These conclusions can be useful for other researchers in case they would implement the algorithm in their studies.

Finally, a list of suggestions is provided to enhance the algorithm's performance in further development of the reconstruction methodology. The algorithm is implemented in a modular C++ program, which makes such improvements easy to incorporate.

Chapter 2

Self-induced sloshing

As already mentioned in Chapter 1, the motivation of the developed methodology is the still lacking fundamental research on the self-induced sloshing phenomenon during lock filling. Additionally, this topic serves as a first application area for the developed reconstruction methodology.

In case a navigation lock is filled and emptied by openings in the lock gates, this causes submerged jets that can result in oscillations within the lock chamber. Knowledge about sloshing allows to obtain insight in the expected surface oscillations that will need to be reconstructed. To that end, this chapter provides a theoretical background on the excitation mechanism of fluid motion by a horizontal injected jet.

2.1 Overview of previous research on self-induced sloshing

Sloshing can be defined as the fluid motion within a partially filled container (Park et al. (2014)). The understanding and prediction of this oscillatory motion is important for many other engineering applications besides lock filling: liquid cargo in vessels or tank trucks, motion of cooling liquids in systems subjected to earthquakes,... Sloshing of a fluid within a container can in general be categorized into two types: externally-induced sloshing and self-induced sloshing.

Externally-induced sloshing implies a certain mobility of the liquid container in combination with an external force acting on the container. The motion of the container induces an oscillating motion of the fluid mass inside the container. A navigation lock is in contrast fixed and normally no external forces act on the tank. In earthquake-prone regions, the earthquake induced vibrations could be an additional factor in design. This specific aspect will however not be treated within this thesis.

Self-induced sloshing is excited by flow without any other external force. Okamoto et al. (1991) discovered that in a rectangular tank the free surface oscillates periodically due to an incoming horizontally injected jet for certain conditions of flow rate and water level. The frequency of the free surface sloshing motion equals in that case the eigenvalue of the water mass within the test tank.

A few numerical simulations have been done to obtain a better understanding of self-induced sloshing and several numerical growth models have been proposed for a test setup with a downward outflow. Amano and Iwano (1991) concluded that the surface oscillations were excited by the imbalance of vortices that are formed above and below the incoming jet near the inlet. Takizawa et al. (1992) also carried out numerical simulation to investigate self-induced sloshing.

They were able to solve the two-dimensional Navier-Stokes equations with Physical Competent Boundary Fitted Coordinate (PCBFC) (Takizawa and Kondo (1995)). It was suggested that the sloshing growth is the result of flow directly under the free surface, which is a secondary flow due to the potential variation. The oscillation energy supplied by the surface potential was assumed to be transformed into kinetic energy, resulting in a forced vibration due to the nonlinear secondary flow. This model failed however to describe the sloshing growth quantitatively.

Saeki et al. (1998) used numerical simulation to determine a growth mechanism in which the sloshing excitation energy is supplied by the nonlinear interaction between the jet fluctuation and the corresponding sloshing motion. Although this could be used to explain the occurrence of self-induced sloshing in numerical simulation, this model was difficult to validate by experimental results as it requires high-resolution velocity distributions. At that time however, the available experimental measurement techniques only allowed relatively sparse and low accurate velocity information. This proved insufficient for verification as stated by Okamoto et al. (2000). It must be mentioned that at the present time, experimental velocity measurements have significantly improved which solves this issue for further research.

Several experimental studies have been carried out to understand the sloshing behavior and to develop theoretical models in order to model the sloshing growth imposed by a horizontally injected water jet (with downward outflow). Okamoto et al. (1991) were the first to experimentally investigate the oscillatory motion of the free water surface. They observed the first sloshing mode and concluded that the oscillation energy was supplied by the surface potential variations due to the flow pattern transformations. Based on these results, a theoretical model for this first mode sloshing was developed by Okamoto et al. (1992). To derive this model, they assumed that the pressure fluctuations are the result of the circulating flow and sloshing motion.

Okamoto et al. (1996) reported two kinds of sloshing modes and suggested that the self-induced sloshing was caused by the interaction of the plane jet flow with the free surface elevation. Saeki et al. (1999) investigated both sloshing modes and discovered two sloshing conditions, first and second stage sloshing, that are necessary for sloshing to occur. They derived the modified Strouhal number St_s as governing parameter for sloshing growth, and clarified the overall physical oscillation mechanism using a closed feedback mechanism between jet fluctuation and sloshing motion. Hu et al. (1999) observed that for a certain experimental tank geometry, the water surface can oscillate in its first or second mode but also multi-mode sloshings can occur.

The results of Saga et al. (2000b), in which PIV was used to study the flow pattern and the evolution of the vortex structures in a rectangular tank, confirmed the earlier findings of Saeki et al. (1998) and Saeki et al. (1999). The continuous, instantaneous PIV measurements allowed to determine both the time- and phase-averaged flow fields. Additionally, the velocity power spectrum of the flow field was calculated using a FFT transformation. This confirmed that the characteristic frequencies in the velocity power spectrum are indeed the frequencies of self-induced sloshing of the free water surface in the test tank. Using the high resolution velocity information, the oscillating movement component was extracted. This clearly showed the vortical an turbulent structures, as well as the form of the oscillating wave. They concluded that resonance oscillation occurred due to interaction of the periodic fluctuation of the inlet jet and the periodically shedding of unsteady vortices.

Okamoto et al. (2000) used experimental results based on PIV measurements in order to obtain a better understanding on the dependency of the first mode sloshing on inlet velocity, water level and tank geometry. Additionally, second mode self-induced sloshing and the observation of two separate sloshing conditions with respect to the inlet velocity was investigated. Due to the turbulent behavior of the injected jet, fast and accurate velocity measurements seemed necessary. Although they were able to qualitatively prove the excitation mechanism and sloshing conditions for both modes of oscillation, it appeared to be difficult to obtain a high-resolution time and space data set. The limitation of the available measuring techniques at that time required a different approach to quantitatively explain the self-induced sloshing phenomenon based on high accurate velocity field information.

That is why a numerical simulation of the self-induced sloshing motion was conducted in Saeki et al. (2001) to obtain detailed flow data. These numerical results were verified with experimental results and used to describe the sloshing motion quantitatively in terms of oscillation energy fed back to the sloshing motion. They concluded that the self-induced sloshing was generated by the nonlinear interaction between sloshing motion and jet fluctuation. The spatial phase state was found to be the most determining parameter for the sloshing growth.

In contrast with previous studies, it seemed that circulating flow and free-surface flow are less significant for the excitation mechanism. A conditional equation was derived that predicts a combination of the sloshing mode and stage for sloshing conditions in the same mode as the jet mode or for multi-mode sloshing. Finally, the feedback mechanism of Saeki et al. (1999) was validated in which the positive feedback energy for the sloshing motion is supplied by the fluctuation of the incoming jet.

Many studies have been done to investigate the nonlinearities of the free surface oscillation due to an external force (externally-induced sloshing). But only a few studies have focused on the self-induced sloshing phenomenon in the absence of external forces. The main limitation of these studies is that they are not applicable to the filling process of a lock. Most of them consider the self-induced sloshing motion in case a horizontally injected plane jet has a downward outlet on the bottom of the tank. Other studies focus on a vertically injected water jet, impinging at the free surface. The phenomenon of a horizontally injected water jet without an outlet has in contrast not been studied in earlier research and should be investigated in further research.

2.2 Physical insights in the phenomenon of self-induced sloshing

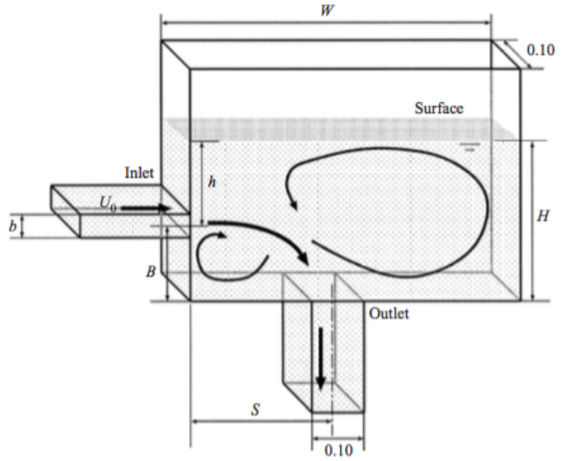
The most recent insights in self-induced sloshing, mainly based on the paper of Saeki et al. (2001), are described hereafter. Although this paper does not reveal new insights in the phenomenon, the cross-validation between experimental and numerical data makes their work the most scientific significant work on self-induced sloshing so far.

2.2.1 Flow regimes

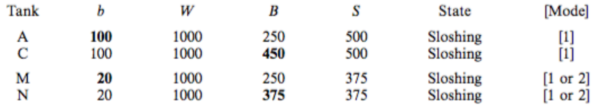
The self-induced sloshing phenomenon was studied in a series of simplified, two-dimensional rectangular test tanks as shown in Figure 2.1. The characteristics of these tanks are the tank width W, the tank depth from the front D, the inlet height from the bottom of the tank B, the outlet location from the left side wall S, the inlet width b and the outlet width s.

The determining parameters on the flow pattern within the test tank are the tank geometry, the jet inlet velocity U_0 and the mean water level H. Depending on these three factors, it was observed that three flow states can occur in the test tank which are schematically shown in Figure 2.2.

First of all, a stable state in which both the free surface and the flow pattern are stable and unique can be discerned. In that case, two streams can be distinguished: a submerged water jet from the inlet that turns downward to the outlet as well as an underwater stream from the



(a) Schematic view (dimensions in mm)



Tank	*b*	*W*	*B*	*S*	State	[Mode]
A	**100**	1000	250	500	Sloshing	[1]
C	100	1000	**450**	500	Sloshing	[1]
M	**20**	1000	250	375	Sloshing	[1 or 2]
N	20	1000	**375**	375	Sloshing	[1 or 2]

(b) Overview of dimensions (in mm)

Figure 2.1: Experimental test tank. From Saeki et al. (2001).

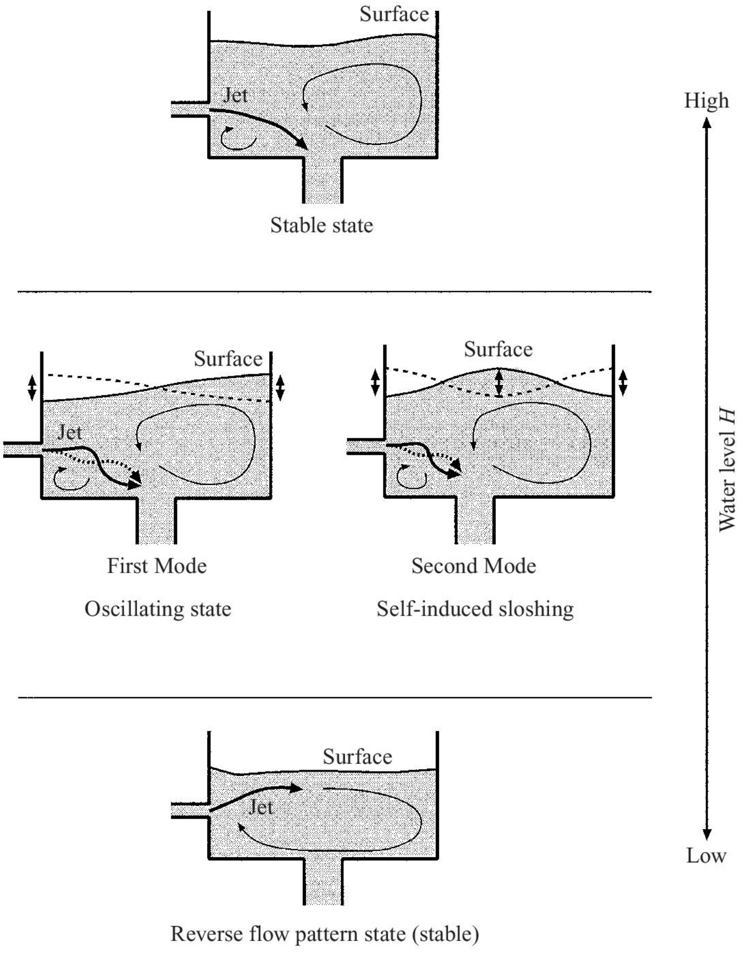


Figure 2.2: Overview of possible flow states in the tank. From Saeki et al. (2001).

far sidewall towards the inlet sidewall. This leads to a large counterclockwise circulating flow in the far side of the tank and a small clockwise flow below the jet at the entrance.

In case the water level becomes lower and closer to the inlet, a reverse flow pattern is distinguished. Only one large clockwise circulating flow, opposite in direction as in the stable state, is present and a rough wavy surface with significant bubble entrainment is formed.

A third pattern corresponds with self-induced sloshing, in which the water surface shows a periodic oscillation in time. First mode self-induced sloshing occurs when both ends of the surface move up and down alternately and a node appears in the midsection of the tank. Superimposed on this standing wave, a nonlinear wave is present which propagates against the free surface flow. In case of second mode self-induced sloshing, both ends oscillate in phase w.r.t. each other and with a phase difference of 180° w.r.t. to the midsection. Two nodes, at which the water surface remains stationary, are therefore present within the tank. Together with this sloshing motion, the jet also shows an oscillatory behavior, which will be further discussed later in this chapter.

2.2.2 Sloshing frequency

The theoretical frequency of the n^{th} mode of sloshing without circulating flow is considered to be equal to the eigenvalue of the water in the test tank. This can be calculated using following expression, given by Lamb (1932):

$$f_s^n = \frac{1}{2\pi}\sqrt{g\frac{n\pi}{W}tanh(\frac{n\pi H}{W})} \tag{2.1}$$

This formula, which predicts that the sloshing frequency increases with increasing water level, was verified with experimental results using the fast Fourier transform (FFT) on the time-series water level data. These theoretical derived frequencies were calculated based on the dimensions of the test-tank and are represented by the dotted lines in Figure 2.3. It can be noticed that the experimental results observed in the test tank show good correspondence with the predicted values.

It must be mentioned that the dominant frequencies of self-induced sloshing under the higher velocity condition were on average lower than theoretical eigenvalues, both in the experiments as well as in numerical simulations (see Section 2.2.3).

2.2.3 Excitation map

For one specific test tank setup, two separate sloshing regions of the same or different modes of oscillation are distinguished. Test tank A showed two conditions for self-induced sloshing: high velocity first stage sloshing and low velocity second stage sloshing. In case the inlet velocity corresponds to one of both stages, first mode sloshing will occur. For inlet velocities in between the first and second stage, a sustained water oscillation was not noticeable.

Secondly, only for tanks M and N second mode sloshing could be established. For those tank geometries, the inlet width b was reduced to one fifth compared to tanks A and C. The influence of the parameter b is however complicated as it influences both the inlet velocity and tank geometry. As such, a simple formula to account for the variation in b is difficult to derive.

Additionally, it was found that the jet oscillates asymmetrically and wavely and eventually becomes synchronized with the sloshing motion. The wave number of the jet fluctuation is in that case approximately twice as large compared to first stage sloshing: one or two large vortices are formed between the inlet and outlet for respectively first and second stage sloshing. These vortices are then transported along the jet towards the outlet of the rectangular tank. The spatial phase of jet fluctuation, called the mode of jet fluctuation, is therefore different for first and second stage sloshing.

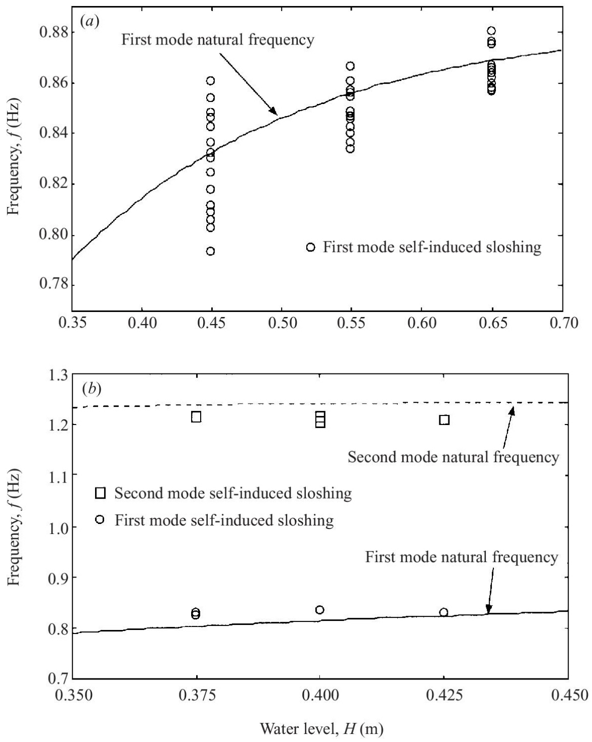


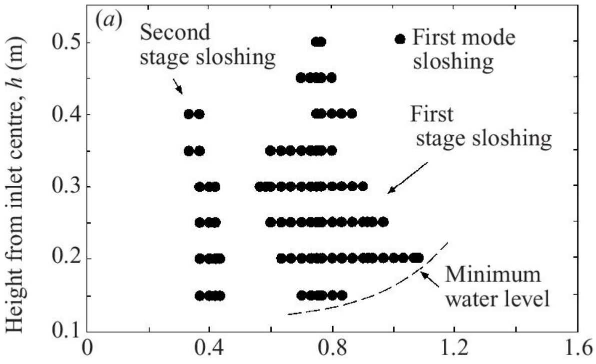
Figure 2.3: Frequency of self-induced sloshing for two different tank setups: a) Tank A; b) Tank M. From Saeki et al. (2001).

Finally, ‘multi-mode sloshing’ did not occur for every tank setup and was only observed in case of a small nozzle width. A decrease in the distance from the outlet to the inlet sidewall (S) in tank N appeared to result in ‘multi-mode sloshing’. In that case, the sloshing mode is different than the mode of jet fluctuation. The geometry of the test tank additionally determined whether first mode sloshing occurred at low velocities and second mode sloshing at high velocities or the other way around. A physical reason for this observation could however not be derived.

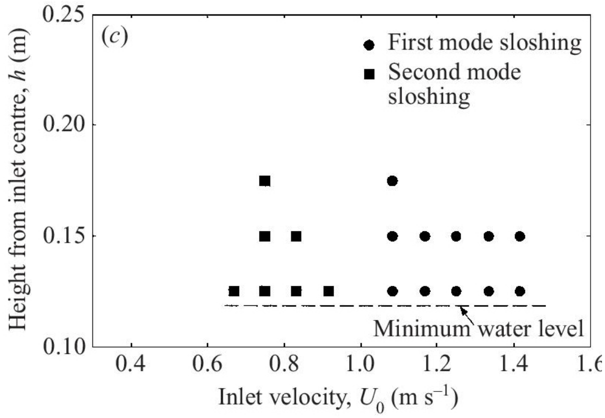
Saeki et al. (2001) derive in their paper excitation maps for self-induced sloshing, based on experimental data. These predict the occurrence of self-induced sloshing in function of the inlet velocity U_0 and inlet-surface distance $h = H - B$ for one specific geometric test tank setup. Figure 2.4 shows two examples of such experimentally derived excitation maps for two particular test tanks. As can be seen in Figure 2.4, both the free surface and the flow inside the tank tend to become stable with increasing h. A higher water level requires therefore more energy for free surface oscillations. However, when h becomes very small, the jet is directed towards the surface and a reverse flow pattern develops without further sloshing oscillations.

These experimental results were verified with a two-dimensional laminar code. This code, based on a numerical solution of the Navier-Stokes and continuity equation, simulates the transient flow with a free surface in a rectangular tank and includes the horizontally injected plane jet. Both the stable flow pattern as well as self-induced sloshing were simulated and the simulated frequencies and water motions were analyzed. The frequencies corresponding with self-induced sloshing showed good agreement with the theoretical frequencies.

Secondly, the simulated excitation map was derived in function of h and U_0 for the tank setup in which first mode sloshing occurred in the experimental study. Numerical self-induced sloshing was observed under two separate conditions: high velocity sloshing A and low velocity sloshing B. Between both sloshing conditions, a stable condition was present in which the free surface was not self-excited. As can be seen in Figure 2.5, these sloshing conditions correspond qualitatively



(a) First mode sloshing: a) Tank A, b) Tank C



(b) First and second mode sloshing: c) Tank M, d) Tank N

Figure 2.4: Experimentally derived excitation map. From Saeki et al. (2001).

with respectively first and second stage sloshing in the experiments.

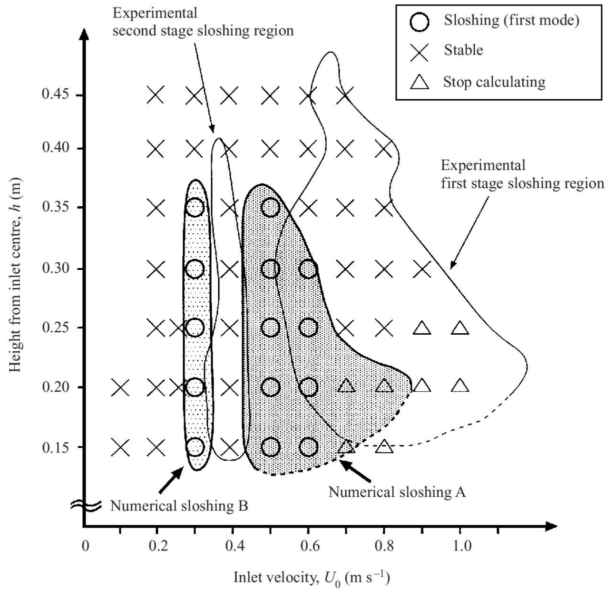


Figure 2.5: Simulated excitation map with first mode sloshing (Tank A). From Saeki et al. (2001).

2.2.4 Excitation mechanism

Self-induced sloshing is caused by a closed feedback mechanism, in which the flow-surface interaction enhances the sloshing motion (Figure 2.6). The free surface oscillation can be described in terms of potential flow, in which the sloshing potential is used to approximate the free-surface oscillation. The first mode sloshing potential is given by Lamb (1932):

$$\phi_s(x,y,t) = a\frac{\omega_s W}{\pi}\frac{cosh(\frac{\pi(y+H)}{W}}{sinh\frac{\pi H}{W}}\ sin(\frac{\pi x}{W}))\ sin(\omega_s t+\delta) \tag{2.2}$$

where ω_s, W and H denote respectively the angular sloshing frequency, tank width and water level in the tank. The amplitude a is determined as the average value of the water surface fluctuations. In this study, the phase delay δ was set to zero, as the phase-averaged data were measured using the surface phase information.

The influence of the unsteady flow in the test tank can be represented as the force on a fluid volume F_{unst}, which can be calculated by means of the momentum theory (volume integrated form of Navier-Stokes equation):

$$\underbrace{\frac{d}{dt}\int_V \rho u \ \mathrm{d}V}_{F_{unst}} = \underbrace{\int_S \rho u(u\cdot n)\ \mathrm{d}S}_{F_{con}} + \underbrace{\int_V \rho(grad\ p+\rho g)\ \mathrm{d}V}_{F_{press}} + \underbrace{\int_V \rho\nu\nabla^2 u\ \mathrm{d}V}_{F_{diss}} \tag{2.3}$$

where S and V denote the surface and volume of a control surface over the entire width of the tank and n is the normal vector on this control surface.

The oscillation energy supplied for sloshing E_n in a control volume per unit of time can then for each term in eq. (2.3) be calculated using the sloshing velocity $grad(\phi_s(x,y,t)$:

$$\Delta E_n(x,y,t) = F_n(x,y,t)\cdot grad(\phi_s(x,y,t)) \tag{2.4}$$

with n: con, press, diss and unst from eq. (2.3). The local oscillation energy over a natural sloshing period T_s can be found from:

$$\overline{\Delta E_n(x,y)} = \int_{T_s} \Delta E_n(x,y,t)\ \mathrm{d}t \tag{2.5}$$

This is finally space-integrated for the whole field (test tank), which results in the total oscillation energy supplied for sloshing motion:

$$E_n = \frac{1}{WHa^2}\int_{S_{tank}} \overline{\Delta E_n(x,y)}\mathrm{d}S \tag{2.6}$$

where S_{tank}=two-dimensional area of the tank. The oscillation energy is subsequently converted into energy per unit mass $S_{tank} = WH$ and the squared amplitude a^2.

In case this procedure is followed for every force in the Navier-Stokes equations, the total oscillation energy for the unsteady term can be found by the sum of the other oscillation energies:

$$E_{unst} = E_{con} + E_{press} + E_{diss} \tag{2.7}$$

In case this total unsteady oscillation energy is positive, self-induced sloshing will be excited. The different terms in eq. (2.7) were determined based on the numerical simulations. It was observed that with increasing water level, E_{unst} becomes smaller and less energy is available for the sloshing of the water mass. For negative E_{unst}, the sloshing motion even becomes damped and a stable situation is in that case established. When the contributions of the different energies are considered, it can be found that the pressure term E_{press} is independent of sloshing and has no impact on sloshing growth. Additionally, the dissipation term E_{diss} is very small and its effect can therefore be neglected. As such, the sign of the oscillation energy of the convection term corresponds very closely with the sign of the energy for sloshing growth. The nonlinear force causes by flow variation, which is the driving force behind the convection energy, is therefore the main excitation mechanism of self-induced sloshing.

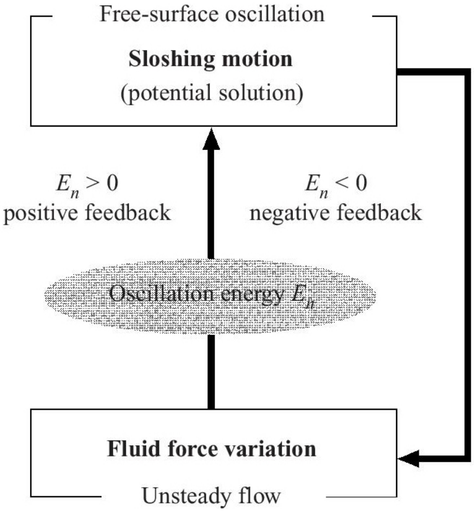


Figure 2.6: Feedback process for self-induced sloshing. From Saeki et al. (2001).

In case one considers the magnitude of the oscillation energy distribution for the convection term $\overline{\Delta E_n(x,y)}$, it is found that the absolute value of the local oscillation energy is the largest along the jet.

Figure 2.7 shows the sloshing energy distributions for one particular test case, calculated based on the measured velocity distributions. Its sign changes along the flow path from inlet to outlet and the number of these sign oscillations decreases with increasing inlet velocity. In case the total amount of positive peaks is less than the amount of negative peaks along the jet, sloshing inside the tank is damped. Additionally, it was observed that even if the number of positive and negative peaks are equal, a minimum inlet velocity U_0 is required to obtain a positive growth rate. At very low inlet velocities, the inlet energy seems insufficient for E_{con} to sustain self-induced sloshing.

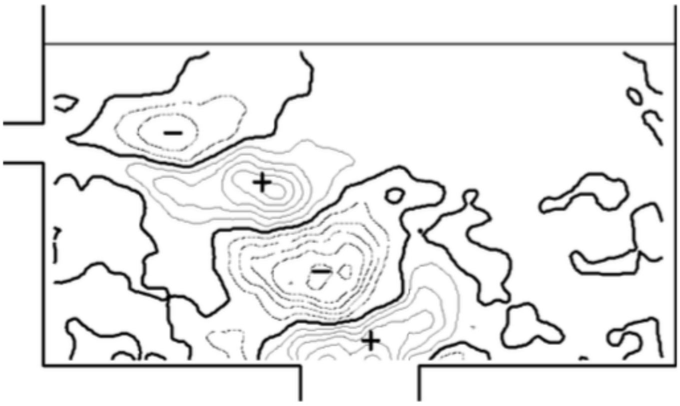


Figure 2.7: Sloshing energy distribution. From Okamoto et al. (2000).

This was verified by numerical simulations, which confirmed that in case the amount of positive peaks is equal to the amount of negative peaks, self-induced oscillation is possible. The amount of the (+/-) energy pattern pairs along the jet also determines the stage of self-induced sloshing: 1^{st} and 2^{nd} stage sloshing occurred for respectively 1 or 2 pairs. The number of pairs was observed to be inversely proportional with inlet velocity U_0. The results of these numerical studies is shown in Figure 2.8.

It can be concluded that the spatial phase state of the jet fluctuation, responsible for the emerging of large vortices along the jet, determines the sloshing regime inside the tank. The circulating flow pattern and under-surface flow are in contrast not contributing to the sustainable energy for sloshing.

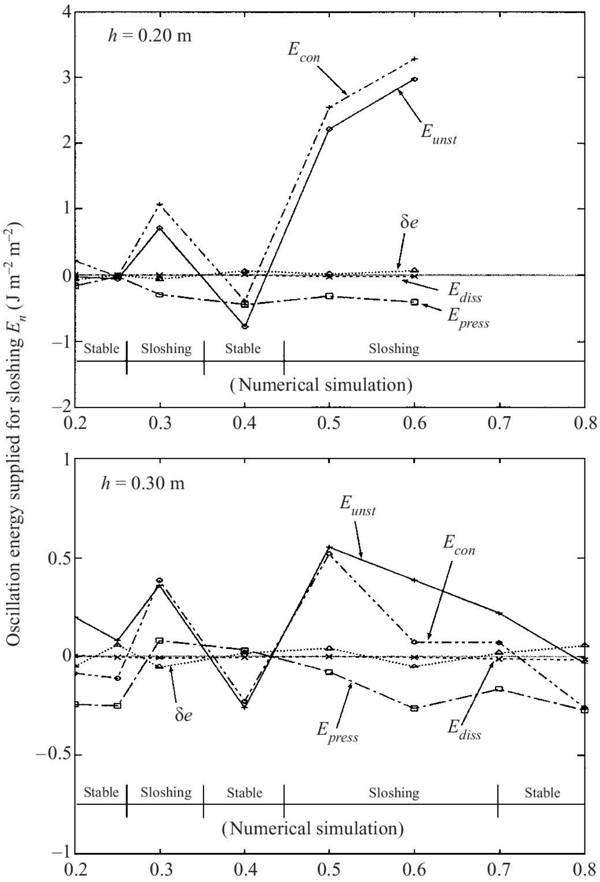


FIGURE 23. Oscillation energy supplied for sloshing E_n.

Figure 2.8: Oscillation energy supplied to self-induced sloshing. From Saeki et al. (2001).

2.2.5 Conditional equation for self-induced sloshing

The unstable behavior of a jet is normally described by the use of the Strouhal number:

$$St = \frac{f_r l_r}{U_r} \tag{2.8}$$

in which f_r, l_r and U_r are respectively the representative frequency, length and velocity. In case the jet fluctuation is synchronized with the sloshing motion, the frequency in the Strouhal number equals the natural frequency of sloshing f_s^n.

Because the excitation of self-induced sloshing is largely dependent on the tank geometry, the distance from inlet to outlet $L = \sqrt{B^2 + S^2}$ is used as representative length. Turbulent shear flow along an unsteady jet results in turbulent jet fluctuations. These have different phase velocities but show a certain dominant phase velocity u_{con} due to their mutual interaction. The representative velocity depends on the longitudinal distance along the jet due to spatial damping. It is known that the velocity gradient of the jet is the largest where the jet velocity is approximately half on the jet centerline velocity u_m. Blake (1986) suggests u_{con} is taken at this most unstable position:

$$u_{con} \approx \frac{1}{2} u_m(x) \tag{2.9}$$

where x represents the distance along the jet in the direction of propagation.

Based on these results, a modified Strouhal number can be defined for a turbulent jet in conjunction with a sloshing motion:

$$St_s = f_s^n \int_0^L \frac{1}{u_{con}} \mathrm{d}x \approx f_s^n \int_0^L \frac{1}{\frac{u_m}{2}} \mathrm{d}x \tag{2.10}$$

St_s can be considered as the time necessary for a turbulent jet disturbance to propagate a distance L from inlet to outlet, expressed as a multiple of the sloshing period. St_s is therefore an indicator of the wavenumber of the turbulent jet, more precisely the number of large vortices that emerge from inlet to outlet over one sloshing period.

The integration of eq. (2.10) was done by using an experimental velocity distribution of the turbulent free jet. Based on the work of Abramovich and Schindel (1963), the effect of spatial damping can be taken into account by assuming following velocity distribution:

$$u_m(x) = \frac{1.2U_0}{\sqrt{\alpha(x + 0{,}41b)/b}} \tag{2.11}$$

where b is the inlet width and $\alpha \approx 0.22$ an experimental constant.

It must be mentioned that this expression is a rough simplification as it neglects effects of the tank walls, free surface and circulating flow, ... In case these effects are neglected, substitution of this expression in eq. (2.10) results in:

$$St_s = \frac{f_s^n L}{U_0} \kappa \sqrt{\frac{L + 2b}{b}} \qquad \text{in case } L >> b \tag{2.12}$$

in which the experimental constant $\kappa \approx 0{,}67$ and the term $\sqrt{\frac{L+2b}{b}}$ allows to account for the tank geometry.

Under certain conditions of the modified Strouhal number, self-induced sloshing is excited. Based on experimental results of several authors (Okamoto et al. (1991); Okamoto et al. (1993); Fukaya et al. (1995)), a conditional equation was derived that needs to be fulfilled for self-induced sloshing to occur:

$$\begin{aligned} &m - 0.25 < St_s < m + 0.25 \\ &\text{with (m=1: first stage or m=2: second stage)} \end{aligned} \tag{2.13}$$

where m is an integer and equals the number of large vortices that occur along the jet during one sloshing period. The value of $m \pm 0.25$ represents a quarter phase of the jet oscillation $= \pm\pi/4$. This conditional equation is visually shown in Figure 2.9 for different tank geometries. Depending on the tank geometry, initial water depth and inlet velocity, either the sloshing mode or jet stage can change.

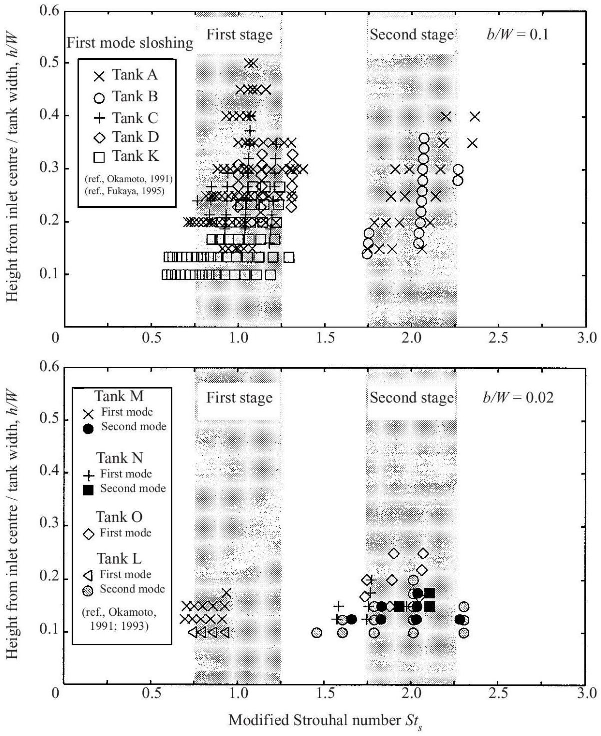


Figure 2.9: Excitation condition for self-induced sloshing in function of St_s. From Saeki et al. (2001).

The excitation map for sloshing can in an alternative representation be expressed in function of the Reynolds number Re and St_s (Figure 2.10). The Reynolds number for sloshing is defined as (Saga et al. (2000a)):

$$Re = \frac{\rho U_0 b}{\mu} \tag{2.14}$$

where ρ, U_0, b, μ are respectively the water density, inlet jet velocity, height of the test tank inlet and dynamic viscosity of water.

The sloshing motion is defined by its jet mode (stage) m and sloshing mode n. As the natural frequency f_s^n for mode n=1 or n=2 is fixed for a certain water level and tank geometry, the relationship between Re and St_s becomes hyperbolic Re $\propto$ St_s. Self-induced sloshing occurs at the intersections of these hyperbolas with the regions that are described by the conditional equation. The combination (n,m) of sloshing mode and jet-stage is dependent on the tank geometry, initial water depth and inlet velocity U_0. Sloshing and jet fluctuations can occur at the same mode but also multi-mode sloshing ($m \neq n$) can occur at different and separate conditions of U_0.

2.3 Conclusion

The results given earlier in this chapter allow to obtain a good understanding and description of the excitation mechanism of self-induced sloshing. In case of self-induced sloshing, the jet fluctuation becomes synchronized with the sloshing motion. The jet is that case fluctuated by the pressure oscillations that are synchronized with the sloshing motion. The inlet velocity U_0 and tank geometry, defined by the inlet-outlet distance L and inlet width b, determine the behavior of this horizontal injected jet. This jet fluctuation generates one or two large vortices

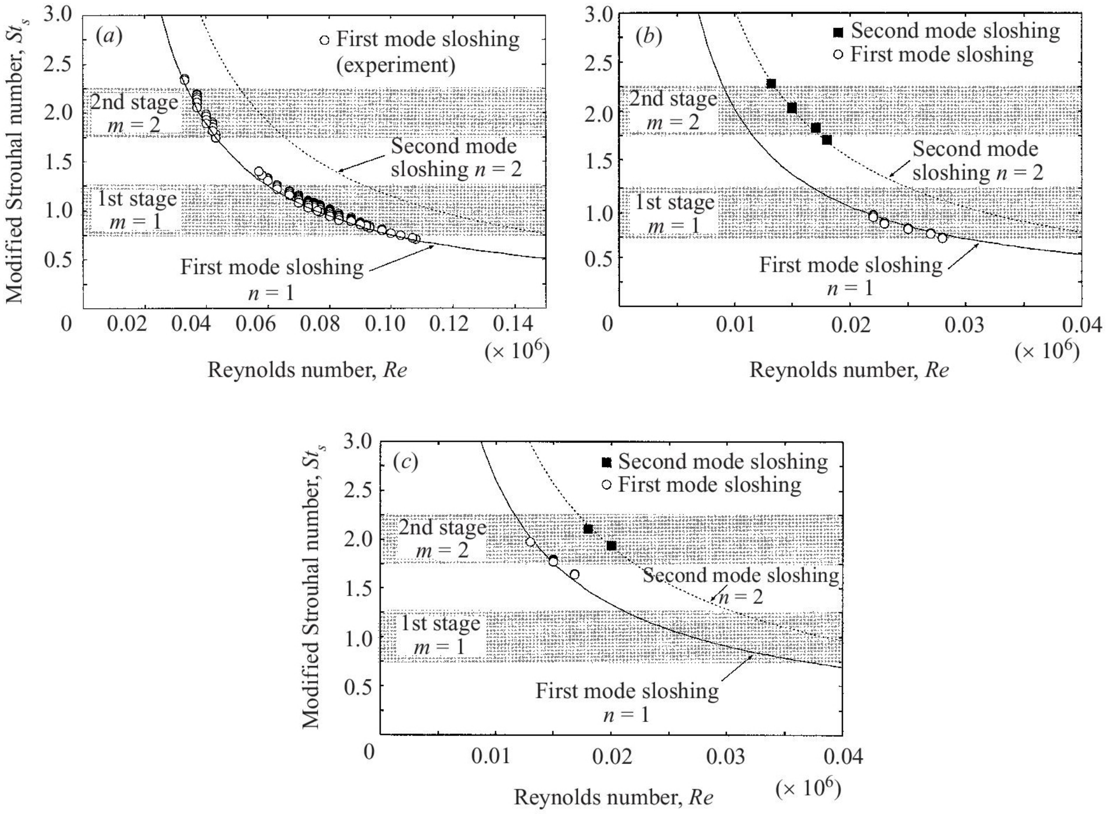


Figure 2.10: Relationship between Re and St_s for different tank setups: a) Tank A; b) Tank M; c) Tank N. From Saeki et al. (2001).

that are transported along the jet from inlet to outlet, the number of which is dependent on the previously mentioned parameters.

In case the large vortices satisfy eq. (2.13), positive feedback energy ($E_{con} > 0$) will be supplied to the sloshing motion by the jet fluctuation. The spatial and temporal interaction of the jet fluctuation and sloshing motion can therefore be considered as the excitation mechanism of self-induced sloshing. In that case, the feedback loop is closed and the sloshing motion in the tank is self-excited and will grow. However, it must be mentioned that the predicted self-induced sloshings, based on eq. (2.13), cannot be excited in certain conditions. Experiments showed that for too low inlet velocities or too high flow velocities below the free surface, sloshing was not self-excited. A graphical representation of this excitation mechanism is shown in Figure 2.11.

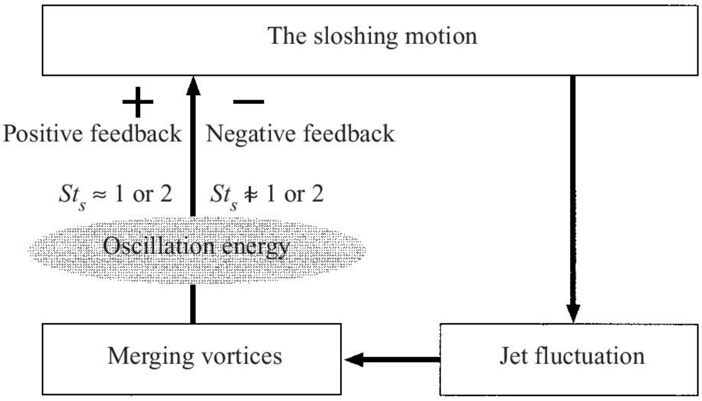


Figure 2.11: Excitation mechanism of self-induced sloshing. From fromSaeki et al. (2001).

Chapter 3

Image-based surface reconstruction

This chapter explains different methods that have been developed to obtain three-dimensional information of objects in a scene. Our goal is to develop a method that allows to describe the spatio-temporal variability of a water surface in an experimental test setup. Because these measurements should interfere as less as possible with the hydrodynamic phenomena that are reconstructed, optical techniques are for this purpose preferred over traditional mechanical measurement techniques. The latter could cause direct changes to the flow pattern and the resulting water surface. Additionally, mechanical devices could require adaptations of the test tank/flume which results in a less realistic representation of the real-life scene to be modeled.

Hence, we focus on image-based approaches as they allow high accurate measurements without interfering with the scene itself. The overview that is given in the rest of this chapter serves as reference work to gain insight in the possibilities of optical techniques. Additionally, it serves as a source of possible adaptations and improvements to the technique that is developed during this thesis. The finally chosen technique is explained in Chapter 4, on the basis of the considerations made in Section 3.3.

3.1 Difficulties in water surface reconstruction

The reconstruction of a time-varying water surface has been subject of many studies but still remains a challenging problem. The analysis of liquids from video recordings and images comprises several difficulties compared to traditional 3D photography applications. Normally, image-based 3D reconstruction is used to determine the 3D-shape of static objects. As pointed out by Morris and Kutulakos (2005), liquids are in contrast much more difficult to capture due to several reasons:

- **No prior scene model.**
  Because the spatio-temporal evolution of water is only constrained by the laws of fluid mechanics, a low-degree-of-freedom parametric model can for such scenes be unreliable. The error or inaccuracy due these simplifications is however largely dependent on the phenomenon that is studied. Regular waves can for example be described by a sinusoidal function that describes their spatial and temporal variability. In contrast, the complex surface shape in case water is poured into a glass would be very hard to approximate by a theoretical solution.

- **Nonlinear light path.**
  Due to refraction at the interface of water and air, incident light rays are broken and a point below the surface will therefore be projected along a nonlinear path to a viewpoint above the water surface.
- **Shape-dependent appearance modulation.**
  Absorption, scattering and Fresnel transmission cause the appearance of points below the surface to depend on the light's path and hence on the surface shape. Moreover, these effects are dependent on the wavelength of light and therefore different colors of the visible light spectrum will be affected in a different way.
- **Turbulent behavior.**
  As liquid flow is a volumetric phenomenon, not only the time-varying surface but also the vector field describing the internal motion must be determined.
- **Instantaneous 3D capture.**
  Because liquids behave dynamically and the shape of the water surface can change rapidly, instantaneously captured information is necessary to analyze the time-evolution of the water surface.

Multiple techniques to determine the shape of objects captured in a sequence of images are available. In the overview that is given in the rest of this chapter, most methods are however not described in detail because they are not suitable to capture water surface variations.

3.2 Overview of optical reconstruction techniques in computer vision

Computer vision comprises a broad research domain, which focuses on mathematical techniques to recover the three-dimensional shape and appearance of objects in a scene. Based on analyzing and processing these images, computer vision techniques allow to steer artificial systems and reconstruct the properties of objects (e.g. shape, color, ...) depicted in those images (Szeliski (2010); Bradski and Kaehler (2008)). This is difficult because in theory there exist an infinite amount of combinations of 3D scenes that correspond with the same 2D image. Noise, insufficient image quality and distortion even further complicate this ill-posed problem. For more information about computer vision techniques and image processing, we refer to Szeliski (2010). We will limit ourselves to two basic optical techniques in computer vision: appearance modeling and stereo reconstruction. After this basic introductory, we elaborate the different techniques that have been developed to reconstruct specular reflective, transparent and refractive objects.

3.2.1 Fundamental techniques in computer vision

3.2.1.1 Appearance modeling

Although the problem of 3D modeling and simulation of 3D objects is not in the scope of this thesis, some basic understanding about generating novel views of a 3D scene is required. A brief summary of 'appearance modeling' is given hereafter because several methods discussed later in this chapter are based on techniques developed in this branch of computer vision.

'Appearance modeling' captures the appearance of the scene through images and subsequently produces novel views of the scene (new viewpoints or modifying other aspects such as background). For this purpose, no shape information of the scene itself is however extracted. In the following, we describe the two mostly used techniques: 'plenoptic measurements' and '(environment) matting'.

a. Plenoptic measurements
In plenoptic measurements, images are rendered by sampling the so-called 'plenoptic function'. This is a parameterized function to describe everything that can be seen from all possible viewpoints in the scene.

In a full description of the plenoptic function P, this is function of 7 parameters. P depends on one hand on the spherical coordinates θ and ϕ that describe the direction of the light rays that converge in one point. The location of this converging point is denoted as (V_x,V_y,V_z). In case the scene is captured by a camera, the light rays converge in the center of projection of the camera. The light intensity of each ray is also dependent on the wavelength λ and in case of temporal sequences, a time-parameter t must be included. This results in the plenoptic function $P(\theta,\phi,\lambda,t,V_x,V_y,V_z)$ which fully describes the light rays converging in one particular point. In case P is known for every point in the scene, this allows to calculate the plenoptic function parameterized by the camera's field of view with the converging point located at the camera's center of projection.

It must be noted that it is most often assumed that the plenoptic function is redundant in 'free-space', i.e. with no occluding objects. As such, the intensity of the light rays remains the same as long as they do not strike an object. In case the sampled light rays converge on the camera's center of projection, the CCD elements (charge-coupled devices) of the camera record the light intensity. The image pixels values can then be considered as the average intensity of the rays that are recorded by the light receptor of the corresponding pixel. By taking many images of a scene ('sampling the plenoptic function'), new views of the scene can be created using some sort of interpolation of the recorded images.

Sampling the plenoptic function for imaging water remains however difficult. Firstly, the water surface is dynamic which makes sampling from all expected angles at one particular time instance necessary. Secondly, water's optical properties such as reflection and refraction cause that its appearance is actually the reflected or refracted appearance from another object in the scene. It can be concluded that this technique is not ideal to capture the position and shape of fluctuating water surfaces and will therefore not be further elaborated.

b. Matting and environment matting

The technique of 'matting' separates the background from the foreground in images. To that end, a matte is constructed which is opaque over the background, partially transparent at the edges of the foreground and fully transparent over the foreground.

Zongker et al. (1999) propose a technique called 'environment matting' to approximate the appearance for transparent objects in the foreground. In case the foreground object reflects/refracts part of the background light, it is not possible to simply exclude the background of the image. The environment matte allows to model the effects of reflection, refraction, translucency, gloss and inter-reflection.

For every pixel in the foreground image (image of the foreground object), a function is created which includes the original matte as well as the contribution of the reflected and/or refracted background light. The technique relies on taking several images of the object that needs to be reconstructed, with structured textures on screens surrounding the object.

Each screen of the texture set is composed of a hierarchy of vertical and horizontal stripes with varying widths. The pattern used is also different for every screen as is shown in Figure 3.1. The color of every foreground pixel is composed of direct part originating from the color of the foreground object and a contribution of the background (patterns on the screens). The latter comprises both a reflected and refracted part, each corresponding to a different pattern/screen. Several images are taking with varying stripe thickness and orientations of the patterns. Next, an objective function is minimized over the series of photographed images for each covered (foreground) pixel in the scene. This finally allows to determine the axis-aligned rectangular patches of the background that best approximate the reflection and refraction on the foreground object. Using this knowledge, novel views of the foreground object in which it is placed into a new environment (background) can artificially be produced. This technique has however two disadvantages.

First of all, it requires static objects because multiple images must be captured to estimate mappings from the background through the foreground object. Chuang et al. (2000) solve this issue by using a darkened room and a single color gradient map background. This allows to capture and matte dynamic refractive objects against arbitrary backgrounds, although it remains limited to one single viewpoint. The second limitation is the use of a fixed viewpoint. Several methods have however been developed to determine the reflectance of objects captured from multiple viewpoints. These techniques remain limited to static objects and are consequently unsuitable for water surface reconstruction.

3.2.1.2 Stereo reconstruction

Stereo reconstruction is used frequently to determine the shape of objects in a scene. Initially, we will consider basis stereo reconstruction of Lambertian scenes. A Lambertian surface is defined as a surface which is characterized by diffuse or Lambertian reflectance and the luminous intensity obeys in that case Lambert's cosine law. This basically means that the brightness of a particular

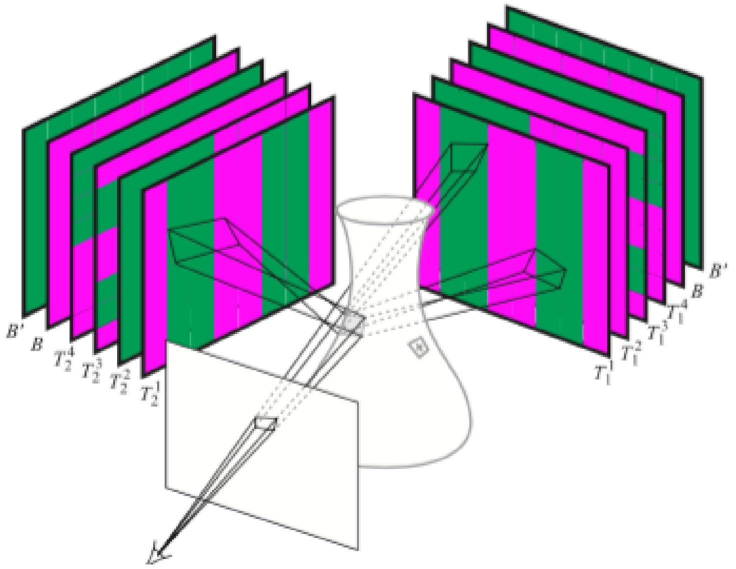


Figure 3.1: The environment matting process: multiple screens with varying patterns are placed around the object and their contribution to the appearance of the object is used to construct an environment matte for the refractive and reflective object. From Zongker et al. (1999).

point on the surface is independent on the observers' view angle. Because a water surface is non-Lambertian, several difficulties have to be overcome which will be explained further in this chapter.

Stereo reconstruction uses the same principle as the human visual system in which the parallax of both eyes is matched to determine the position of a specific point observed by both eyes. The basis of stereo reconstruction consists of computing the depth (z) of a point from the stereo baseline based on the disparity between two images of a certain point. Disparity can be defined as the distance between two corresponding points in the left and the right image of a stereo pair.

Figure 3.2 explains the concept of disparity and disparity mapping. A point X will be seen in the left image plane at $X_L = (u,v)$, which can be found by the intersection of the line $X - O_L$ with the image plane. The same can be done for the right image plane for which a location $X_R = (p,q)$ is found. The difference in position of the point X in the corresponding two (left and right) images, measured in pixels, is denoted as 'the disparity'.

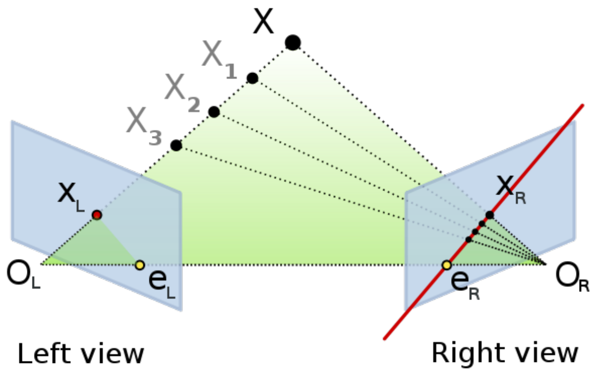


Figure 3.2: Principle of stereo reconstruction using disparity between images. Author unknown.

In order to find the disparity of a point, it is necessary to find the match (called 'corresponding point') in the other image plane. After finding the match for every pixel in the left image, the distance between two corresponding pixels is computed. This results in a disparity value for every pixel in the left image, which is inversely proportional to the depth along the line $X - O_L$. Using the computed disparities, it is possible to define a mapping from a (u,v,d)-triple to a three-dimensional position.

In conventional stereo vision, the depth (z) of a point is, as mentioned earlier, defined as the distance from that point to the stereo baseline, i.e. the line connecting the centers of projection of both views. For the simplified case of a binocular view with two cameras having parallel optical axes and the same focal length, the depth z can be determined as followed:

$$z = \frac{b \cdot f}{d} \tag{3.1}$$

where b and f denote respectively the length of the stereo baseline and the focal length of the cameras as is shown in Figure 3.3.
The corresponding x- and y-coordinates can then be determined using following set of equations:

$$\begin{cases} x = u \cdot z/f \quad \text{or} \quad b + p \cdot z/f \\ y = v \cdot z/f \quad \text{or} \quad q \cdot z/f \end{cases} \tag{3.2}$$

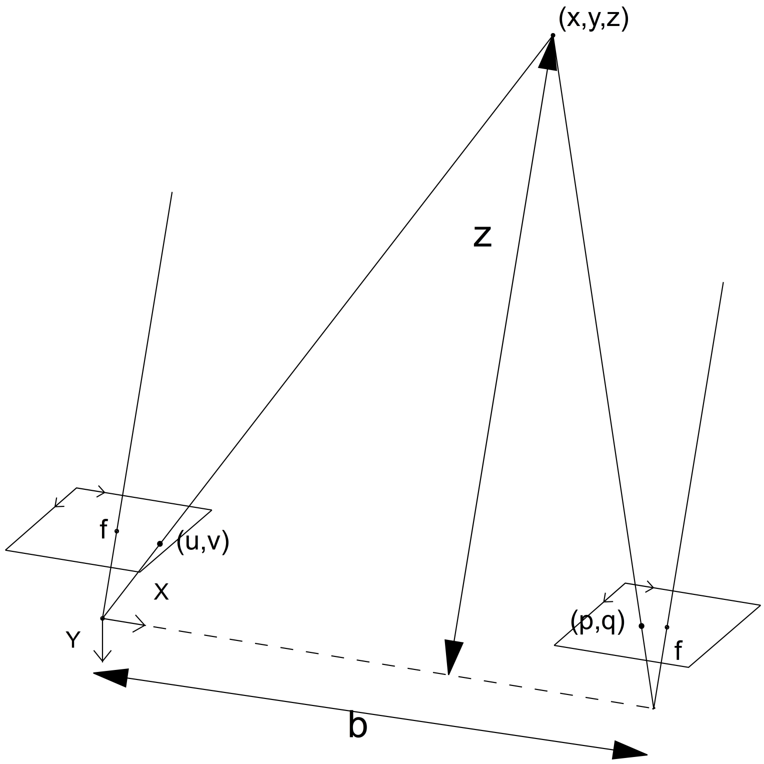


Figure 3.3: Two cameras separated by a baseline b, observing the same object at distance z from the stereo baseline. Based on Bradski and Kaehler (2008).

This methodology can be used to determine the 3D position of a point for the more general case in which both the focal length and optical axes of the cameras are not the same. For simplicity's sake, this will however not be elaborated in this short overview.

In most of the stereo algorithms, the stereo reconstruction of an object comprises three stages which will be elaborated below.

Matching cost determination

In this first step, correspondences between views is determined in order to assign a certain cost to each candidate: the squared or absolute difference between the assumed matching pixels in the stereo views is usually adopted. Several techniques have been developed to improve this matching. In order to make the technique more robust against mismatched pixels, Black and Anandan (1993) use robust estimators, truncated quadrics or Lorentzian estimators to eliminate outliers. Cox et al. (1996) propose a pixel-based stereo technique in which matching is performed on the individual pixel intensities. The algorithm proposed optimizes a maximum likelihood cost function, subject to ordering and uniqueness constraints. This means that they assume that corresponding features in the left and right images follow a common normal distribution.

Birchfield and Tomasi (1998) match pixels in one image by interpolated sub-pixel offsets in the other image. This technique showed to be insensitive to the image sampling because it uses the linearly interpolated intensity functions surrounding the pixels. Okutomi and Kanade (1993) attempt to derive a stereo matching technique with less mismatches using multiple stereo images from different baselines. This proves to be efficient and especially useful for objects with repeated textures or edges.

In case of stereo reconstruction of water, the water surface is rather smooth and will exhibit few occlusions when viewed from overhead. This makes the matching cost techniques developed for handling discontinuities normally unnecessary (in case splashing etc. is avoided). The optical properties of water however require that the matching technique needs to be performed on an image projected on the water surface. As the water surface fluctuations will distort this image, applying stereo reconstruction to water surfaces inhibits several challenges which will be dealt with in Section 3.2.2.1.

Cost aggregation

Subsequently, the aggregation of the costs of all points is done in which a 'matching window' around the pixel of interest is constructed. The intensity difference in the left and right image corresponding to the assumed disparity is subsequently summed over the entire window. For this purpose, use can be made of square windows, shift-able windows or even windows with adaptive size. The sum of the intensity differences in these windows can be taken as the sum of absolute differences, squared differences, normalized cross correlation, ... Additionally, Scharstein and Szeliski (1998) proposes a technique named 'iterative-diffusion'. The weighted cost of neighboring pixels is in this approach added to the cost of the pixel of interest.

Disparity computation and refinement

One has to distinguish between global and local stereo methods: in global methods the correspondence is found for every pixel whereas local methods match distinctive features between images.

In global methods, the disparity values are determined for every point after which an optimization procedure is followed to minimize the sum of a certain energy function. This energy function is typically expressed as:

$$E(d) = E_{data}(d) + \lambda E_{smooth}(d) \tag{3.3}$$

where the $E_{data}(d)$ measures the pixel similarity, meaning how much the left image is matched in the right image. $E_{smooth}(d)$ penalizes disparity variations, which can be associated with the smoothness in the support region around each point. Extensive research has been done on robust smoothing functions. Also several minimization/optimization routines to minimize the energy function have been developed. These will however not be further elaborated in this thesis.

In local or window disparity methods, each point is optimized separately by aggregating the cost of the neighborhood or support region as was explained in the previous paragraph. The advantage of local (area-based) approaches is that the computation cost is much lower than global methods. The disadvantage of these methods is however that points in the reference image may not have 1-to-1 mapping to points in the second image. This makes the optimization for these points obviously difficult.

3.2.2 Reconstruction based on specular reflection

Reconstruction of water surfaces is challenging because they are highly specular reflective and therefore no Lambertian shading model can be obtained. For non-Lambertian scenes, the color or intensity of a specific 3D patch is not independent of the angle of the viewing rays w.r.t. the surface. In this section, several methods are discussed to deal with the reconstruction of specular reflective but still opaque (non-transparent) scenes.

Perfect (mirror-like) specular reflection is defined as the reflection in which light (or any kind of waves) is reflected by the surface in a non-diffuse manner. For a single incoming ray, the ray is (partly) reflected in only one outgoing direction. The relationship between these two rays is governed by the law of reflection, which states that the angle between the surface normal and the incoming or outgoing ray stays the same. Specular reflection is valid for smooth surfaces, such as relatively quiet water, for which a bundle of light will stay concentrated when it leaves the surface after reflection.

Surfaces that are microscopically rough (e.g. clothing or an asphalt roadway) are on the other hand characterized by diffuse reflection. In that case, the surface imperfections have a similar or even larger magnitude then the wavelength of the incoming light. Light is then reflected in a diffuse manner and the rays of a single bundle of light leave the surface in different directions. A graphical representation of specular and diffuse reflection is shown in Figure 3.4.

Surface reconstruction of specular surfaces is difficult because surface features of the reflecting object cannot be observed directly. In contrast, the surface shows reflections of its environment. The location of particular features from these indirect views of the original object changes in case the viewpoint is altered. In case the viewing ray and a 3D world position of the incident light ray on the surface are fixed, the depth and surface normal are determined up to a one-dimensional family of possible solutions. This depth-normal ambiguity can subsequently be solved by assuming distant illumination or by measuring an additional point on the incident light ray.

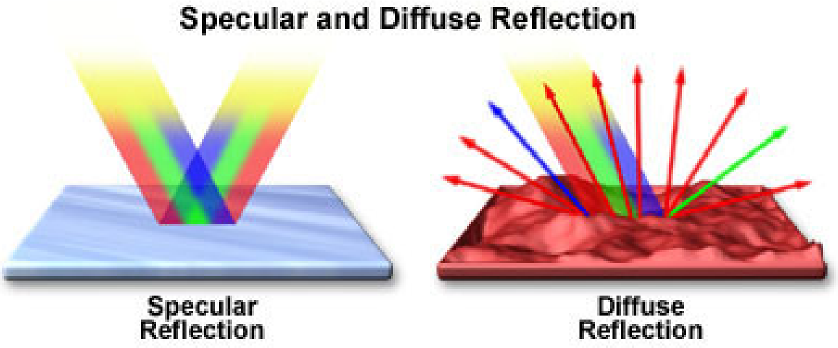


Figure 3.4: Comparison between specular and diffuse reflection. From Davidson and The Florida State University (2013).

Additionally, the complexity of the projection process of original objects on the mirror surface has led to new image-based techniques such as environment matting which was explained earlier. These approaches do not compute the shape of objects but only determine a function that maps points of a pattern placed near the scene to pixels in the distorted and reflected view of the pattern in the image.

Reflection-based approaches all assume non-transparency and therefore seem inappropriate to determine the shape of water surfaces. However, by mixing the water with a certain substance that makes the water non-transparent (e.g. kaolinite clay), the water surface becomes approximately an opaque scene. Methods used for mirror-like objects could therefore theoretically be applied and are investigated hereafter. The rest of this chapter is based on previous work of Ihrke et al. (2010) and Morris (2004). The former make a comparative study of the different techniques that have been developed to reconstruct the surface shape of specular or transparent objects.

The general classes of optical techniques that will be described in this chapter[1] are listed in Table 3.1. Although most of these techniques are able to reconstruct the shape of general 3D objects in a scene, we will focus on their applicability to reconstruct a dynamically changing water surface. Table 3.1 shows that the different techniques can be grouped into three main categories, each exploiting a different optical property of water.

---

[1] This review is based on extensive research done by Ihrke et al. (2010). We do not claim to be authentic and refer for a more elaborated comparison to the aforementioned authors. In this thesis, only a brief summary of their work is given to obtain more insight in the possibilities of image-based techniques.

Table 3.1: Overview of the general classes of optical techniques to reconstruct the shape of 3D objects. Based on Ihrke et al. (2010).
Note that this list is only a selection of the methods available in literature, in which we focus on methods which seem suitable for water surface reconstruction.

	Refractive objects	**Specular reflective objects**	**Transparent objects**
Stereo reconstruction/ Multi-view reconstruction		**X**	**X**
Photometric stereo reconstruction		**X**	
Shape from polarization		**X**	**X**
Shape from distortion	**X**	**X**	
Shape from specularities/reflectance	**X**	**X**	
Direct ray measurements	**X**	**X**	
Laser rangefinders	**X**	**X**	
Computerized tomography			**X**
Separation of local and global light transport			**X**
Inverse ray-tracing	**X**		
Thermographic surface reconstruction	**X**		
Shape from refractive irradiance	**X**		

3.2.2.1 Reflective stereo reconstruction

Several methods have been developed to obtain a stereo reconstruction of objects that mirror light (for which a Lambertian shading model is not valid). A first approach is to detect and subsequently remove specular highlights in the data by treating them as outliers. Bhat and Nayar (1995) use a trinocular stereo system, in which pairs of images are compared to identify these highlights. Nayar et al. (1997) suggest to remove specular highlights in a pre-processing stage by separating the specular and diffuse components of reflection from the images. Blake and Brelstaff (1988) use ad-hoc constraints that describe deviations from Lambertian reflectance to identity these highlights.

A second approach does not assume Lambertian reflectance (color constancy of a common feature between view points). This approach generalizes the multi-view matching constraint to obtain a more sophisticated model of the color variation between the different views. Stich et al. (2006) detect discontinuities in epipolar plane images using a multi-view stereo setup. Yang et al. (2003) use a linear color variational model, based on the observation that the reflected colors for most real-world surfaces are co-linear in the RGB color space. Next, they develop a photo-consistency measure, valid for both specular and Lambertian surfaces, which can be incorporated in existing space carving methods.

Jin et al. (2003) use the color variation to make a constrained radiance tensor, which defines the discrepancy between the image and the underlying model. Assuming the rank of this tensor equal to 2, this allows to reconstruct the surface while the reflectance properties of the surface are simultaneously estimated. Additionally, the estimate of the radiance can be used to generate novel views of the non-Lambertian appearance of the scene.

3.2.2.2 Photometric stereo reconstruction

A second set of techniques can be classified under 'photometric stereo', which uses the variation in illumination for a static view-point to compute shape. For a point source of light, the fraction of the incident light that is reflected in a given direction can be considered as a smooth function of the surface normal, angle of the incident light ray and the emitted (reflected) light ray. This allows to determine the shape of the object in case the reflectivity function for a single viewpoint and the position of the light-source(s) is known. This function can be derived based on the properties of the object's surface and the light source location. The observed radiance under changing, calibrated illumination is for this purpose used to determine the surface bidirectional reflectance distribution function (BRDF) and surface normal. This procedure is in literature denoted as 'photometric calibration'. The special case where the data consists of a single image, known as 'shape from shading', was studied extensively by Horn (1970).

The reflectivity and the gradient of the surface can be related by a nonlinear first-order partial differential equation in two unknowns. These consist of the first partial derivatives of z (height of the object above the xy-plane) with respect to x and y, denoted p and q respectively:

$$p = \frac{\partial z}{\partial x} \quad q = \frac{\partial z}{\partial y}$$

Several approaches have been developed to compute a solution for the unknown height z but will not be further elaborated in this summary.

Traditionally, shape from shading algorithms assume a Lambertian reflectance model, for which the reflective properties are isotropic and the shading is independent of the viewing angle. Because water is highly specular, the technique needs further improvement as mentioned by Horn (1970). Several challenges have to be overcome:

- Multiple light sources are needed.
  A point light source will not be sufficient, as only a part of the surface will be highlighted, i.e. where the viewing angle and the angle of the light ray that is emitted are equal. Other parts of the surface will however not be lit, as the incident light ray will not reflect towards the camera.

- Splashing of water impedes the reconstruction.
  Water surfaces do normally not show occlusions or discontinuities which facilitates surface reconstruction. However, in case of splashing a high degree of non-linearity exists in case the surface slope is determined based on irradiance. This makes the method mentioned above less appealing.

- Small reflection of light at grazing angles.
  Because the Fresnel coefficients govern reflectivity, it is important to note that at grazing angles very little reflection will occur. A very large light source will therefore be required to capture the water surface at these small angles.

Goldman et al. (2010) consider the surface BRDF as a linear combination of basis BRDFs in which the unknown coefficients are optimized. However, the combination of photometric stereo with multi-view stereo approaches makes a BRDF-invariant reconstruction of surfaces possible. Based on this principle, several approaches haven been developed.

A first technique consists of using natural lighting conditions in combination with traditional stereo image matching (Shemdin (1990)). This technique is however not advised, as the resolution of the reconstruction appears to be insufficient for the determination of small wavelength-waves. Other problems are the observed correspondence error due to specular bias between binocular views and the fact that several specular artifacts complicate the reconstruction of small amplitude-waves.

A second technique is based on specular highlight falloff to compute the shape of the water surface. The surface slope at various points on the surface is in that case determined. Under certain assumptions mentioned by Schultz (1994), these can be used to determine the surface orientation based on the calculated and observed irradiance. This technique allows to determine the surface shape accurately in controlled circumstances.

Zickler et al. (2003) use on the other hand Binocular Helmholtz stereopsis, which is based on the principle that viewing rays and incident light rays can be switched without altering the surface reflectance. They use a reciprocating camera and light source to ensure that the pixel intensities in two images are only dependent on the surface shape and not on the object's reflectance properties. This allows to determine the shape of objects with arbitrary reflectance properties and in textureless regions.

Treuille et al. (2004) propose a technique which avoids the need of a reciprocity camera and light setup. They rely on the observation of the target object along with a reference object with a known geometry, having the same reflectance properties and observed under the same illumination conditions. Their work is based on orientation-consistency, which states that surface points with the same surface normal and material properties have the same radiance. Additionally, their model proves robust for self-shadowing of objects in case the lighting itself is calibrated. On the other hand, assumptions such as orthographic projection and distant lighting are made which decreases the general applicability and accuracy of this method.

Davis et al. (2005) combine photometric information with multi-view stereo by varying the incident radiance at every scene point but keeping the direction of the incident light constant. The varying illumination intensity can be used to find stereo matches without the need for photometric calibration.

Li et al. (2011) combine shape from shading with water incompressibility to develop a method in which the conservation of mass is used to determine the optical flow. The principle of optical flow will be explained more in detail in Chapter 8. The suggested technique uses shape from shading to acquire a prior (initial estimate) for the water surface, after which the horizontal and vertical velocities can be determined based on the mass conservation constraint. By using a set of Euler-Lagrangian equations, an objective energy function is minimized. This function is a weighted combination of intensity-conservation, mass-conservation and a smoothness constraint and results in the horizontal velocities u and v at every location (x,y) for a specific time instance t. Based on the derived horizontal velocities, the final vertical velocity (w) can be calculated and the change of the water surface height can be re-estimated. This allows to reconstruct a height field h(x,y,t) which represents the water surface at time t (Figure 3.5). It can be concluded that this method looks very promising. There are however several limitations to the proposed method.

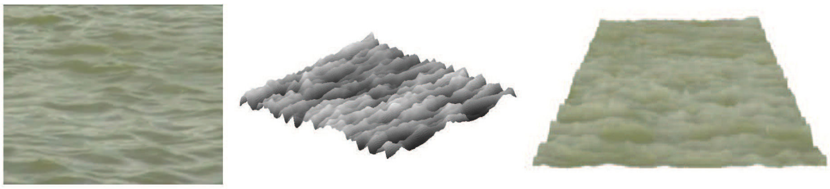


Figure 3.5: Li et al. (2011) uses a single input video to compute a height field of the water surface (in realtime). From Li et al. (2011).

First, it strongly depends on the surface prior acquired from shape from shading. Li et al. (2011) show that shape from shading works consistently well over a wide range of water surface shapes that have opaque and Lambertian properties. However, failure modes can appear in case this technique is used for water (transparent and highly specular) for which refraction and reflection distort the reconstructed surface. The proposed technique uses a basic low-pass filter to remove the extreme bright or dark pixels in the image. This proves insufficient for some applications but could be improved by using better specular/shadow removal methods. Additionally, the height field representation works efficiently well for calm water surfaces but fails to describe complex scenes such as splashing and breaking waves. A more sophisticated fluid representation is needed in those cases to handle the topological changes.

3.2.2.3 Shape from polarization

The shape of a mirror-like object can also be obtained from polarization, i.e. the phenomenon that light waves sometimes vibrate in a single plane. Because reflected light becomes partially polarized in the direction of the surface normal, the phase image of the object encodes the orientation of the reflection plane (plane defined by surface normal and incident light ray). Once the orientation of the reflection plane is known, several techniques exist to determine the surface normal and location of the surface.

Rahmann and Canterakis (2001) propose to use multiple views of the surface, after which the reconstruction problem is solved with an optimization scheme. In their paper, they show that the presented method is independent of the illumination of the surface and the optimization algorithm can be considered as converging and noise resistant.

Secondly, Miyazaki et al. (2004) solve the ambiguity problem by tilting the object over a small angle. By calculating the difference in polarization degree between two sets of data, the correct surface normal can be obtained. The use of polarization-based approaches becomes however difficult in case transmission instead of specular reflection dominates the image formation, making this approach less suitable for water reconstruction.

3.2.2.4 Shape from distortion

The shape of the mirror surface will deform the reflected view from the scene. As such, the distortion of a projected pattern can be used to reconstruct the surface shape. The adopted pattern is in some cases projected as a radiance map in case distant illumination is assumed. Others propose to place it close to the object, which results in depth-normal ambiguity.

Oren and Nayar (1997) use the apparent motion of features captured by a moving camera. They make a distinction between two types of features. A first type is denoted as 'real scene-features', which are physical points on the object and remain stationary independent of the camera position. 'Virtual features' are on the other hand specular reflections of features that are not located on the object and travel on the surface in case the camera is moved. For the 2D case in which the camera motion and surface profile are coplanar, the surface can be reconstructed by tracking two virtual features. For a 3D surface, an algorithm is developed that allows to reconstruct a 3D curve on its surface using a single virtual feature on the occluding boundary of the object.

A different approach uses the tangential information of the specular surface on the intersection of distorted lines. The curvatures of these intersecting lines are used to determine the surface normal, which then allows to compute the surface location of these intersection points. To that end, the local geometry of a planar world pattern is combined with the observed reflections on the mirroring surface. This information can then be used to locally reconstruct the surface itself, but requires a priori information about the mirroring surface or an approximation of its distance.

This principle was first used by Savarese and Perona (2001) and later improved by Savarese et al. (2005), who determine the surface depth and higher order properties at these sparse set of intersection points. This allows to extract up to third-order information from a single viewpoint in case the calibrated planar scene is known and at least six reflected scene points are captured.

Based on their results, Rozenfeld et al. (2011) present a system in which several images with parallel stripes are displayed on the mirroring surface at different angles. The camera captures these distorted stripes on the mirroring surface and based on the difference between the distorted stripes and projected stripes, a 1D-homography can be estimated. Initially correspondence is only extracted for a sparse set of points and by minimization of a self-defined error measure, the depth and first-order local shape is obtained. Dense surface recovery is subsequently performed using constrained interpolation. In their paper, they show that the developed cost function usually results in a single minimum and allows computations that only depend on local information.

A third approach uses a certain pattern of features that is localized in the reflected image after which multi-view voxel carving is used in combination with a normal consistency check (Bonfort and Sturm (2003)). The three-dimensional space around the surface is discretized in voxels, which can be considered as the three-dimensional equivalent of pixels (i.e. boxes in a 3D-regular grid). For each of these voxels, a surface normal is computed that corresponds with the appearance of the reflected image in case the surface passes through that voxel. This is done for every viewpoint, which results in series of normal sets. The voxels associated with surface normals which are not consistent between the different viewpoints are then discarded. The collection of voxels and surface normals can then be be used to determine the approximate shape of the surface in 3D space. Similarly as stated earlier, this method requires some knowledge about the surface shape and position and again multiple images of the object under different viewpoints are needed.

Nehab et al. (2008) also use the consistency of normal directions as an alternative cost function in the standard stereo algorithm. Because the matching is not robust, mismatches have to be removed by an aggregation method based on anisotropic diffusion (Scharstein and Szeliski (1998)). This cost computation is performed over a window around the point of interest, which is in this paper expressed as a 2D or 3D convolution. Additionally, the authors discuss the different ambiguities that can occur during reconstruction.

Tarini et al. (2003) suggest a technique in which a single camera and a monitor displaying several images on the mirroring surface are adopted. An improved environment matte for the mirroring object can then be constructed. A dense, sub-pixel accurate correspondence between the observed image and the displayed image by the monitor is subsequently obtained using the uniformity of some photometric properties. These correspondences result in restrictions on the depths and surface normals, which are finally used together with smoothness assumptions to determine the surface shape in a global optimization procedure. The disadvantage is that this method and its accuracy are highly dependent on the photometric calibration. Additionally the technique remains limited to static objects because several images of the object are necessary to construct the environment matte.

3.2.2.5 Shape from specularity

These approaches are based on the detection of discrete surface highlights caused by specular reflection at particular points of the reconstructed object. Healey and Binford (1988) compute the local orientation and principal curvatures/directions of a specular surface by examining the image intensities around a specularity. Based on the statistics of this radiance fall-off, a complete local characterization of the surface up to second order can be made.

Other methods rely on the movement of these specular highlights in case the imaging sensor is moved over a known distance (Zisserman et al. (1989)) or in case of a moving object observed by a static camera (Zheng et al. (1996); Zheng et al. (1997);Zheng and Murata (1998);Zheng and Murata (2000)). Based on the apparent motion of these highlights on the object surface, the surface shape can be derived.

Another method proposed by Sanderson et al. (1988) uses an array of light sources that are positioned far from the object. By alternately activating these point light sources, the normal field of the surface can be reconstructed by observing the corresponding highlights.

A final approach makes use of the information contained in the polarization of specular highlights. Because light becomes partially polarized due to reflection, polarization will be minimal in the plane containing the incident light ray, viewing ray and surface normal. Saito et al. (2001) use a rotating linear polarizer in front of the camera to determine the polarization state of the light measured by the imaging sensor. This allows to determine the angle between the surface normal and the incident light ray and as such calculate the plane of reflection.

Several variations of these methods have been developed to obtain more detailed surface information from specularity measurements.

Wang and Dana (2006) move a parabolic mirror across the object's surface, and change the incident light directions by shifting a movable aperture. This allows to determine the surface normals and a spatially varying BRDF.

Other approaches are based on static viewpoint sampling with a moving point light source which results in different incident illumination directions. By using a predefined threshold (Chen et al. (2006)) or coded illumination (Francken et al. (2008b); Francken et al. (2008a)), specular highlights in the images are identified which allows to determine the normal field. The final surface shape can then be obtained by integration. Because these approaches can only be used to reconstruct approximately planar surfaces, extensions in which gradient illumination is used have been developed to reconstruct more complex shapes. By comparison of the images under constant and gradient illumination (produced by spherically distributed LED's (Ma et al. (2007)) or an LCD screen (Francken et al. (2008c)), the surface normals of arbitrary objects can be determined. Additionally, the authors propose a method to separate the diffuse and specular reflection. This allows to recover high resolution surface models.

Due to their high accuracy, these methods seem very promising to be used for the reconstruction of specular static objects, although capturing the dynamic shape of water surfaces will require a different approach.

3.2.2.6 Shape from specular flow

A different approach tracks a (semi-)dense set of feature points on the specular surface from a distant environment map. The concept of specular flow is relatively new (Roth and Black (2006)). They define specular flow as the image motion due to a moving camera around a reflective object. Specular flow represents the dense (or semi-dense) flow field (vector field) on these specular surfaces. They additionally expand their model to surfaces which are considered to be a mixture of diffuse and specular regions.

Their algorithm is based on a known camera position and a derived vector field describing the optical flow between image pairs (Figure 3.6). The authors incorporate an initially unknown material distribution (specular or diffuse) and segment the flow field in diffuse optical flow and specular flow using a maximization algorithm. Regions moving due to diffuse and specular flow can as such be distinguished from each other. Based on the assumption of a distant illumination field, the parametric models of diffuse and specular flow can then be used for surface reconstruction.

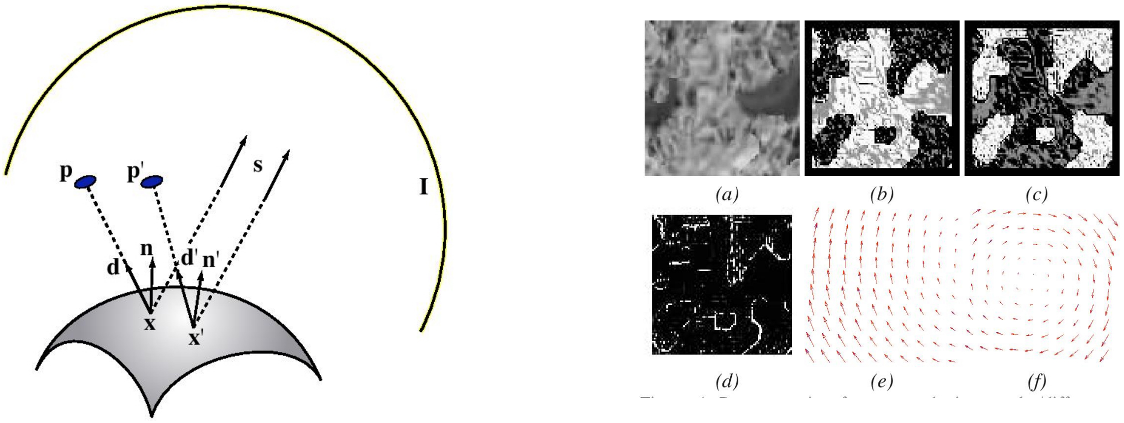


Figure 3.6: Roth and Black (2006) use distant illumination of a specular surface, for which the diffuse (e) and specular flow (f) of feature points on the surface are separated. These are finally used to derive the material properties and shape of the surface. From Roth and Black (2006).

Several improvements to this technique have been made. Assuming orthographic projection and far-field illumination, Adato et al. (2007) use a method that relates the observed specular flow velocities through a pair of coupled non-linear partial differential equations. These can be solved in case the angular rotation around the optical axis of the camera remains constant. Finally, Vasilyev et al. (2008) propose a similar approach to determine the shape from specular flow induced by three rotations around arbitrary axes.

However, it must be mentioned that all of these methods require multiple images of the surface from different camera positions which makes their applicability limited to static objects. As water is highly dynamic, specular flow can therefore not be used in this thesis.

3.2.2.7 Direct ray measurements

In case of near-field illumination, depth-normal ambiguity arises. In contrast with far-field illumination, the incident angle at each scene point is different. Direct ray measurements therefore require a second 3D point on the incident light ray which fixes its direction.

Although Kutulakos and Steger (2008) assume that exactly one reflection occurs along the ray, they present a framework that does not impose a priori assumptions on the shape of the surface or the nature of the media (e.g. opaque or transparent). To that end, they formulate the reconstruction problem as a geometric constraint satisfaction problem by using the individual light paths and use a function that maps each image point to two known reference points along its light path. These two points are determined by placing a known, planar pattern at several positions w.r.t. the object. The refracted view of this pattern captured by the camera is then related to the corresponding position of the actual plane in the scene. By using the knowledge about the reflected and incident light ray, the surface normal and the corresponding surface point can be determined separately which allows an even more accurate measurement for planar surfaces. Bonfort et al. (2006) extend this approach to arbitrary surface shapes.

3.2.2.8 Laser rangefinders

Laser rangefinders allow surface reconstruction based on the same principle of triangulation. This involves projecting laser light onto the object surface, and measuring the reflected light at a known receiver. Solving this triangulation problem results in the reconstruction of the object surface. Although these methods are usually developed for Lambertian surfaces, several laser rangefinders have recently been developed to cope with specular surfaces.

Baba et al. (2001) restrict the incident light ray by using a special light device, i.e. a shield mask in front of the image sensor, to only receive light from the expected angle. In further improvements of the proposed technique (Baba et al. (2004)), they equip the laser rangefinder with an image sensor that makes it possible to detect both the position of light ray and the incident angle on the sensor. After determining the incident angle on the sensor, the orientation of the object can mathematically be estimated.

Additionally, this technique allows to determine not only the three-dimensional shape but also the surface reflectance properties of the object. Experiments showed that the improved rangefinder is able to detect the object's shape faster than the original laser rangefinder and can be used to reconstruct specular, Lambertian (diffuse) or hybrid objects. Despite these improvements, the proposed method remains slow and also requires high technological equipment such as laser rangefinders and sensors. A direct application to scan highly dynamic water surfaces is therefore not possible.

3.2.3 Reconstruction based on transparency

Because water is (partially) transparent for light rays, techniques applicable to transparent objects can be adopted to determine the shape of water surfaces. The object (water) is in this case not considered as an obstacle but rather as a transition to a medium with different optical properties than the surrounding environment. A short list of possible methods is given hereafter.

3.2.3.1 Computerized Tomography

In case the wavelength of the light used for illumination is sufficiently high and the refractive index of the object and the medium surrounding the object are approximately equal, the incident rays do not refract at the objects surface. The technique is well known in medical imaging in which X-ray images are taken from different view-angles which results in cross-sectional (tomographic) images of specific areas of the scanned object. To that end, Kak and Slaney (1988) propose a technique, called the 'backpropagation algorithm', to perform the compilation necessary for three-dimensional reconstruction. The disadvantage of this technique is that simultaneous images of the object are required and it is difficult to avoid imaging the capturing equipment at the same time. Additionally, the technique can only be used with X-rays because it fails in combination with direct optical images and ultrasound due to surface refraction.

3.2.3.2 Multi-view reconstruction

Multi-view reconstruction with transparency can be considered as a modified version of voxel carving. For each ray through each pixel, a weight is assigned to the voxels along that ray which indicates how much a particular voxel contributes to the pixel color. These weights can subsequently be used to find the transparency values. The most consistent set of voxels and corresponding weights is finally found by using several views of the object. This approach is also called the Responsibility Weighted 3D Volume Reconstruction (Roxel) algorithm. De Bonet and Viola (1999) mention however that due to the fact that transparency is equated with uncertainty, surface edges become blurry in case the precise location of these edges is uncertain. The Roxel algorithm additionally suffers from the same problems as CT-techniques in combination with water: it does not consider refraction and requires simultaneously images from multiple views.

Tsin et al. (2003) propose a method based on stereo reconstruction but taking into account the presence of reflections and translucency. Image formation is modeled as the additive superposition of layers at different depths. They develop a nested plane sweep framework, which allows to determine depth components in a systematic way. Using a component-color-independent matching error, the depth hypotheses are then optimized. This results in an accurate estimation of the stereo correspondences for complex scenes consisting of multiple layers. Additionally, they present an iterative color update algorithm to extract the correct colors of the different layers.

3.2.3.3 Polarization analysis and separation of local and global light transport

The difficulty for traditional structural light 3D scanning techniques applied to transparent objects is the occurrence of subsurface scattering just below the object surface. This results in a drastic reduction of the signal-to noise ratio and shifts the intensity peak beneath the surface, to a point which does not coincide with the point of incidence. Several methods have been developed to overcome this problem.

A first approach is to use specular highlights as mentioned previously. Because specular reflection is not influenced by sub-surface light transport, specular highlights always appear at the same position independent of the global light transport within the object. (Chen et al. (2006))

Chen et al. (2007) propose a method in which a combination of polarization and phase-shifting is employed. By using polarization-difference imaging, subsurface scattering is filtered because multiple scattering randomizes the polarization direction. In contrast, the surface reflection partially keeps the polarization direction. By using two orthogonal polarized images, they rely on this idea to remove specular highlights. In the same paper, they use a phase-shifting technique on high frequency light patterns which allows the separation of specular highlights and diffuse components.

In further research, they suggest to improve the technique by combining low-frequency and high-frequency light patterns (Chen et al. (2008)). This allows to separate the local direct illumination effects and the light contribution due to global light transport. The improved technique proves to be more robust and eliminates the need for polarization filters compared with the method of Chen et al. (2007). On the other hand, the suggested method only finds an approximate solution, for which the errors are rapidly accumulated in case of several internal reflections.

Morris (2004) suggests that multi-view or a binocular stereo shape from polarization approach could theoretically be used to overcome this problem. However, this is not yet attempted in practice and simpler methods are used to reconstruct the water surface.

3.2.4 Reconstruction based on refraction

The surface description of refractive objects with inhomogeneous material properties is very complex and the image formation for objects with inclusions and cracks is very difficult. In most cases however, refraction can be considered to occur at a single surface which forms the boundary of the observed object and its environment. Almost all methods assume the refractive object therefore homogeneous and most of them are based on a priori known refractive index. Although these assumptions made somewhat limit the general applicability of these methods, the determination of a water surface by using its refraction properties offers two main advantages compared to reflectance-based techniques as is mentioned by Morris (2004):

- Refraction nonlinearities are much lower than those for reflection, which allows a smaller light source or pattern to be used.
- Most refraction techniques light the surface from below and as such, no specular artifacts are present that would occur due to specular reflection.

3.2.4.1 Shape from distortion

These approaches are based on the same principle as for reflective objects. However, an additional complexity arises due to the fact that the light path is not only dependent on the surface normal but also on the refractive index of the media at both sides of the surface.

Murase (1990) applies the concept of shape from refractive distortion to water surfaces. Using a unknown pattern that is placed on the bottom of the water tank, a sequence of images of the water surface is captured. The distortion (due to refraction at the air-water interface) of an initially unknown pattern is then used to derive the shape of the water surface. The reconstruction algorithm is composed of four stages (Murase (1992)):

1. Using the distortion of the image of the pattern, optical flow is computed for the entire pattern.
2. The average of these optical flow displacements is taken as the location of the undistorted background pattern, meaning only refracted at a planar interface.
3. The distortion vectors w.r.t. the mean location can then be calculated for every frame. Based on these optical characteristics, the surface gradient and normal can subsequently be computed for every point.
4. The entire surface is determined by integration of the surface normals.

The proposed technique has however some disadvantages:

- A distant, orthographic and one-view set up with parallel light rays is assumed during his derivations (see Figure 3.7). For camera setups at which the camera is placed at small heights above the water surface, this simplification is not longer valid.
- The method can only be used for low amplitude waves because large amplitudes would result in a too large distortion of the projected pattern.
- The surface can only be reconstructed up to some unknown scale factor. The scale of the surface is in that case influenced by the refractive index and the distance between the water surface and the bottom of the tank.

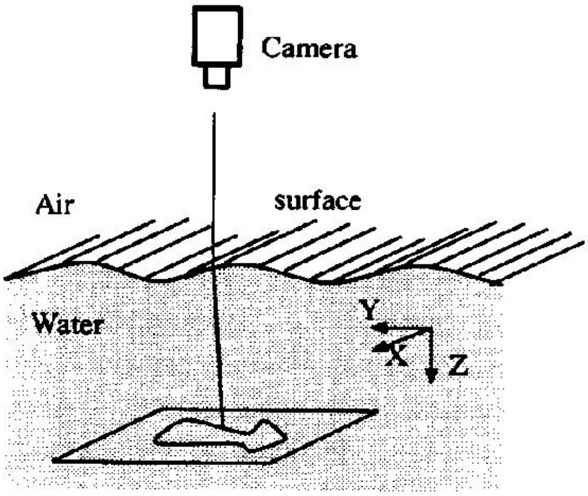


Figure 3.7: Murase (1992) uses an orthographic camera view to detect an initially unknown distorted pattern below the surface. Using the average position as approximation of the undistorted position the pattern, the surface shape can be reconstructed. From Murase (1992).

Morris (2004) derives a method to reconstruct a time-varying water surface in which the same principle of refractive distortion is used. The imaging and statistical assumptions made by Murase (1992) are however avoided by combining refractive distortion with a stereo setup and the use of a known background pattern, making the methodology more physically consistent (Morris and Kutulakos (2005)). The algorithm is a special case study of a general analysis on the reconstruction of light paths (direct ray measurements) conducted by Kutulakos and Steger (2008). The method involves the determination of the surface normals based on the observed feature points on the water surface of which the light rays are refracted at the air-water interface. Based on the known, physical location of these feature points on the bottom of the tank, the geometric problem can be solved in case a certain depth along the viewing ray is assumed. This is done for two stereo cameras, which results in two sets of surface normals and corresponding water surface locations for every time frame. The hypothesized depth is then verified by using an error metric which expresses the dissimilarity between corresponding surface normals at particular locations, each determined with a different camera.

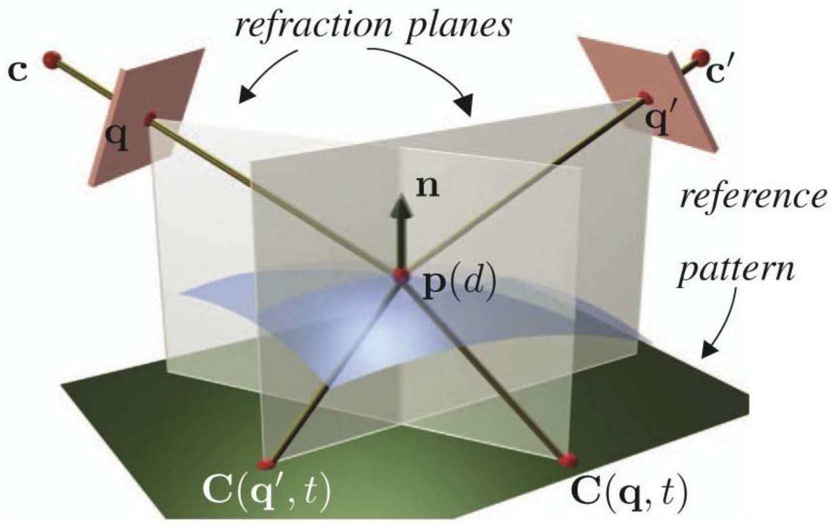


Figure 3.8: Sparse multi-view stereo technique based on shape from distortion. From Morris and Kutulakos (2005).

For this purpose, two different error metrics are suggested: the normal collinearity metric and the disparity difference metric. Based on numerical simulations, the disparity metric proves to give more accurate results for small water depths. An accurate reconstruction of the water surface is finally obtained by a minimization procedure to find the optimal depth for every feature point that minimizes a normal matching cost function. The feature points are tracked using a Lucas-Kanade tracking algorithm to obtain the time-dependent variation of the water surface. This extended stereo setup allows to determine an initially unknown refractive index (Morris and Kutulakos (2005)) and does not rely on an average surface shape as the water depths for the different feature points are optimized separately. The method proves to be robust for the loss of feature points and can also be used in case the tank is partially empty.

The same principle has also been applied to other translucent objects than water. Hata et al. (1996) use multi-stripe lighting that is projected on objects with one planar surface that rest on a diffuse base. By detection of the distorted patterns, the shape of the glass and drop-like objects can be obtained. The light path is in that case refracted twice (entering and leaving the glass object), which makes an analytic solution of the problem difficult. The authors therefore develop genetic algorithms to solve this issue. In their paper, they show that transparent paste drops can accurately and efficiently be reconstructed.

A method proposed by Ben-Ezra and Nayar (2003) is based on feature tracking in which feature points are imaged through a transparent object as shown in Figure 3.9. They implement a model-based approach in which the shape and pose of transparent object is determined by minimizing a certain object function using a steepest descent method. In this way, they are able to reconstruct not only one surface but the complete surface shape of the objects. Their study remains however limited to objects that are parameterized by a single shape parameter ϵ, such as super-quadrics or spheres. They characterize the orientation and position of the object by the rotation matrix R and translation vector T. The proposed method shows promising results to determine the shape of transparent, refractive media, although its applicability remains limited to low-dimensional shapes. Additionally, multiple images with a moving camera are needed to estimate the position of the static object.

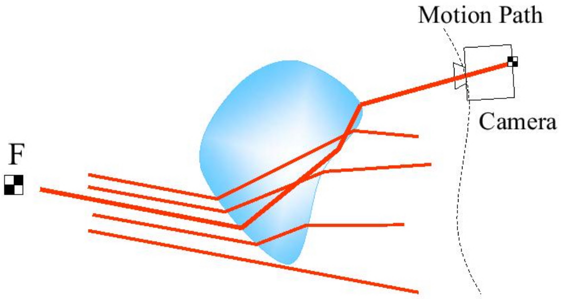


Figure 3.9: Ben-Ezra and Nayar (2003) use parallel incident light rays from a distant feature F that are scattered due to the refraction within the transparent object. The light rays are detected by a moving camera and given the expected shape, the position and shape parameter of the object can be determined. From Ben-Ezra and Nayar (2003).

For methods based on tracking feature points through images of refracted scenes, localization of these feature points proves sometimes difficult. Due to refraction, the pattern can be severely magnified or reduced in size and the distortion can even be so severe that feature matching is not longer possible. Moreover, the pattern might become difficult to distinguish for media that are not completely transparent. Absorption of light might in that case cause variations in the intensity of the tracked feature points.

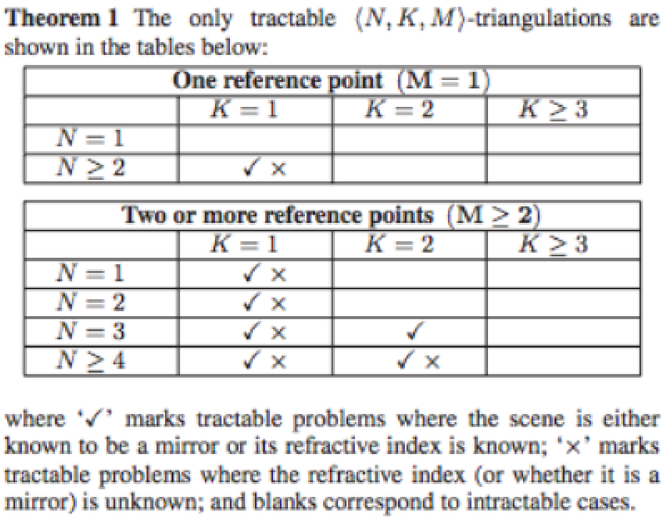
This problem is studied by Agarwal et al. (2004), who generalize the optical flow equation derived by Murase (1992) using a particular choice of optical kernel function. This allows to account for the warping and attenuation caused by a refractive object. Using their algorithm, they are able to solve the warping and attenuation problem in case of refractive objects to obtain a plane-to-plane mapping. For this purpose, a known or unknown movement of a planar background is tracked and the 3D distortion of this plane can then be determined. It must be noted that the proposed method is based on some restricting assumptions. The specular, Fresnel and total internal reflection are not considered and the background behind the object is assumed to be planar.

3.2.4.2 Direct ray measurements

The concept of ray tracing can also be used for refractive media, in which the rays are measured after having passed through the refractive object. A first technique uses a known, planar calibration pattern at several positions w.r.t. the object to measure the incident light ray on the surface as described in Section 3.2.2.7.

Kutulakos and Steger (2008) investigate the application of direct ray measurements extensively for a large range of situations. In their paper, they distinguish for both refractive and specular surfaces different types of light-path triangulations based on 3 parameters: $\langle N,K,M \rangle$. N represents the number of view points necessary for reconstruction, K the number of vertices of which the light rays are composed and M the number of known reference points per image point. By a case-by-case analysis in which these three parameters are varied, they obtain an enumeration of tractable triangulation problems. An overview of their results is given in Figure 3.10.

The $\langle 1,1,2 \rangle$ -triangulation problem was already explained in Section 3.2.2.7. The method proposed by Morris and Kutulakos (2005) is an application of the $\langle 2,1,1 \rangle$ -triangulation problem. A third class, the $\langle 3,2,2 \rangle$ -triangulation, is also detailed described in this paper and can be used to reconstruct glass objects.



Theorem 1 The only tractable $\langle N, K, M\rangle$-triangulations are shown in the tables below:

One reference point (M = 1)			
	$K = 1$	$K = 2$	$K \geq 3$
$N = 1$			
$N \geq 2$	✓ ×		

Two or more reference points (M ≥ 2)			
	$K = 1$	$K = 2$	$K \geq 3$
$N = 1$	✓ ×		
$N = 2$	✓ ×		
$N = 3$	✓ ×	✓	
$N \geq 4$	✓ ×	✓ ×	

where '✓' marks tractable problems where the scene is either known to be a mirror or its refractive index is known; '×' marks tractable problems where the refractive index (or whether it is a mirror) is unknown; and blanks correspond to intractable cases.

Figure 3.10: Tractable triangulation problems in function of $\langle N,K,M\rangle$. From Kutulakos and Steger (2008).

The authors show that a pixel-wise independent reconstruction procedure is not possible for more than two, specular or refractive, intersections of the light ray. Finally, it is proven that more than 2 known 3D points per light ray do not contribute to additional information for the reconstruction problem.

Another approach described by Atcheson et al. (2008) approximates the refracted light rays based on optical flow data in which it is assumed that the object (gas flow in their case) is small compared to the distance between the object and the background pattern. In that case, the small deflections of a known background pattern, shown in Figure 3.11, can be computed with optical flow methods. In their paper, they reconstruct the variations in the refractive index of gases due to temperature changes by studying the small changes in direction of these rays. A linear system which describes these differential changes is finally solved using a least-square minimization method, which results in a volumetric measurement of the refractive index.

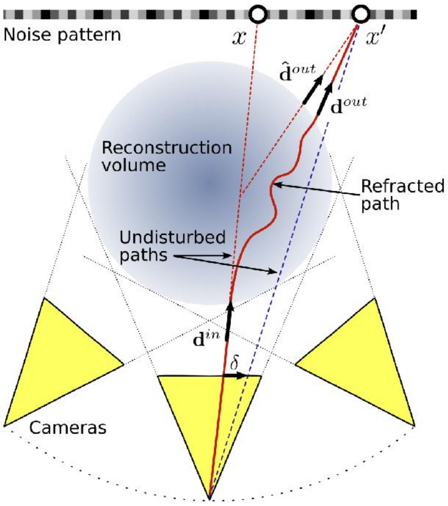


Figure 3.11: Atcheson et al. (2008) first detect the frequency of a dot pattern without the object. Another image is then taken with the object and the deflection of the light ray in the image plane is determined with optical flow. From Atcheson et al. (2008).

3.2.4.3 Laser rangefinders

Special laser rangefinders have been developed to reconstruct water surface based on the refraction of the laser ray. Wu and Meadows. (1990) project a laser ray through the water and measure the ray's deflection due to refraction. Subsequently, the surface normal is found by the detection of the refracted ray with an sensor screen positioned at the opposite side of the surface. These surface normals can then be used to calculate the wave slopes in two dimensions. The water surface intersection point is finally determined by using an iterative search that accounts for the detected ray on the screen.

3.2.4.4 Inverse ray-tracing

In case of inverse ray-tracing, input data is compared with synthetically generated images. Instead of matching distinctive features in the scene with image points (pixels) in the image, they search for an image formation model based on volumetric ray tracing, i.e. the 3D geometry and material properties of the reconstructed object, that best explain the observations. The input data set is then compared with the computed 2D data set (based on the current estimate of the surface shape) and the dissimilarity between both is minimized. Different approaches can be used to obtain this input data, in which usually one of the previously mentioned techniques is used.

A first approach uses the effect of fluorescence or chemiluminescence. Ihrke et al. (2005) dissolve the chemical Fluorescein in the water, after which the water is illuminated by UV Light. A measurement of the thickness of the water can then be obtained from the amplitude of the emitted light in case the self-emission is assumed homogeneous throughout the water column. The water surfaces is then matched by a video frame of synthetically produced images which is generated by a multi-view setup based on a constant self-emissivity model. The visual hull of the water surface is finally calculated by utilizing a weighted minimal surface using the thickness measurements as constraints.

Goldlucke et al. (2007) suggest a similar approach but use chemiluminescence that produces light by a chemical process instead of the reflection of UV.

Wang et al. (2009) dye water with white paint and light patterns are projected onto the water surface. This allows to use standard stereo reconstruction to reconstruct the shape of the water surface. A dense reconstruction algorithm is adopted to determine a depth field which is refined afterwards using physically-based constraints. They suggest that their physically-guided model allows to automatically fill in missing regions, remove outliers, and refine the geometric shape so that the final 3D model is consistent with both the input video data and the laws of physics. This method shows very accurate reconstruction of surface details.

3.2.4.5 Thermographic surface reconstruction

Because the infrared spectrum is not subjected to refraction, using an IR-laser instead of visible light is a promising alternative to be used in surface reconstruction. Eren et al. (2009) develop a method, named 'scanning from heating', to reconstruct the shape of glass objects that are heated to make them detectable with an infrared sensitive sensor. The resolution of this technique remains however limited since the wavelength of incident light is much larger and therefore much more difficult to focus in comparison with visible light.

Hilsenstein (2005) proposes a method based on stereo reconstruction, in which water waves are reconstructed from thermographic image sequences acquired from a pair of infrared cameras.

Infrared stereo reduces the problem associated with transparency, specular reflection and lack of texture that occurs for visible wavelengths. These techniques all require sophisticated and expensive equipment and complex experimental setups which make them less attractive.

3.2.4.6 Shape from refractive irradiance

This approach is based on the idea of relating image intensity or color with a certain surface shape (and corresponding slope). Several techniques have been developed (Daida et al. (1995), Keller and Gotwols (1983), Zhang and Cox (1994), although most of these are based on the same basic principle.

By using a specialized lens, a screen of light is collimated in order to make parallel light ray columns with a certain intensity or color. If these light rays are refracted by the water surface, only a part of these light ray columns will reach the camera, which is positioned far enough from the water surface. The color or intensity detected by the camera, can therefore be associated with a particular slope (and normal) of the surface. The entire surface is subsequently reconstructed by integrating these surface normals.

Morris (2004) mentions that this technique has however several disadvantages:

- An infinitely distant camera is assumed. Parallel incoming rays are however only a rough approximation and a model error is inherently incorporated in this technique.
- Additional errors will occur due to the collimating lens because the light rays may not be perfectly collimated.
- The light attenuation by the water surface will be different for different parts of the image as the underwater path lengths will slightly be different. Therefore, the measured intensities will not only be the result of the slope of the water surface and the measured intensities should be corrected to find the actual surface elevation.

3.2.4.7 Reflection-based reconstruction for refractive media

We finally present an algorithm suggested by Morris and Kutulakos (2007) to determine the exterior surface of refractive and transparent objects with an inhomogeneous interior (e.g. multiple interfaces or reflective interiors). Although the application to self-induced sloshing does not suffer from these problems, the proposed method can still be useful to investigate other fluid-dynamic phenomena. Additionally, it could be used in case the research on self-induced sloshing is expanded to two-liquid media such as fresh-salt water or fresh-muddy water.

The method involves acquiring high-resolution images of a static object from one or more viewpoints, while a light source is moved in a regular 2D (or 3D) pattern. This produces a 2D (or 3D) set of measurements per pixel, the so-called pixel's scatter trace, which can be considered as the trajectory of the light before interacting with the object and arriving at the given pixel. The reflectance measurements, which are the result of both direct reflection and global light transport effects, are then split in its two components by exploiting the physical properties of light transport. Additional constraints are then set in order to formulate a 'scatter-trace stereo algorithm' to compute the shape of the exterior surface (depth and surface normals). The major disadvantage of this method is that multiple images of a static object are necessary to compute the pixel's scatter trace. Because water is highly dynamic, a modification of this approach would be necessary to be applicable for water surfaces.

3.3 Conclusion and choice of the most suitable technique

Based on the overview given in this chapter, a choice was made on which reconstruction method our developed method will be based. As explained earlier, every type of reconstruction method uses a different optical property of water to determine the shape of the water surface. Reflection of light is the basis of the techniques described in Section 3.2.2, for which several possible solutions have been developed to handle specular highlights. Reflectance of light on the air-water interface is however governed by non-linear Fresnel reflection. The amount of light that is reflected is in that case largely dependent on the viewing angle w.r.t. the surface normal. Although the amount of reflected light is large in case these intersect the water surface at a sharp angle, light that encounters the water-air interface approximately perpendicular is almost completely transmitted. This makes such methods less robust and unsuitable in case water surfaces with large variations in surface slope need to be reconstructed. Methods that use transparency of light (Section 3.2.3) usually require multiple images of the same water surface. Although solutions have been sought to accommodate this issue, other approaches seem therefore more appropriate.

Laser rangefinders are an example of sensor-based techniques, in which the reflected or refracted rays are subsequently detected at the same or opposite side of the air-water interface respectively. The available methods seem however to slow to capture highly dynamic water surfaces and require specialized and expensive measurement equipment. Image-based techniques combined with refraction of light are therefore the most obvious choice. Optical or image-based techniques have the advantage that the required imaging equipment is easily available. Additionally, the accuracy of these methods can in most cases simply be improved by adopting higher-resolution cameras and better lighting setups.

Shape from refractive distortion and direct ray measurements seem the most promising approaches compared to other image-based techniques. Most refractive-based methods require however a distant, orthographic camera or a collimating lens under water. The imaging assumptions related to this make an accurate reconstruction more difficult. Inverse ray-tracing seems also promising but requires the media (water) that is modeled to be altered to obtain an image formation model. This makes these methods more difficult to combine with other measurement techniques such as PIV or PTV.

The method proposed by Morris (2004) is based on the same basic principle but takes into account the perspective transformation during capturing of images. They suggest a sparse, multi-view stereo approach in which the distortion of a feature pattern due to refraction is used to derive the shape of the water surface. It allows to avoid some of the assumptions made in other techniques (e.g. the use of a distant orthographic camera view or statistical properties of the surface undulations) which significantly improves the accuracy. For this reason, the chosen technique is based on their work. Some adaptations are however made to improve the robustness w.r.t. feature localization in the images of the water surface.

We combine this refractive stereo technique with a low parameter model of the water surface, similarly as explained in Ben-Ezra and Nayar (2003). They however use only one shape parameter ξ to describe the shape of a reconstructed refractive and static object. This shape parameter is then optimized together with a rotation matrix R and translation vector T to estimate the pose of the object w.r.t. the camera position. We will use more extensive surface models to obtain a theoretical description of the entire water surface shape.

Compared to the original method of Morris (2004), this allows to reduce the computational time for a singe surface reconstruction and improves the robustness of the algorithm with respect to loss of feature points. Additionally, the spatial extent of the surface reconstruction can be improved in case only a limited amount of cameras is available. The theoretical framework, on which our surface reconstruction algorithm is based, will be explained in the next chapter.

Chapter 4

Theoretical framework of the reconstruction algorithm

4.1 Image-based reconstruction of the water surface

The presented method which is used to determine the water surface is based on the sparse multi-view reconstruction approach of Morris (2004). The basic principle in which 3D shape information is derived from refractive distortion was already applied by other authors. However, Morris (2004) does not not assume a distant, orthographic camera view of the surface which is in most methods adopted. Instead, they combine a traditional stereo approach with a shape from distortion approach. As mentioned by Morris (2004) and Morris and Kutulakos (2005), this makes their method more physically consistent and eliminates the need for an extra collimating lens under the water surface. Their technique is therefore more accurate and less difficult to implement in practical applications.

4.1.1 Light refraction at the air-water interface

It is known that light travels at different speeds in media with other densities. At the boundary of two layers, the light rays are refracted or bent. This change in direction between the incident and refracted light rays can be described by Snell's law:

$$r_1\, sin(\theta_i) = r_2\, sin(\theta_r) \tag{4.1}$$

where:

r_1	The refractive index of the first medium.
r_2	The refractive index of the second medium.
θ_i	The angle between the incident ray and the surface normal of this boundary.
θ_r	The angle between the refracted ray and the surface normal of this boundary.

These incident and refracted rays are located in the same plane, which in every situation also contains the surface normal ($\overrightarrow{\boldsymbol{n}}$). Refraction can therefore be considered as a two-dimensional problem as depicted in Figure 4.1. At the interface between water and air, this equation can be further simplified because the refractive index of air equals 1:

$$sin(\theta_i) = r_w\, sin\,(\theta_r) \tag{4.2}$$

where the refractive index of water r_w typically equals 1.33.

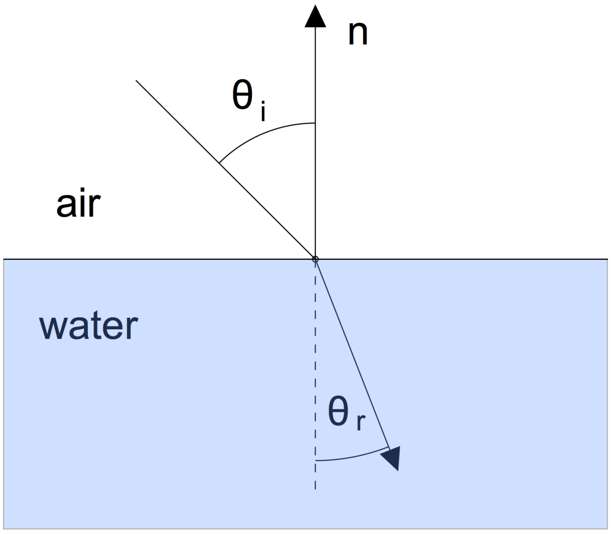


Figure 4.1: Schematic representation of Snell's law: the incident light rays is refracted at the air-water interface with a surface normal $\vec{\boldsymbol{n}}$.

Two important aspects of light refraction must be mentioned. Firstly, light rays that travel from water to air (in the opposite direction as defined earlier) will partly be reflected and partly refracted. The ratio of reflected to refracted light increases as the angle of incidence increases. In the limiting case, no light is refracted and total internal reflection occurs. The angle at which total internal reflection takes place is denoted as the critical angle θ_{cr}. For incident light rays that encounter a boundary between two media at the critical angle, the refracted rays are tangent to this boundary. To find θ_{cr}, the angle of refracted light θ_2 in (4.1) is set equal to 90° and thus $sin(\theta_2) = 1$. The resulting critical angle can therefore be calculated with $\theta_{cr} = sin^{-1}\left(\frac{r_2}{r_1}\right)$. For the water-air interface, this gives a critical angle equal to:

$$\theta_{cr} = sin^{-1}\left(\frac{1}{r_w}\right) \approx 48^\circ$$

Secondly, refraction of light rays is also dependent on the wavelength as red light has a higher refractive index in water than blue light. Also the temperature of the water has an effect on the refractive index. This particular feature has been studied extensively by Harvey et al. (1998), Schiebener et al. (1990) and Thormählen et al. (1985). They developed graphs and empirical formulas that describe the refractive index as a function of the wavelength, temperature and density of the water. Because we will use pure water in this thesis, the final parameter is only dependent on the water temperature. To account for the varying refractive index r, we split the images obtained by the cameras in their different color channels (RGB). The processing of the different color channels is done separately, each with a refractive index that corresponds with the wavelength of that particular part of the light spectrum and temperature of the water. The graphs and formulas provided by the previously mentioned authors are used to get a reliable prediction of the refractive index. Inaccuracies related to an incorrect estimation of the refractive index are in such a way prevented as much as possible.

4.1.2 Schematic overview of imaging setup

The principle of shape from refraction distortion is simple. A regular pattern of feature points **f** on a plain surface F is projected on the water surface. The local shape of the surface results in a deformed, projected pattern that is captured by the camera. Based on the deformation of the detected pattern compared to the actual pattern on plane F, it is possible to derive the 3D shape of the refractive air-water boundary.

Figure 4.2 shows a schematic representation of the imaging setup, in which light rays are traced from discrete points beneath the water surface to an ideal camera. The center of projection of the camera is located at point **c** and the points are imaged on the image plane (I). The intersection of the rays with this image plane determine the location of points **q** and **q'**. These points correspond with the image of feature point **f**, located on the bottom of the tank, respectively without and with water. The same ray from **c** through point **q'** would intersect with the bottom of the tank at point **f'** in case no refraction would occur. For a classical stereo setup, the distance between the image points of one point in the scene in the two image planes is defined as the disparity (as explained in Chapter 3). Based on similarity, the distance between **f** and **f'** is defined by Morris (2004) as 'the refractive disparity'. In the following, we denote the distance between points **q** and **q'** in the image plane as the corresponding 'image disparity shift'.

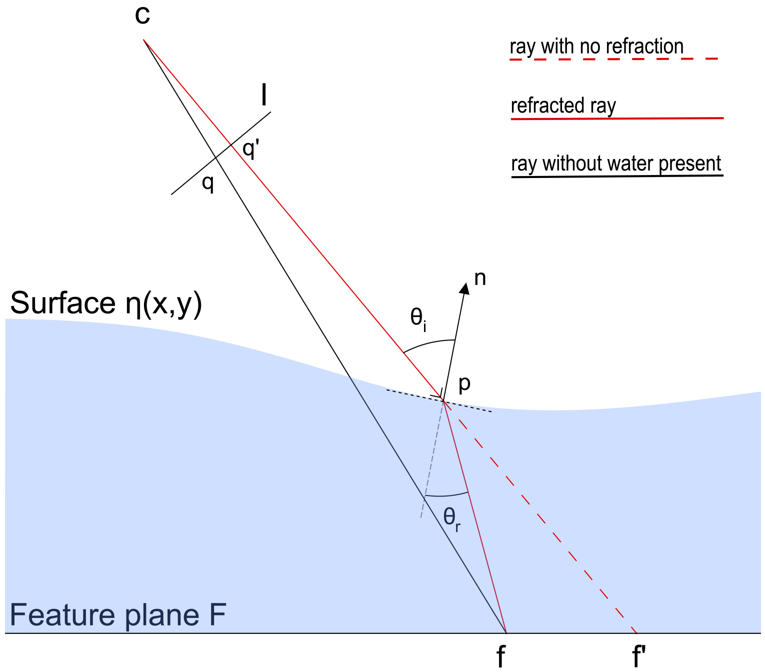


Figure 4.2: Schematic representation of the refraction of the viewing rays through **q'**. In case no refraction occurs, the feature points would be located at **f'**. Due to refraction, the light rays change in direction at the air-water interface and intersect the feature plane at points **f**. Based on Morris (2004).

The solution of the reconstruction problem can be described in function of two unknowns, namely the position of the surface point **p** and the direction of the corresponding surface normal $\vec{n}$. In case either **p** or the direction of $\vec{n}$ is known, the other parameter can be determined based on the equations that are described in Section 4.3.1.

The total solution space contains all surface point-surface normal pairs ($\mathbf{p}$,$\vec{n}$) that refract the light ray from **f** to the image point **q'** as shown in Figure 4.3. The surface normal remains however still bounded by the physical limits of refraction. The angle between the light rays in water and the direction of the surface normal (θ_r) has therefore a maximum equal to the critical angle θ_{cr}.

Morris (2004) solves the ambiguity of this imaging problem by using a hypothesized distance along the light ray (depth along the ray) between the camera center **c** to the water surface. Using the refractive properties of water, the surface normal at the intersection of this light ray

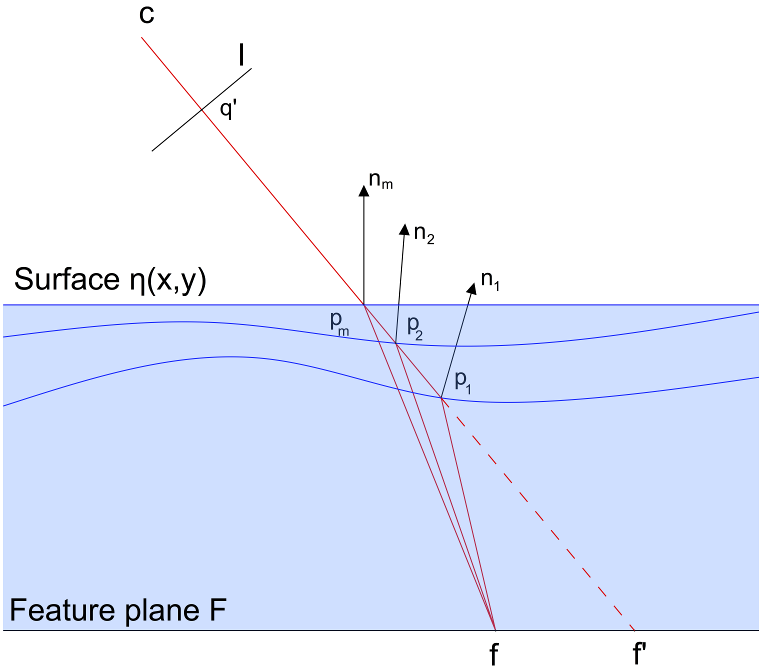


Figure 4.3: Illustration of the possible solutions with surface points $(\boldsymbol{p_1},\boldsymbol{p_2},\boldsymbol{p_i},\ldots,\boldsymbol{p_m})$, each with a corresponding surface normal $\overrightarrow{\boldsymbol{n}}_{\boldsymbol{i}}$ that refracts the ray starting at **f** towards the camera center **c**. Based on Morris (2004).

with the water can then be determined. In order to verify this hypothesized depth along the ray, an alternative prediction of the water surface is required. Because water is a highly dynamic liquid, multiple views of the surface with a single camera for one specific surface shape is however not possible. Morris (2004) uses therefore a second camera as depicted in Figure 4.4. Using the image points detected by the second camera, these secondary refractive displacements can be used to find an optimal depth along the ray. This comprises a one-dimensional search for the optimal depth along the ray of the so-called 'reference camera', which minimizes a certain error metric. This cost function expresses the dissimilarity between the two computed surface normals using one of both cameras in the area where their views overlap.

In this thesis, a different approach is developed. We make use of prior knowledge about the physical phenomenon that is captured to derive a general formulation of the surface shape that needs to be reconstructed. In most cases, a good prediction of the possible shape of the water surface can be made in case we limit ourselves to one specific hydrodynamic phenomenon. As such, a general solution that describes the surface shape can be proposed in which parameters are incorporated to fit the model to the specific water surface under consideration.

The principle of using a low parameter model to describe the shape of refractive objects has already been applied by Ben-Ezra and Nayar (2003). They however assumed a distant background pattern to obtain the best fitting value of a single shape parameter ξ. We intend to combine the approach of Morris (2004) with a parametric description of the water surface to obtain an algorithm which allows the spatio-temporal reconstruction of a varying water surface. A general derivation of this theoretical model for the case of self-induced sloshing is given hereafter. In case another hydrodynamic phenomenon would be studied, the theoretical model to describe the surface shape should be adapted based on the expected shape of the reconstructed water surface.

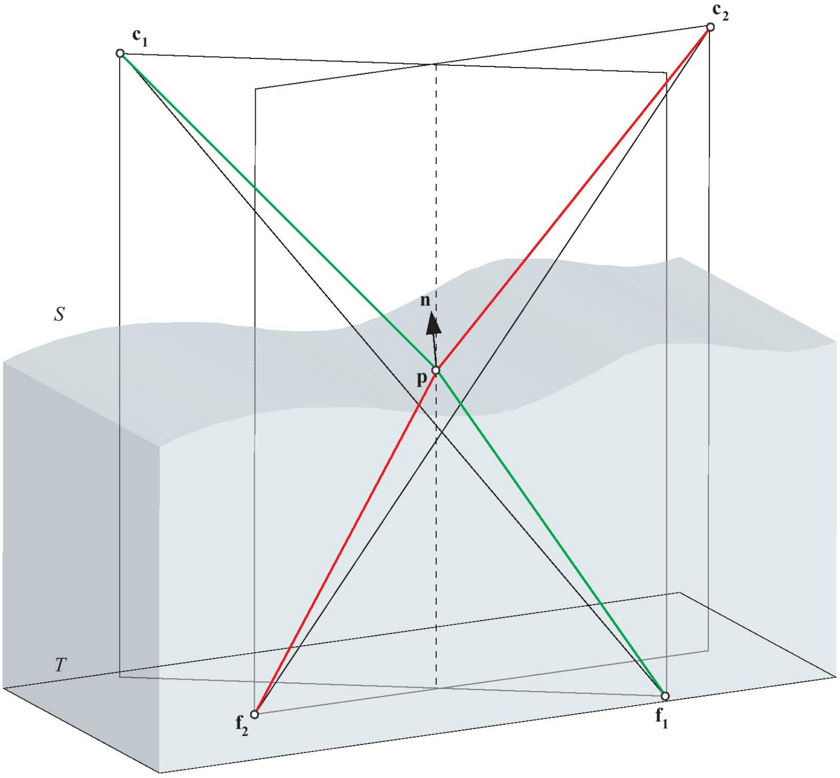


Figure 4.4: Morris (2004) uses a second camera to obtain a second set of surface normals to verify the hypothesized location of surface points **p**. From Morris (2004).

4.2 Deriving a predefined shape for the water surface

4.2.1 Theoretical derivation of the water surface shape

This section deals with the description of oscillating water surfaces in a rectangular tank. The theory of Lamb (1932) is used to obtain a theoretical model that represents the water surface shape. The derivation given hereafter is based on a more recent and practical description of Lambs theory in Anonymous.

The initial water surface is supposed to be a plane sheet located at a uniform depth h relative to the bottom of the tank. In case the vertical acceleration can be neglected, we can assume that the fluid motion is uniform over the entire depth h. The horizontal motion of the fluid particles is therefore the same for all particles on the same vertical line. A Cartesian coordinate system is defined in which the x- and y-axes are assumed to be horizontal. The horizontal velocity components are further denoted as u and v in respectively the x- and y-direction. ζ(x,y) is the corresponding elevation of the free surface above the undisturbed water level at a point (x,y) in the tank. The equation of continuity is expressed by calculating the flux of water through a columnar volume over a rectangular area $\partial x \partial y$.

$$\frac{\partial(u\,h\,\partial y)}{\partial x}\,\partial x + \frac{\partial(v\,h\,\partial x)}{\partial y}\partial y = -\frac{\partial}{\partial t}((\zeta + h)\,\partial x\,\partial y)$$

Neglecting the terms of the second order, this gives:

$$\frac{\partial \zeta}{\partial t} = -h\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) \tag{4.3}$$

The dynamic equation is a formulation of the Bernoulli equation for unsteady irrotational flow in case no disturbing force is present. This results in:

$$\rho\frac{\partial u}{\partial t} = -\frac{\partial p}{\partial x}, \qquad \rho\frac{\partial v}{\partial t} = -\frac{\partial p}{\partial y} \tag{4.4}$$

in which the pressure term is defined as:

$$p - p_0 = \rho g(h + \zeta - z)$$

where:

p_0	The atmospheric pressure.
h	The ordinate of the free surface in undisturbed state.
z	The vertical position above the bottom of the tank as shown in Figure 4.5.

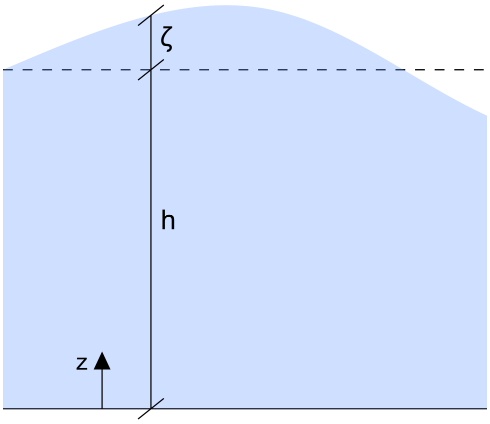


Figure 4.5: Illustration of the relationship between the different terms used to derive eq.(4.4)

This allows to further simplify the dynamic equation as follows:

$$\rho\frac{\partial u}{\partial t} = -\frac{\partial \zeta}{\partial x}, \qquad \rho\frac{\partial v}{\partial t} = -\frac{\partial \zeta}{\partial y} \tag{4.5}$$

By eliminating u and v this finally results in:

$$\frac{\partial^2 \zeta}{\partial t^2} = c_{wave}^2\left(\frac{\partial^2 \zeta}{\partial x^2} + \frac{\partial^2 \zeta}{\partial y^2}\right) = c_{wave}^2 \Delta\zeta \tag{4.6}$$

where the constant c_{wave} is the propagation speed of the wave: $c_{wave}^2 = gh$. This equation is also known as the two-dimensional wave equation, in which the expression between parentheses is called the Laplacian $\Delta\zeta = \nabla^2\zeta$. In the case of simple harmonic motions, the equations can be shortened if a complex time factor $e^{i(\sigma t+\epsilon)}$ is assumed. This results in:

$$\begin{cases} u = \frac{i\,g}{\sigma}\frac{\partial \zeta}{\partial x} \\ v = \frac{i\,g}{\sigma}\frac{\partial \zeta}{\partial y} \end{cases} \tag{4.7}$$

$$\frac{\partial^2 \zeta}{\partial x^2} + \frac{\partial^2 \zeta}{\partial y^2} + \nu^2\zeta = 0 \tag{4.8}$$

where $\nu^2 = \frac{\sigma^2}{c_{wave}^2}$ This last expression in eq. (4.8) derived from the general eq. (4.6) is also called the two-dimensional Helmholtz equation.

In order to solve this problem, boundary conditions need to be specified. Each boundary condition is described as a conditional equation on the boundary curve C in the xy-plane. Three main types can be distinguished:

I. **Dirichlet boundary condition**: u is prescribed on C.

II. **Neumann boundary condition**: the normal derivative $u_n = \frac{\partial \zeta}{\partial n}$ is prescribed on C.

III. **Robin boundary condition**: $\zeta + b\frac{\partial \zeta}{\partial n}$ is prescribed on C.

The vertical bounding wall of the rectangular tank corresponds with 'a free boundary' which can be expressed by the following Neumann B.C.:

$$\frac{\partial \zeta}{\partial n} = 0 \tag{4.9}$$

where ∂n denotes the normal to the boundary of the element. In case a rectangular spatial domain is adopted, further denoted as $\Omega((x,y)|0 < x < L_x, 0 < y < L_y)$ with L_x and L_y the width and length of the tank, the boundary conditions can be formulated in function of x and y:

$$B.C. \quad \frac{\partial \zeta}{\partial x}(0,y,t) = 0, \quad \frac{\partial \zeta}{\partial x}(L_x,y,t) = 0, \quad 0 \leq y \leq L_y, t > 0 \tag{4.10}$$

$$B.C. \quad \frac{\partial \zeta}{\partial y}(x,0,t) = 0, \quad \frac{\partial \zeta}{\partial y}(x,L_y,t) = 0, \quad 0 \leq x \leq L_x, t > 0 \tag{4.11}$$

The partial differential equation (PDE) given by eq. (4.8) can further be simplified by applying separation of variables. The function $\zeta(x,y)$ is split into H(x)Q(y) and substituting this in eq. (4.8), this result in:

$$H''(x)\,Q(y) = -H(x)\,(Q'(y) + \nu^2 Q(y))$$

The two variables can be separated by dividing both sides by H(x)Q(y) which results in:

$$\frac{H''(x)}{H(x)} = -\frac{1}{Q(y)}(Q''(y) + \nu^2 Q(y))$$

Both sides must be constant as the left hand side is only dependent of x and the right hand side only dependent of y. It is assumed that this constant is negative and equal to - k^2:

$$\frac{H''(x)}{H(x)} = -\frac{1}{Q(y)}(Q''(y) + \nu^2 Q(y)) = -k^2$$

This yields in following ordinary different equations (ODE's) for H and Q:

$$H''(x) + k^2 H(x) = 0, \qquad 0 < x < L_x \tag{4.12}$$

$$Q''(y) + p^2 Q(y) = 0, \qquad 0 < x < L_y \tag{4.13}$$

where $p^2 = k^2 - \nu^2$. Both boundary conditions can be formulated in function of H and Q as follows:

$$H_x(0) = 0, \qquad H_x(L_x) = 0$$
$$Q_y(0) = 0, \qquad Q_y(L_y) = 0$$

The general solution for eq. (4.12) and eq. (4.13) in combination with the boundary conditions can therefore be solved separately.

The general solution for H(x) can be written as $H(x) = \varphi_x\,cos(kx) + \psi_x\,sin(kx)$, and by substituting $H'(0) = 0$; $H'(L_x) = 0$, this results in:

$$\varphi_x\,sin(kx) = 0, \quad \psi_x = 0 \tag{4.14}$$

in which $A \neq 0$ in order to find a solution that is not identically zero. This results in:

$$k = k_m = \frac{m\pi}{L_x}, \quad H_m(x) = \varphi_{x,m}\, cos(\frac{m\pi x}{L_x}), \quad m = 1{,}2, \dots \tag{4.15}$$

The problem for Q(y) can be solved similarly which results in:

$$p = p_n = \frac{n\pi}{L_y}, \quad Q_n(y) = \varphi_{y,n}\, cos(\frac{n\pi y}{L_y}), \quad n = 1{,}2, \dots \tag{4.16}$$

The solutions for the Helmholtz equation can as such be written as:

$$\zeta_{mn}(x{,}y) = H_m(x)Q_n(y) = \varphi_{x,m}\, cos(\frac{m\pi x}{L_x})\, \varphi_{y,n}\, cos(\frac{n\pi y}{L_y}), \quad m{,}n = 1{,}2, \dots \tag{4.17}$$

where $\nu_{mn}^2 = k_m^2 + p_n^2 = (\frac{m\pi x}{L_x})^2 + (\frac{n\pi y}{L_y})^2$.

This equation describes the water surface fluctuations in the entire spatial domain at each specific time instance t. In order to describe the actual surface height, the initial water depth has to be added which is represented by the component corresponding with m=0 and n=0. This results in following double Fourier's series:

$$\eta(x{,}y) = \sum_m \sum_n A_{mn}\, cos(\frac{m\pi x}{L_x})\, cos(\frac{n\pi y}{L_y}), \quad m{,}n = 0{,}1{,}2, \dots \tag{4.18}$$

4.2.2 Surface model adopted in this work

The surface model that will be used in this thesis is inspired on the derivations of Lamb (1932) as explained in the previous section. We will however apply our surface model to cases in which his assumptions are strictly speaking no longer valid (e.g. vertical velocity=0, ζ considered small, ...). Our surface model will however be parameterized to fit the actual shape of the water. Accurate reconstruction is in that case determined by the choice of an appropriate model (that is able to describe the actual surface shape), rather than the assumptions made during the derivation. The error made by violating Lamb's assumptions can therefore be considered negligible.

In Chapter 2, we showed that in case of self-induced sloshing usually the first and second order modes are present. For this reason, we will limit the infinite sum in eq. (4.18) and only adopt a limited amount of cosine terms to describe the shape of the water surface. For most hydrodynamic phenomena where the water surface remains smooth, the error made by neglecting the higher order terms can be considered small. The model will however be extended with a linear component in both the x- and y-direction to cope with small inclinations of the tank bottom as will be explained in Chapter 6. This finally results in our 'surface function' $\eta(x{,}y)$ that will be used to describe the surface heights within the tank at one specific time instance:

$$\begin{aligned}\eta(x{,}y) =& A_{00} + A_{10}\, cos(\frac{\pi x}{L_x}) + A_{01}\, cos(\frac{\pi y}{L_y}) + A_{11}\, cos(\frac{\pi x}{L_x})\, cos(\frac{\pi y}{L_y}) + \\ & A_{20}\, cos(\frac{2\pi x}{L_x}) + A_{02}\, cos(\frac{2\pi y}{L_y}) + B\frac{x}{L_x} + C\frac{y}{L_y}\end{aligned} \tag{4.19}$$

where A_{mn}, B and C are the unknown, time-dependent coefficients that need to be fitted to describe the instantaneous surface shape.

In case the function $\eta(x{,}y)$ that describes the water surface is assumed to be known, the surface point locations **p** of the projected features points **f** on the water surface can easily be determined. To that end, we make an estimate of the coefficients A_{mn}, B and C in order to define a ‘hypothesized surface’ $\eta(x,y)$. Each surface point **p** is then computed as the intersection point of the viewing ray $\overrightarrow{\boldsymbol{cq'}}$ with this hypothesized water surface. Given the location of **p**, every normal is found as the surface normal that accounts for the refraction of the incident light ray at point **p** towards point **f**.

In order to verify our hypothesis about the coefficients A_{mn}, B and C, an alternative set of surface normals is required similarly as in Morris (2004). For this purpose, we use the mathematical formulation of eq. (4.19) to compute the normal at every location on the surface defined by $\eta(x,y)$. This will further be elaborated in the section 4.3.2. Both sets can then be compared in order to verify if the hypothesized coefficients in eq. (4.19) are correct.

4.3 Description of the global reconstruction algorithm

This section involves the determination of the surface shape in an ideal imaging model, in which inaccuracies related to incorrect camera calibration, pose estimation or feature localization are not considered. We combine two methods to determine the surface normal given a known surface location. Based on a certain error metric which expresses the dissimilarity between both, the best-fitting surface can be found by minimizing this cost function over all feature points.

4.3.1 Determination of the surface normal based on refractive displacements

The location of the physical feature points **f** is known, as well as the position of the calibrated camera with center **c**. For every feature point **f**, its location in the image plane I due to refraction at the air-water interface is detected resulting in a list of image points **q’**. In the previous section, we showed that in case the coefficients A_{mn}, B and C in eq. (4.19) are hypothesized, every surface point **p** can be computed as the intersection of the ray $\overrightarrow{\boldsymbol{cq'}}$ with the 3D surface shape defined by $\eta(x,t)$. Given these 4 known points (**f, c, q’** and **p**), the entire refractive geometry is defined and the surface normal can be computed.

Firstly, the angle θ_δ in Figure 4.6 is defined as follows:

$$\theta_\delta = \theta_i - \theta_r \tag{4.20}$$

where both θ_i and θ_r are initially unknown.The locations of surface point **p**, feature point **f** and image point **q’** are however given, which define the vectors of the incident ($\overrightarrow{\boldsymbol{u}}$) and refracted rays ($\overrightarrow{\boldsymbol{v}}$):

$$\overrightarrow{\boldsymbol{u}} = \overrightarrow{\boldsymbol{cq'}} = \overrightarrow{\boldsymbol{q'f'}} \tag{4.21}$$

$$\overrightarrow{\boldsymbol{v}} = \overrightarrow{\boldsymbol{pf}} \tag{4.22}$$

Both vectors can be normalized to obtain respectively $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{v}}$:

$$\hat{\boldsymbol{u}} = \frac{\overrightarrow{\boldsymbol{u}}}{\|\overrightarrow{\boldsymbol{u}}\|} \tag{4.23}$$

$$\hat{\boldsymbol{v}} = \frac{\overrightarrow{\boldsymbol{v}}}{\|\overrightarrow{\boldsymbol{v}}\|} \tag{4.24}$$

Based on simple linear algebra, the inner product of $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{v}}$ allows to compute the angle θ_δ:

$$\theta_\delta = cos^{-1}(\hat{\boldsymbol{u}} \cdot \hat{\boldsymbol{v}}) \tag{4.25}$$

If eq. (4.20) is substituted into Snell's law, the use of trigonometric identities allows to obtain an expression for the incident angle θ_i:

$$\begin{aligned} sin(\theta_i) &= r_w sin(\theta_i - \theta_\delta) \\ sin(\theta_i) &= r_w[sin(\theta_i)cos(\theta_\delta) - cos(\theta_i)sin(\theta_\delta)] \\ tan(\theta_i) &= \frac{r_w sin(\theta_\delta)}{r_w cos(\theta_\delta) - 1} \\ \theta_i &= tan^{-1}\left(\frac{r_w sin(\theta_\delta)}{r_w cos(\theta_\delta) - 1}\right) \end{aligned} \tag{4.26}$$

This equation allows to calculate θ_i for a known refractive index r_w. The surface normal $\overrightarrow{\boldsymbol{n}}$ is finally determined by rotating $\hat{\boldsymbol{u}}$ by θ_i about the axis defined by $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{v}}$:

$$\overrightarrow{\boldsymbol{n}} = R(\theta_i, \hat{\boldsymbol{u}} \times \hat{\boldsymbol{v}})(-\hat{\boldsymbol{u}}) \tag{4.27}$$

where R(θ, $\hat{\boldsymbol{X}}$) represents the rotation matrix of an angle θ about an (normalized) axis $\hat{\boldsymbol{X}}$. This rotation is done based on a right-handed Cartesian coordinate system, in which a rotation is considered positive in counter-clockwise direction. The corresponding rotation matrix for the rotation around an arbitrary, normalized axis $\hat{\boldsymbol{X}}(X_x,X_y,X_z)$ over an angle θ is given by:

$$R = \begin{bmatrix} \cos\theta + X_x^2\left(1-\cos\theta\right) & X_xX_y\left(1-\cos\theta\right) - X_z\sin\theta & X_xX_z\left(1-\cos\theta\right) + X_y\sin\theta \\ X_yX_x\left(1-\cos\theta\right) + X_z\sin\theta & \cos\theta + X_y^2\left(1-\cos\theta\right) & X_yX_z\left(1-\cos\theta\right) - X_x\sin\theta \\ X_zX_x\left(1-\cos\theta\right) - X_y\sin\theta & X_zX_y\left(1-\cos\theta\right) + X_x\sin\theta & \cos\theta + X_z^2\left(1-\cos\theta\right) \end{bmatrix} \tag{4.28}$$

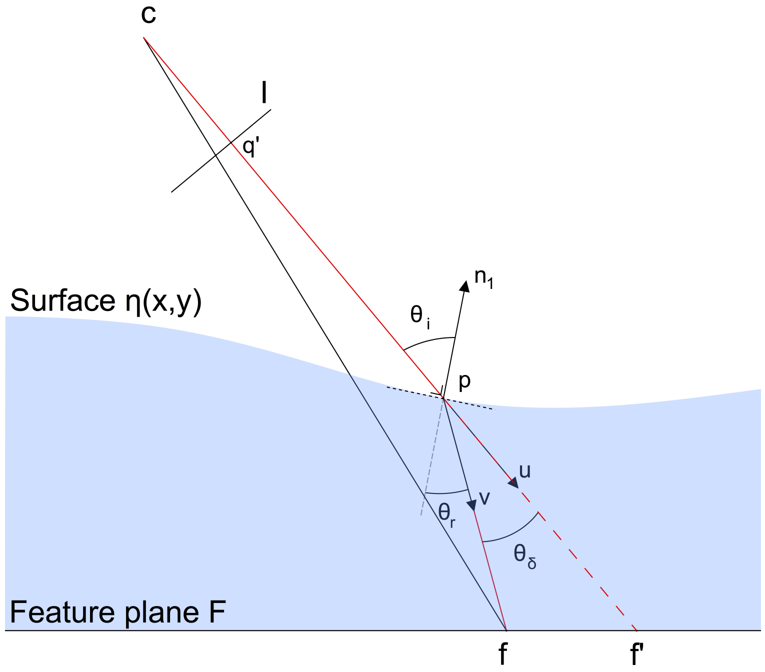


Figure 4.6: Illustration of derivation of surface normal using the refractive disparity. Given the location of **p**, the image disparity shift from **q** to **q'** in the image plane allows to compute surface normal $\overrightarrow{\boldsymbol{n}}_{\boldsymbol{1}}$. Based on Morris (2004).

Morris (2004) proves that for every refractive disparity of a point **f** observed by a camera with its center of projection at **c** and a certain hypothesized surface point **p**, there is at most one normal

$\overrightarrow{\boldsymbol{n}}$ such that the ray from **c** to **p** is refracted to **f**. In the rest of this chapter, the computed surface normal based on the refractive displacements of the feature points will be denoted as $\overrightarrow{\boldsymbol{n_1}}$ to distinguish it from the derivation explained in the following paragraph.

4.3.2 Determination of the surface normal based on the theoretical description of the water surface

As already explained in Section 4.2, the water surface can be described by a parametric equation which gives the relationship between the x-, y- and z-coordinates of the points located on the water surface. In case terms of the third order or higher are neglected, our surface function which describes the local water height above the bottom of the tank becomes:

$$\begin{aligned}\eta(x,y) = z = & A_{00} + A_{10}\,cos(\frac{\pi x}{L_x}) + A_{01}\,cos(\frac{\pi y}{L_y}) + A_{11}\,cos(\frac{\pi x}{L_x})\,cos(\frac{\pi y}{L_y}) + \\ & A_{20}\,cos(\frac{2\pi x}{L_x}) + A_{02}\,cos(\frac{2\pi y}{L_y}) + B\frac{x}{L_x} + C\frac{y}{L_y}\end{aligned} \tag{4.29}$$

By rearranging the different terms, this equation can be written in its implicit form as F(x,y,z)=0. This function F(x,y,z) is given by:

$$\begin{aligned}F(x,y,z) = z - & A_{00} - A_{10}\,cos(\frac{\pi x}{L_x}) - A_{01}\,cos(\frac{\pi y}{L_y}) - A_{11}\,cos(\frac{\pi x}{L_x})\,cos(\frac{\pi y}{L_y}) - \\ & A_{20}\,cos(\frac{2\pi x}{L_x}) - A_{02}\,cos(\frac{2\pi y}{L_y}) - B\frac{x}{L_x} - C\frac{y}{L_y}\end{aligned} \tag{4.30}$$

The surface normal at a specific point (x,y) on the surface can then be determined by taking the derivative to x, y and z to find respectively the x-, y- and z-components of the surface normal $\overrightarrow{\boldsymbol{n}}$:

$$\begin{cases} n_x = \frac{\partial F}{\partial x} = A_{10}\,\frac{\pi}{L_x} sin(\frac{\pi x}{L_x}) + A_{11}\,\frac{\pi}{L_x} sin(\frac{\pi x}{L_x}) cos(\frac{\pi y}{L_y}) + A_{20}\,\frac{2\pi}{L_x} sin(\frac{2\pi x}{L_x}) - \frac{B}{L_x} \\ n_y = \frac{\partial F}{\partial y} = A_{01}\,\frac{\pi}{L_y} sin(\frac{\pi y}{L_y}) + A_{11}\,\frac{\pi}{L_y} cos(\frac{\pi x}{L_x}) sin(\frac{\pi y}{L_y}) + A_{02}\,\frac{2\pi}{L_y} sin(\frac{2\pi y}{L_y}) - \frac{C}{L_y} \\ n_z = \frac{\partial F}{\partial z} = 1 \end{cases} \tag{4.31}$$

which is finally normalized:

$$\hat{\boldsymbol{n}} = \frac{\overrightarrow{\boldsymbol{n}}}{\|\overrightarrow{\boldsymbol{n}}\|} \tag{4.32}$$

The normalized vector $\hat{\boldsymbol{n}}$, further denoted as $\overrightarrow{\boldsymbol{n_2}}$ will in the following paragraph be compared with the surface normal $\overrightarrow{\boldsymbol{n_1}}$ obtained in Section 4.3.1.

4.3.3 Error computation: expressing the dissimilarity between n_1 and n_2

The previous two sections explained two alternative methods to calculate the surface normal at discrete locations of the water surface. We assumed that the entire water surface shape was known, based on a set of hypothesized coefficients as defined in Section 4.2. These two sets of normals, $\overrightarrow{\boldsymbol{n_1}}$ and $\overrightarrow{\boldsymbol{n_2}}$, are in this step compared to verify if the hypothesized coefficients are correct. Different error metrics can be used to obtain a cost function that quantifies the dissimilarity between both. Two alternatives will be used in this work, based on the error metrics defined by Morris (2004).

A first option is to use 'the normal collinearity metric' (M1), which directly matches the surface normals derived with both methods. This metric (E_{col}) represents the angle between two corresponding normals and can be computed as follows:

$$E_{col} = cos^{-1}(\overrightarrow{\boldsymbol{n_1}} \cdot \overrightarrow{\boldsymbol{n_2}}) \tag{4.33}$$

For corresponding surface normals that are collinear, i.e. a small angle between the two normal directions, this results in a small value of E_{col}. In contrast, a large angle between both gives a clear indication that the assumed coefficients are not correct.

A second error metric is based on 'the disparity difference metric' derived by Morris (2004). He uses the change in refractive disparity that occurs when an alternative set of surface normals (obtained with the 'verification camera') is swapped with his first set $\overrightarrow{\boldsymbol{n_1}}$ (obtained by the 'reference camera'). The incident rays from the camera are refracted assuming an orientation of the water surface defined by the second set of surface normals. The refracted rays are subsequently traced back to the bottom surface and the 'disparity difference' is measured.

The metric used here is an application of this principle. The normal set $\overrightarrow{\boldsymbol{n_1}}$ is swapped with $\overrightarrow{\boldsymbol{n_2}}$ at the discrete points $\mathbf{p}$ of the hypothesized water surface. In case $\overrightarrow{\boldsymbol{n_1}}$ and $\overrightarrow{\boldsymbol{n_2}}$ are not the same, a certain 'refractive displacement' of the detected feature point on the feature plane F will occur. Figure 4.7 shows this displacement from $\boldsymbol{f}$ to $\boldsymbol{f_2}$. The distance between these paired points defines the second error metric, denoted as the disparity difference metric E_{disp} (M2):

$$E_{disp} = |\boldsymbol{f} - \boldsymbol{f_2}| \tag{4.34}$$

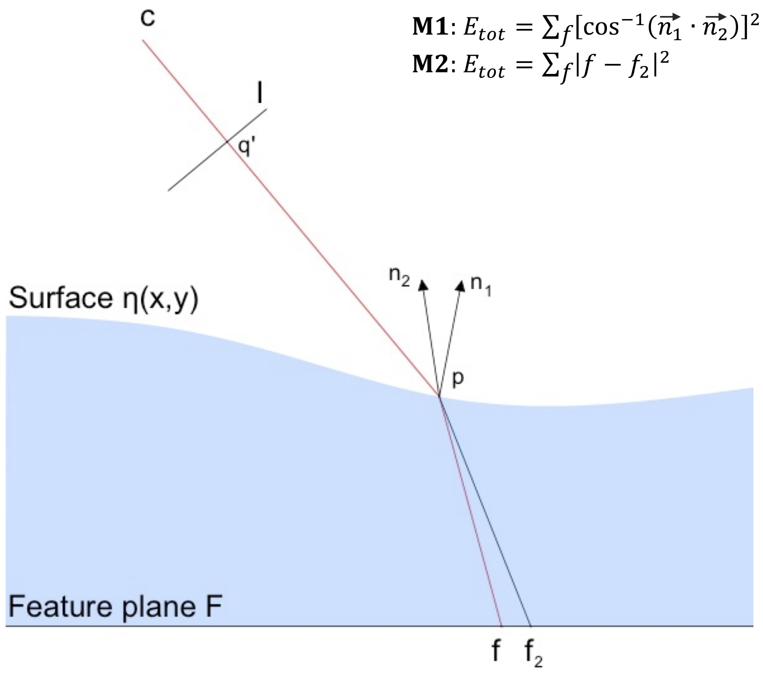


Figure 4.7: Illustration of the two possible error metrics that measure the dissimilarity between the refractive-based surface and the parameterized surface model: the normal collinearity metric (M1) and the disparity difference metric (M2). Based on Morris (2004).

On of both error measures is computed for all feature points $\mathbf{f}$, which are finally combined in a global error function E_{tot}:

$$E_{tot} = \sum_f E_f^2 \tag{4.35}$$

The most suitable set of surface shape coefficients is then found as the solution of a multivariate optimization which minimizes the total error E_{tot}. A detailed description of the practical implementation of our reconstruction algorithm is presented in Chapter 6.

4.4 Advantages and disadvantages of our multivariate reconstruction algorithm

As stated in Section 4.1, the developed method in this thesis is based on the work of Morris (2004). He presents a method in which the viewing rays of the camera are traced back to the bottom of the tank. The viewing ray for every feature point is refracted at the air-water interface and the most appropriate depth-normal pair is determined for each feature point separately. A 'verification camera' is then used to validate if the hypothesized 'depth along the ray' is indeed correct. In contrast, our approach does not consider the viewing rays of the feature points independently. We combine the entire set of feature points which allows to fit a parameterized surface model to the observed refractive disparities. This has both advantages as disadvantages compared to the original approach of Morris (2004).

4.4.1 Advantages

A list of advantages compared to the original method of Morris (2004) is given hereafter:

1. Loss of feature points can more easily be handled.

   Both methods are based on the same principle in which the projection of a feature pattern on the water surface is viewed by the camera. In Chapter 8, optical flow methods will be elaborated which are used to track these feature points in an image sequence or video of the water surface. Feature points can however become difficult to locate in case of strong fluctuations of the water surface. The corresponding irregular surface shapes cause large displacements of the surface points, whose positions are governed by Snell's law. The corresponding 'velocities' of these feature points over the surface might in that case be to large to follow and also motion blur then becomes a risk. In case the optical flow methods cannot locate a feature in the image, the original method of Morris (2004) can no longer be used to predict the surface normal and water depth at that location. Feature points need to be located for both the the 'reference camera' and the 'verification camera' in order to avoid that the necessary interpolation (or extrapolation) of surface normals is based on too few known positions. Using two camera however doubles the risk that some feature points are not located by at least one of the cameras.

   Our approach however allows to reconstruct the water surface based on the observed surface points of a single camera. Loss of feature points by one camera can in that case even be compensated by other cameras for which the feature points were not lost. Less interpolation is therefore needed to reconstruct areas where feature points were lost, allowing a more accurate surface reconstruction.

2. The computational time is greatly diminished.

   Our approach uses a theoretical surface model to describe the water surface. The coefficients in this surface function need to be optimized to describe the instantaneous shape of the water as good as possible. This involves a multivariate optimization of a limited amount of parameters that need to be fitted to the observed image disparity shifts. The total error E_{tot} (given by eq. (4.35)) is adopted as a cost function that needs to be minimized. In contrast, Morris (2004) optimizes a hypothesized 'depth along the ray' for each individual feature point. Every surface reconstruction therefore requires N one-dimensional optimization procedures, in which N denotes the number of adopted feature points in the reconstruction.

A comparison was made between our single, multivariate optimization and these N one-dimensional optimizations. This showed that for a reasonable amount of feature points, the computational cost of the original approach is significantly larger. Using only a limited number of coefficients to describe the water surface, the time necessary for optimization can therefore greatly be diminished. This advantage becomes more important in case a large surface area needs to be reconstructed. The amount of feature points to obtain an accurate reconstruction then rapidly increases, which makes our approach even more beneficial.

3. The technique is more robust for inaccurate feature localization.

   Because our solution combines the image disparity shifts of all feature points in a single optimization, inaccurate localization of a single feature point is mitigated as it only contributes for a small part to the total error E_{tot}. As will be explained in Chapter 7, the detection of the image points **q** and **q'** is not always perfect. This causes that the input parameters of the geometric problem corresponding to a single feature point are not exactly correct. The perfect solution corresponding to the actual water surface can in that case never be found. For large localization errors, the solution might even diverge and a very irregular and incorrect surface could be obtained. By combining the errors of each feature point in a total error function, the reconstruction becomes much more robust. Inaccuracies related to some feature points is in that case relieved by the feature points for which a more correct position was determined. Incorrect feature localization is therefore less detrimental, resulting in a more stable algorithm.

4. The amount of cameras and their relative position can be varied.

   In the method of Morris (2004), the surface points that are viewed with the verification camera do not coincide with the surface points viewed by the reference camera. Bilinear interpolation is therefore required to obtain two surface normals in the same point **p** that subsequently can be compared. Surface reconstruction is in that case only possible in the area of the surface where their views overlap. In contrast, our method is able to reconstruct the water surface based on a combination of a single camera with a parameterized form of the surface shape. The entire area covered by surface points can therefore accurately be reconstructed and no verification camera is needed.

   In Chapter 7, it will however be shown that the accuracy and robustness of our reconstruction algorithm increases in case multiple cameras are used. As will be explained in Chapter 7, errors related to one camera can in that case be mitigated by the other camera in the area where their views overlap.

   Secondly, multiple cameras could be adopted to reconstruct a larger area of the water surface. Each individual camera is in that case used to reconstruct the area of the surface that is viewed by that particular camera. A schematic representation of such a camera setup is given in Figure 4.8. Overlap between the camera views is in theory not needed, although it might improve smoothness of the finally obtained solution.

4.4.2 Disadvantages

Our approach has however also some disadvantages compared to the original proposed method:

1. The expected surface shape needs to be known.

   In Morris (2004), the algorithm proposed does not assume a certain shape of the water surface. Each surface point is optimized individually, which allows to reconstruct arbitrary

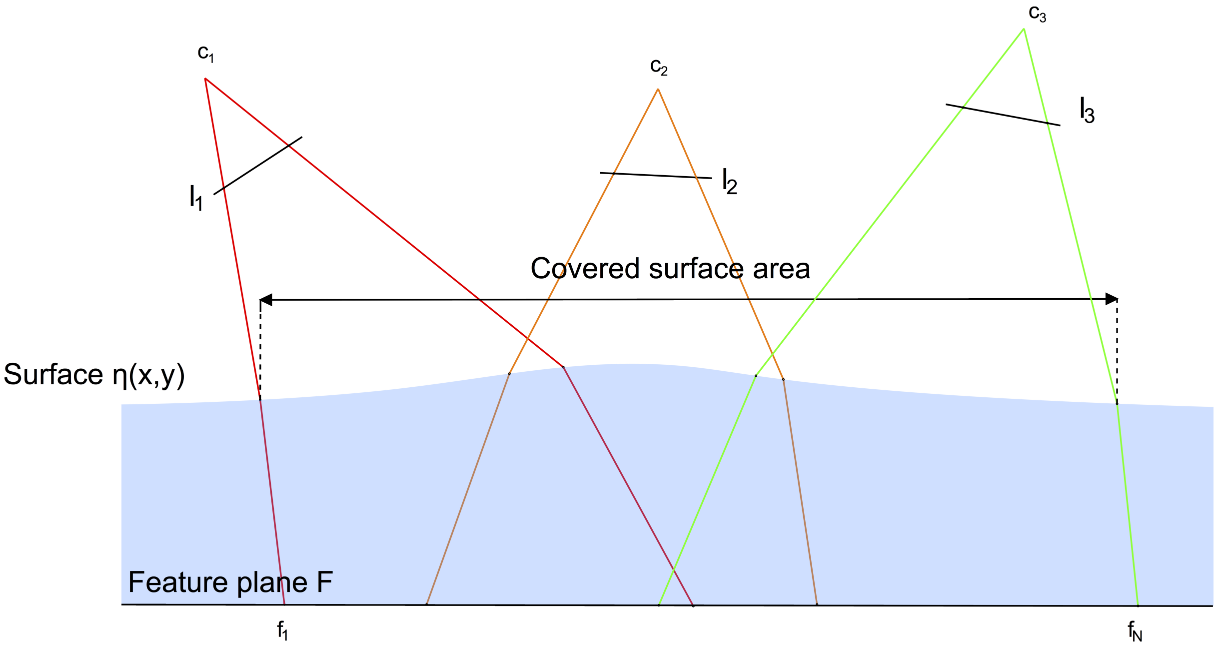


Figure 4.8: Spatial extent of the surface reconstruction can be enlarged by using multiple cameras.

surface shapes. Irregular surfaces or even test cases in which the tank is only partly filled have successfully been reconstructed using this method.

Our approach requires however prior knowledge of the phenomenon that is studied. Because the final solution is the optimized form of a predefined parameterized surface model, it is necessary that this model is sufficient to describe the actual surface shape.

In Section 4.2, we presented a surface model that can be used to describe the water surface in case of self-induced sloshing. In that case, the water surface oscillations remain small and the surface can be approximated by our chosen surface function $\eta(x,y)$ (eq. (4.19)). If our method is applied to study other hydrodynamic phenomena, this smooth representation of the surface might not be valid anymore. The surface function should in that case be adapted to allow a good representation of what is observed in the scene. For some phenomena, the water surface is however difficult to describe by a mathematical closed-form solution. This limits the general applicability of our approach.

2. The performance of the reconstruction is dependent on the surface function that is used.

   In Chapter 7, the influence of using different possible surface functions is studied with respect to accuracy and robustness. As will be shown, the performance of the algorithm is dependent on the adopted surface function. Although additional parameters allow a more irregular surface to be reconstructed, this decreases the robustness of the algorithm w.r.t. inaccuracies of feature localization or the estimation of the camera parameters. Additionally, incorporating more coefficients in the surface function partly cancels the advantage of a shorter computation time compared to the N one-dimensional optimizations.

   Because it is the intention to apply our approach to self-induced sloshing, the reduced model described by eq. (4.19) is assumed sufficient. For the tested image sequences, this proved to be robust and accurate and offered the advantage of fast processing.

Chapter 5

Corner and feature detection

The previous chapter explained the general methodology that will be used to reconstruct the water surface. Our reconstruction procedure is based on the localization of a sparse set of feature points in images of the oscillating water mass within the tank. These observed features are in fact projections on the water surface of points that are located on the bottom of the tank in plane F (see Figure 4.2 in Chapter 4). In order to solve the geometric reconstruction problem, feature detection is required to obtain the pixel coordinates of the pattern in the image plane (**q'** in Figure 4.2).

In the following, a brief overview of possible corner and feature detection algorithms is given, as well as the practical implementation of these methods used in this thesis.

5.1 Overview of corner detection algorithms

5.1.1 Introduction

Corner detection is used in many image processing and computer vision applications, such as motion detection, video tracking, object recognition, image stitching, camera calibration and stereo matching. Corner detection partly overlaps with finding interesting points in images (features) and detecting corresponding points across multiple images.

Firstly, corners can be defined as the intersection of two edges, or more precisely as the point where two dominant and different edge directions in the neighborhood around that point are present. Secondly, interesting points in an image (feature points) are points that are well-defined and can robustly be detected. Typically, these are points with a local maximum or minimum in intensity or curvature; 'real corners' being only one example of such points.

5.1.2 Types of corner detection methods

Corner detection methods can be subdivided in three basic categories (Patel and Panchal (2014); Kahaki et al. (2014)):

1. **Template- or model-based corner detectors.**
   These methods use a representative template which is matched to the image in order to detect the corners in the image. They prove more advised in terms of robustness and efficient computational cost. These methods however suffer from the fact that their performance is highly dependent on the templates that are used. In case different types of

corners are present, the used templates must be matchable with every possible corner in the image. A second consideration is that after correlating the image with the templates, the choice of threshold to retain a region in the image as corner severely affects the amount of corners detected.

2. **Contour-based corner detectors.**
   These corner detection methods are based on the idea that initially edges (i.e. lines) in the images are detected. Based on the found contours, contour-based methods search subsequently for corner points along these edges.

3. **Direct or intensity-based corner detection methods.**
   Direct corner detection methods are based on mathematical computations on the images, usually by applying statistical operations on the image. These typically consist of computing the first- or second-order derivatives of the gray-level intensity. The results of these computations are subsequently used to locate the corners in the image.

In the overview given hereafter, we start with discussing intensity-based corner detectors. These methods are the oldest corner detection algorithms and form the basis for several recent developed methods. We then discuss several suggestions to locate corners with sub-pixel accuracy (Section 5.1.3.3 - Section 5.1.3.8). The final section of this overview gives an example of a template-based approach.

5.1.3 Overview

5.1.3.1 Moravec corner detector

This algorithm, derived by Moravec (1980), is one of the earliest corner detectors that was developed and searches for points with a low self-similarity. To that end, they compare a patch centered around the tested pixel with nearby, largely overlapping patches. A measure for the similarity between the patches is defined as the sum of squared differences (SSD) between two patches.

In case the pixel is located in a region with uniform intensity, the patches will be similar and the SSD will be small. For pixel located on an edge, the centered patch will show a large difference with patches perpendicular to the edge but in the direction of the edge, the change will be small. Pixels with variation in all directions will finally have a small similarity with patches in every direction and the SSD will thus be large. The so-called ‘cornerness strength’ can therefore be defined as the smallest SSD in all directions. Corners are subsequently defined as local maxima of the SSD-function.

This approach however suffers from non-isotopy because the centered patch is only compared with its neighbors in horizontal and vertical direction and on the two diagonals. In case edges are present in another direction, the smallest SSD will still be large and the edge will incorrectly be considered as a corner.

5.1.3.2 Harris-Stephens type of detection algorithms

The principle of this corner detection algorithm was developed by Harris and Stephens (1988), in which they rely on the observation that the pixel intensity shows significant changes in a close neighborhood of a corner. The method is based on the autocorrelation function of the image (Misra et al. (2012)), in which an image patch over an area (x,y) is shifted by a certain displacement (u,v). Subsequently, the difference in intensity between the original and shifted patch is computed. The difference in intensity is then multiplied with a window function (rectangular or Gaussian window) which gives weight to the surrounding pixels:

$$E(u,v) = \sum_{x,y} w(x,y)[I(x+u,y+v) - I(x,y)]^2 \tag{5.1}$$

where:

$w(x,y)$	The window function at position (x,y).
$I(x,y)$	The intensity at (x,y).
$I(x+u,y+v)$	The intensity in the moved window at (x+u,y+v).

Because corners correspond with large variations in pixel intensity, they are found as local maxima of the function E(u,v). This requires that the second term is maximized, which can be rewritten using a Taylor expansion:

$$E(u,v) \approx \sum_{x,y} w(x,y)[I(x,y) + uI_x + vI_y - I(x,y)]^2 \approx \sum_{x,y} w(x,y)[u^2 I_x^2 + 2uvI_xI_y + v^2 I_y^2]$$

where I_x and I_y are the image derivatives in x- and y-directions respectively. By expanding the equation, this can be written in matrix form as follows:

$$E(u,v) \approx \begin{bmatrix} u & v \end{bmatrix} \left(\sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} \right) \begin{bmatrix} u \\ v \end{bmatrix}$$

which can be formulated as:

$$E(u,v) \approx \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix} \tag{5.2}$$

The matrix M is the structure tensor of the image evaluated in pixel (x,y) and also known as the Harris matrix. Because corners are characterized by a large variation of E in all directions of vector (u,v), a corner characterization can be defined by analyzing the eigenvalues of M. Based on the magnitude of the eigenvalues, λ_1 and λ_2, criteria can be derived to decide whether or not a region can be considered as a corner, edge or a flat region. These three cases can be summarized in a classification matrix given in Figure 5.1.

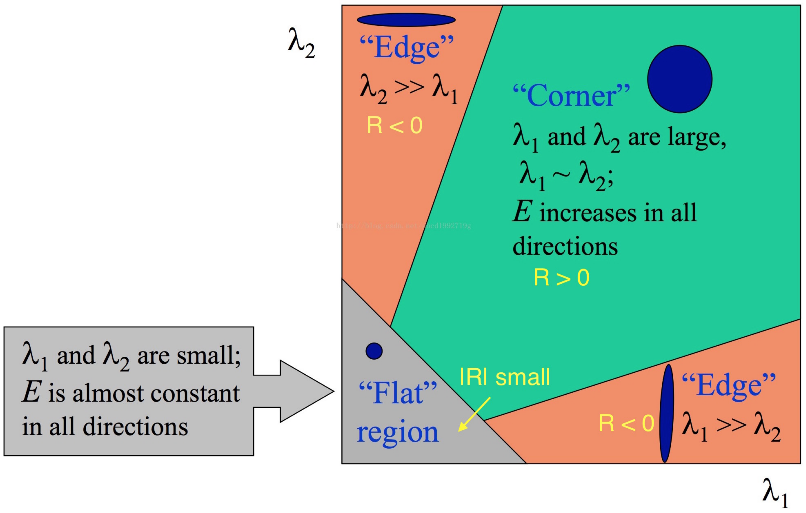


Figure 5.1: Classification of tested image point according to Harris-Stephens corner detector. Adapted from Gilbert (2014).

On this basis, Harris and Stephens (1988) suggest a scoring function for every patch/window as they noted that the computation of the eigenvalue decomposition of M is computationally expensive. To reduce the computational cost, they propose the scoring function R:

$$R = det(M) - \kappa\,(trace(M))^2 \tag{5.3}$$

for which it can be proven that:

$$det(M) = \lambda_1\lambda_2$$

$$trace(M) = \lambda_1 + \lambda_2$$

where κ is a tunable sensitivity parameter. A value of κ needs to be determined empirically, although in literature values ranging from 0.04-0.15 are suggested. The three cases that are graphically represented in Figure 5.1 can then be formulated as follows:

- $|R|$ small: This corresponds with small values for λ_1 and λ_2, indicating a flat region.
- R < 0: This corresponds with $\lambda_1 >> \lambda_2$ or vice versa, indicating the region is an edge.
- R large: This corresponds with λ_1 and λ_2 both large and $\lambda_1 \approx \lambda_2$, indicating the region as a corner.

Because this measure only requires the evaluation of the determinant and trace of M, it allows a much faster implementation. Additionally, Tuytelaars and Mikolajczyk (2008) prove that this approach also detects L-junctions and points with a high curvature as corner locations. The Harris Corner detector is invariant to translation, rotation and illumination change (Tuytelaars and Mikolajczyk (2008)) but is on the other hand not invariant to large scale changes (Schmid et al. (2000)). To solve this issue, Tuytelaars and Mikolajczyk (2008) suggest a modification, known as the Harris-Laplace or Harris-Affine detector, which is both scale and affine invariant.

Shi and Tomasi (1994) suggests a small modification to the Harris corner detector, in which the scoring function is redefined:

$$R = min(\lambda_1,\lambda_2) \tag{5.4}$$

In case R is larger than a certain threshold value λ_{min}, the region corresponding to R can be considered as a corner. A graphical representation of this method is given by Figure 5.2.

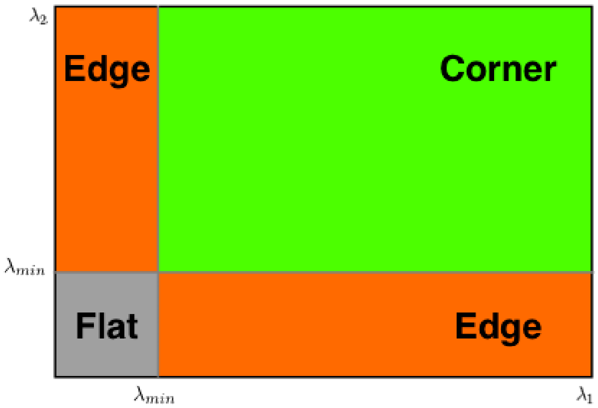


Figure 5.2: Classification of tested image point according to the Shi-Tomasi Corner Detector. Adapted from Itseez (2015).

In their paper, it is shown that under certain assumptions, this redefined measure function gives better results that the Harris corner detector. Additionally, the detected corners are found more stable for tracking. The method is also sometimes referred to as the Kanade-Tomasi corner detector.

Criticism to this corner detection method is that it proves quite sensitive to noise because it is a gradient-based detector. Any pixel affected by ‘salt noise’, i.e. sparsely occurring white and black pixels, will have a large gradient in all directions and can incorrectly be marked as a corner. Suggestions have been made to use a larger Gaussian window in which the gradient is computed. This makes the method more robust w.r.t. noise but also more computational demanding and adversely affects localization. Secondly, Parks and Gravel (2004) mentions that the detector shows poor localization for some junction-types, although the significance of this limitation is application dependent. Finally, it must be mentioned that the operator is not a through rotational invariant corner detector for every situation. Although the matrix M is rotationally invariant, the derivatives used in this method are only taken in the horizontal and vertical direction and therefore the corners detected vary with rotation. However, recent studies have shown that by taking a circular (weighted) window, e.g. a Gaussian window, the response becomes isotropic.

5.1.3.3 The Förstner corner detector

In the following sections, we discuss methods that have been developed to obtain sub-pixel accuracy in the localization of the detected corners.

A first, contour-based method, finds sub-pixel accurate corners based on boundary curvatures. Such type of corner detection algorithms are therefore also denoted as curvature-based corner detectors. This algorithm, developed by Föstner and Gülch (1987), is used to determine an approximate least-square solution in which the point located most closely to all tangent lines of the ‘corner’ in a given window is found (Figure 5.3). In case of an ideal corner, the tangent lines to the local boundary of an element in the image would cross at a single point. In reality, corners appear somewhat fuzzy and a single intersection point of the tangent lines is difficult to determine. The expression for the tangent line through pixel **x’** is given by:

$$T_{\mathbf{x}'}(\mathbf{x}) = \nabla I(\mathbf{x}')^\top(\mathbf{x} - \mathbf{x}') = 0 \tag{5.5}$$

where $\nabla I(\mathbf{x}') = [I_{\mathbf{x}}, I_{\mathbf{y}}]^\top$ is the gradient vector of the image I at $\mathbf{x}'$. In case a sharp transition exists between an edge and the rest of the image, this vector is perpendicular for every point on that edge. The method then finds the point $\mathbf{x}_0$, located closest to all tangent lines in a local search window N around an initial estimate of the corner location. A least-square solution for $\mathbf{x}_0$ is searched by using following expression:

$$\mathbf{x}_0 = \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^{2\times 2}} \int_{\mathbf{x}'\in N} T_{\mathbf{x}'}(\mathbf{x})^2 d\mathbf{x}' \tag{5.6}$$

This integral is weighted by the gradient magnitude in every point **x’**. The weighted sum of squared distances from **x’** to all tangent lines in window N can be written in full as follows:

$$\begin{aligned}
\mathbf{x}_0 &= \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^{2\times 2}} \int_{\mathbf{x}'\in N} (\nabla I(\mathbf{x}')^\top(\mathbf{x} - \mathbf{x}'))^2 d\mathbf{x}' \\
&= \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^{2\times 2}} \int_{\mathbf{x}'\in N} (\mathbf{x} - \mathbf{x}')^\top \nabla I(\mathbf{x}') \nabla I(\mathbf{x}')^\top (\mathbf{x} - \mathbf{x}') d\mathbf{x}' \\
&= \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^{2\times 2}} (\mathbf{x}^\top A\mathbf{x} - 2\mathbf{x}^\top \mathbf{b} + c)
\end{aligned}$$

with:

$$A = \int \nabla I(\mathbf{x}')\nabla I(\mathbf{x}')^\top d\mathbf{x}' \in \mathbb{R}^{2\times 2} \tag{5.7}$$

$$\mathbf{b} = \int \nabla I(\mathbf{x}')\nabla I(\mathbf{x}')^\top \mathbf{x}' d\mathbf{x}' \in \mathbb{R}^{2\times 1} \tag{5.8}$$

$$c = \int \mathbf{x}'^\top \nabla I(\mathbf{x}')\nabla I(\mathbf{x}')^\top \mathbf{x}' d\mathbf{x}' \in \mathbb{R} \tag{5.9}$$

The first derivative w.r.t. x is set to zero to find a local minimum:

$$2A\mathbf{x} - 2\mathbf{b} = 0 \Rightarrow A\mathbf{x} = \mathbf{b} \Rightarrow x_0 = A^{-1}\mathbf{b} \tag{5.10}$$

The matrix A is again the structure tensor, sometimes also denoted as the window image second moment matrix. In case the rank of this matrix is 2, A is invertible and a solution of the problem can be found.

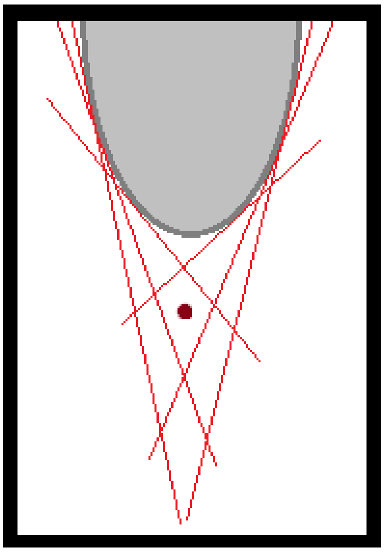


Figure 5.3: The Förstner corner detector: the corner location is determined with sub-pixel accuracy as the intersection point of the tangent lines. From Patel and Panchal (2014).

5.1.3.4 Other curvature-based corner detectors

Lindeberg (1998) present an automatic scale selection for the the corner detection method explained above. This makes the corner detection method more robust to image noise as stated by Patel and Panchal (2014).

Wang and Brady (1995) suggest a detector by considering the total curvature of a gray-level image. This is considered proportional to the second order directional derivative in the direction tangential to the edge normal or inversely proportional to the edge strength. It therefore finds places where the curvature along an image edge is large and the edge direction suddenly changes. They suggest a corner score C, which is the sum of the cornerness measurement and a second false corner suppression term:

$$C = \nabla^2 I - c|\nabla I|^2, \tag{5.11}$$

where the parameter c can be chosen to determine how edge-phobic the detector is. Secondly, they propose smoothening to reduce image noise and a sub-pixel addressing mechanism.

The curvature-based corner detectors mentioned in the previous two sections are however sensitive to noise. Moreover, they have a small application domain because they can only be used to detect X, Y and T junctions. For this reason, other corner detection algorithms were developed that also offer sub-pixel accuracy.

5.1.3.5 SUSAN corner detector

Smith and Brady (1997) suggest an intensity-based corner detector, in which they use a circular mask that is placed on the pixel to be tested (nucleus). Every pixel within the mask is compared to the nucleus by applying a comparison function that considers the brightness of the compared pixels. The area within this mask which has the same (or similar) brightness as the nucleus is defined as the 'USAN', an acronym standing for 'Univalue Segment Assimilating Nucleus'. This circular mask with the corresponding USAN is shown in red in Figure 5.4, located at different positions of a black rectangle. In their paper, a corner detection algorithm is suggested based on the center of gravity and the area of the USAN.

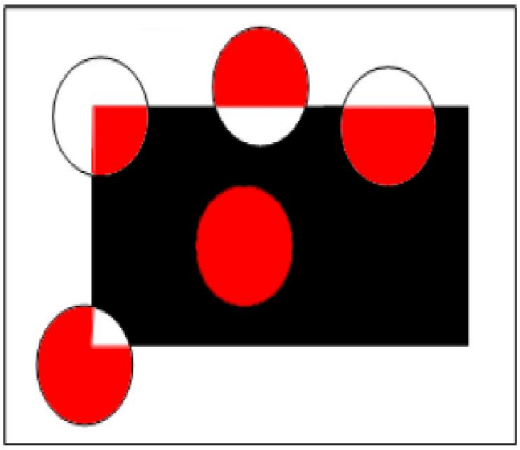


Figure 5.4: USAN for different circular masks on a uniform rectangle. From Patel and Panchal (2014).

The advantage of this method is that the SUSAN edge and corner detector uses no image derivatives which gives it a good performance in the presence of noise. The method allows in fact strong noise rejection due to the integration that is used in combination with its non-linear response. Additionally, the controlling parameters are more simple and less arbitrary and therefore easier to automate than most other methods.

However, it is noticed that this type of detector generates corners on the lines of the grid and additional steps are needed to detect actual corners.

5.1.3.6 The Trajkovic and Hedley corner detector

Trajković and Hedley (1998) suggest a detector which is based on SUSAN, by checking the self-similarity of a pixel by evaluating the intensity at surrounding pixels. They assume that a corner can be defined as a location for which the change in image intensity must be high in all directions. As such, a corner response function (CRF) is developed by considering an arbitrary line containing the nucleus and intersecting the boundary P of a circular window at two opposite points p and p' (Figure 5.5). The response function is subsequently defined as:

$$r(\vec{c}) = \min_{\vec{p} \in P} \quad (I(\vec{p}) - I(\vec{c}))^2 + (I(\vec{p'}) - I(\vec{c}))^2 \tag{5.12}$$

where $\vec{c}$ is the central point that is tested.

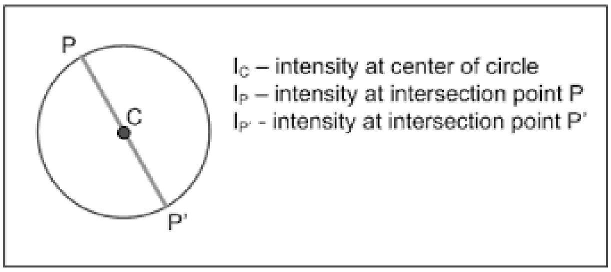


Figure 5.5: Test circle for Trajkovic operator. From Kumar and Nagappan (2012).

In case there exists no direction for which the tested pixel is similar to two nearby pixel along a diameter, this function will be large. The tested circle P is discretized (a Bresenham circle), and a minimum of the CRF on the circle is found by iteration. Corners are subsequently found as local maxima of the CRF function.

In order to obtain sub-pixel accuracy, the intensity at inter-pixel locations is computed by linear combination of the corresponding endpoint intensities. This is however computationally intensive and therefore a three-step algorithm is used to find the corners. This involves detecting potential corners on a low resolution image, which are subsequently refined using an inter-pixel approximation on the full resolution image. Finally, an algorithm named non-maximum suppression (NMS) is used to find pixels with a local maximum in the CRF. Because in the vicinity of a corner more than one point will have a high CRF, the one with the largest CRF is considered as corner point.

5.1.3.7 AST-based feature detectors

AST (accelerated segment test) detectors are a type of relaxed SUSAN detectors, in which the pixels on a Bresenham circle of radius r around the tested pixel are considered. The tested pixel (or nucleus) is marked as a corner in case n contiguous pixels are all brighter or darker than the nucleus by at least threshold t. By optimizing the order in which the pixels are tested, this results in a very fast corner detector. The detected features are in that case also very stable, as stated by Rosten and Drummond (2006).

Rosten and Drummond (2006) derive one of the mostly used methods based on this principle, named FAST (features from accelerated segment test). They suggest a value of r=3 (corresponding to a Bresenham circle with a circumference of 16 pixels) , which results in very good results in case n = (minimum) 9. The disadvantage of FAST is that it needs to be augmented with pyramid schemes for scaling invariance and combined with a Harris detector in order to reject edges as because it does not provide a cornerness strength measure.

5.1.3.8 Local sub-pixel refinement methods

Several methods have been developed to refine initially detected corners by other methods and as such obtain sub-pixel accuracy. Sroba et al. (2015) studies several of these methods w.r.t. their performance.

Weixing et al. (2009) propose an improved SUSAN algorithm that is used to detect chessboard corners. They suggest the use of the USAN's regions geometry structure (see Section 5.1.3.5) to differentiate corners from edges. Subsequently, the corner positions are found with sub-pixel accuracy based on the orthogonal vectors theory (Liang et al. (2006)). They develop a corner detector based on the principle that a vector from the corner (denoted as q) to any adjacent area (denoted as p) should be perpendicular to the image gradient at point p (see Figure 5.6). Based on linear algebra, the dot product between both vectors can be used to quantify the angle between them:

$$\epsilon_i = \nabla I_{p_i}^T \cdot (q - p_i) \tag{5.13}$$

where ∇I_{p_i} stands for the image gradient at one of the points p_i in the neighborhood of q. In order to find a corner, the pixel coordinate q that minimizes the expression should be found. In case ϵ_i is set to zero, following system of equations can be found:

$$\sum_i (\nabla I_{p_i} \cdot \nabla I_{p_i}^T) \cdot q - \sum_i (\nabla I_{p_i} \cdot \nabla I_{p_i}{}^T \cdot p_i) = 0 \tag{5.14}$$

where the gradients are summed within a local neighborhood ('search window') around q. By denoting the first term as A and the second term as b, the set of equations can be rewritten as:

$$q = A^{-1} \cdot b \tag{5.15}$$

As can be seen, the same result as in Section 5.1.3.3 is eventually obtained. The point position q, found as the solution of (5.15), is then taken as new center of the search window after which the whole process is repeated. The algorithm iterates until the center stays within a certain set threshold.

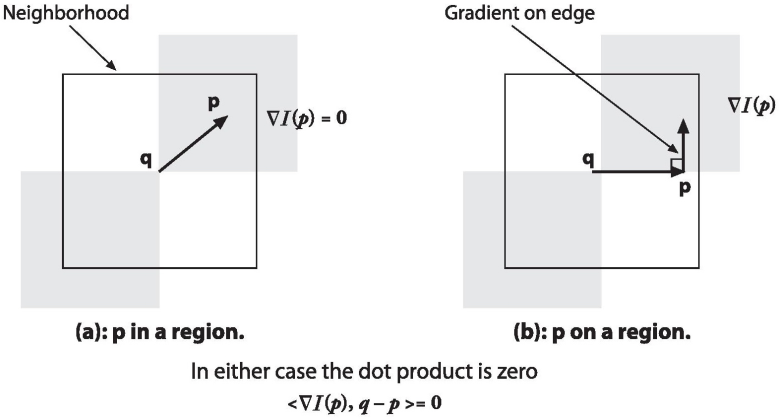


Figure 5.6: Principle of sub-pixel corner detection according to the method proposed by Weixing et al. (2009). From Bradski and Kaehler (2008).

A second method proposed by Lucchese and Mitra (2002) refines the corner positions to sub-pixel accuracy by fitting a quadratic curve to the brightness intensity profile in both the x- and y-direction. This function describes a hyperbolic paraboloid in three dimensions, whose saddle points correspond with the required corner locations. The explicit equation of this surface can be written as:

$$h(x,y) = ax^2 + bxy + cy^2 + dx + ey + f \tag{5.16}$$

This equation has six unknowns, and as such six brightness values in a chosen search window are necessary to construct a system of six equations. This system can be solved with e.g. the least squares method. In case the coefficients $a, \ldots, f$ are calculated, the sub-pixel position is then computed by taking the first derivative of this function:

$$\begin{bmatrix} x & y \end{bmatrix} = - \begin{bmatrix} 2a & b \\ b & 2c \end{bmatrix}^{-1} \cdot \begin{bmatrix} d & e \end{bmatrix} \tag{5.17}$$

A third approach to obtain sub-pixel positions of the corner point is based on a second order Taylor polynomial (Chen and Zhang (2005)). The polynomial describes the local intensity profile around the tested corner point. A refinement of the initial position is obtained by shifting the corner over a distance ϵ_x and ϵ_y from the initial position. In case one assumes that the corner point corresponds with a saddle point in the intensity profile, setting the first order partial derivatives of the polynomial to zero allows to determine a more accurate position. The necessary shifts are computed using the first- and second-order derivatives (differences) in the x- and y-direction:

$$\epsilon_x = \frac{I_y I_{xy} - I_x I_{yy}}{I_{xx} I_{yy} - I_{xy}^2} \tag{5.18}$$

$$\epsilon_y = \frac{I_x I_{xy} - I_y I_{xx}}{I_{xx} I_{yy} - I_{xy}^2} \tag{5.19}$$

5.1.3.9 Fuzzy systems

Fuzzy systems are developed in order to efficiently handle impreciseness and incompleteness. Due to the imaging process (defocussing, variations in illumination, etc.), noisy and imprecise images can be obtained. A few fuzzy approaches to detect corners have been developed. Banerjee and Kundu (2008) propose an algorithm based on the fuzzy edge strength and gradient direction. Corners are obtained by comparing a certain threshold with the fuzzy edge map. Várkonyi-Kóczy (2008) develop a fuzzy system in which a local structure matrix is used and a continuous transient between localized and not-localized corners takes place. Although these methods are robust, they are very computationally expensive compared with other classical corner detection algorithms.

A recent robust corner detection algorithm employs a rule-based fuzzy system (Cuevas et al. (2011)). Fuzzy systems use a rectangular $N \times N$ window (usually 3×3 pixels), in which the gray level of the central pixel is compared with its neighbors. The results are stored in the matrix E:

$$E = \begin{bmatrix} P_{m,n} - P_{m-1,n-1} & P_{m,n} - P_{m-1,n} & P_{m,n} - P_{m-1,n+1} \\ P_{m,n} - P_{m,n-1} & 0 & P_{m,n} - P_{m,n+1} \\ P_{m,n} - P_{m+1,n-1} & P_{m,n} - P_{m+1,n} & P_{m,n} - P_{m+1,n+1} \end{bmatrix} \tag{5.20}$$

where m and n represent the coordinated of the central pixel. For homogeneous regions, the gray level in the neighborhood of the central pixel will be (almost) equal to the central pixel. The matrix E will in that case contain values near zero. For corners on the other hand, the matrix E will possess a specific configuration which is dependent on the corner type. The region around the central pixel can then be divided in two connected regions, one with positive (pixel type A) and one with negative (pixel type B) difference values.

Next, two different types of rules, namely **THEN-rules** and **ELSE-rules**, are used to identify corners but at the same time also cancel other inconsistencies such as noise. For each configuration, a set of rules is defined to determine if the tested pixel is a corner or not. Additionally, the procedure handles noisy pixels by using a reduced set of rules (configurations). The proposed corner detector therefore applies twelve THEN-rules to detect twelve possible corner configurations, combined with one ELSE-rule to discard every pixel that is not a corner. The different corner cases are graphically represented in Figure 5.7.

Each rule relies on the principle that the pixel can be considered as a corner in case the structure in E, defined by a certain configuration of A and B pixels, corresponds with one of the twelve THEN-rules. A threshold value t_h is chosen to evaluate these difference values, allowing to control the sensitivity of the method. Cuevas et al. (2011) suggest a value $t_h = 20$. This

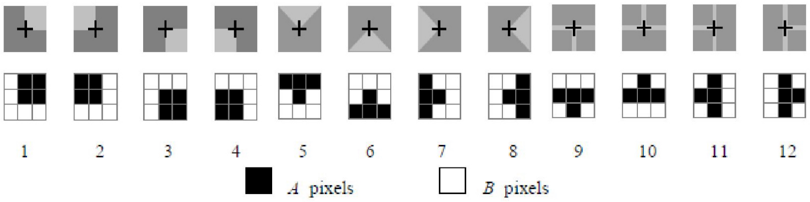


Figure 5.7: Different corner cases to be considered for building the fuzzy rules. The image region containing the corner is shown in the upper section while the resulting 3x3 template is shown below each case. From Patel and Panchal (2014).

threshold t_h is then used to compute two auxiliary matrices E^p and E^n for every pixel and every corner configuration. Next, they compute membership values $\mu_c(m,n)$ (where c=1,2,...12) as the antecedent of each employed THEN-rule. The 12 fuzzy rules are finally used to compute one final fuzzy value, being the maximum of the twelve $\mu_c(m,n)$'s.

The proposed technique was tested on several benchmark images and compared with well-known conventional corner detectors. The results show that the method is able to tolerate implicit imprecision as well as impulsive noise. Additionally, the computational cost for analyzing the benchmark images seemed much lower than other fuzzy methods, allowing a shorter processing time.

5.2 Conclusion and choice of the implemented feature detection algorithm

5.2.1 Assessment of early implementations

Feature points **f** should be easily detectable in the different images. In Morris (2004) for example, a chessboard pattern is placed on the bottom of the tank. This monotone checkered pattern offers hard edges and distinct corners, which are distinguishable from their surroundings.

Template matching is with this in mind advisable: for every corner, a combination of two white and two black squares of the chessboard pattern can then be located in the image. These templates are in that case locally specific and will not match against any of the four nearest corners. The neighbors of a corner in the chessboard pattern are surrounded by a reversed combination of black and white checkers, which makes them easily distinguishable from the corner that needs to be detected (see Figure 5.8). This facilitates feature detection and minimizes the risk that a wrong corner is identified.

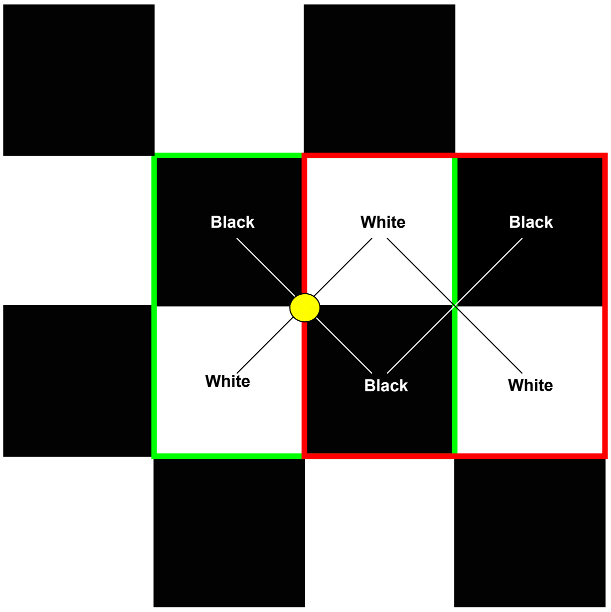


Figure 5.8: Because of the reciprocal black and white checkers, a neighboring corner of a chessboard pattern is characterized by a reversed 2 × 2 combination of checkers. This makes the pattern locally specific and avoids that a corner is identified as one of its neighbors.

Based on the overview given Section 5.1, the template-based fuzzy systems seem the most promising method for the localization of these corner points. It has been proven that both the accuracy and robustness of these methods is significantly better than the traditional corner detection algorithms. Because of the limited time available in this thesis, a self-written implementation of such a feature detector was not possible. We therefore searched for a preprogrammed algorithm, which could be directly incorporated in the global reconstruction procedure. As will be explained in Chapter 6, the OpenCV library written in C++ was adopted because several of the methods mentioned in the previous section are implemented in OpenCV. Template matching is however still not fully optimized in the OpenCV library which made this difficult to incorporate. OpenCV offers in contrast several alternative solutions which will be explained below.

Firstly, a function *findChessboardCorners* is available in the OpenCV library. This function attempts to identify a chessboard pattern in the image, given the width and height of the chessboard pattern to be found. To facilitate detection of the corner points, the function uses its knowledge about the relative positions of the corners w.r.t. each other. In case the entire chessboard pattern is detected, the function returns the individual pixel coordinates of the internal chessboard corners.

An advantage of this algorithm is that the corner points are listed in systematical order, based on the number of rows and columns of the pattern. This eliminates the need for further sorting and allows to detect the feature points in a fast and user-friendly way. For this reason, we will use this function during camera calibration, as will be explained in Chapter 6. The distortion of the projected feature pattern on the water surface made feature localization with *findChessboardCorners* however more difficult. Additionally, the function failed to locate the chessboard pattern in case a large amount of corners needs to be found. This method could therefore not be used in the reconstruction algorithm.

Secondly, the function *goodFeaturesToTrack* was adopted which is based on two of the mostly used corner detection algorithms. This function calculates a corner quality measure at every pixel of the source image, in which both the Harris corner measure and the minimum eigenvalue measure of Shi and Tomasi (1994) can be applied. The method computes the second order derivatives automatically (using the Sobel operators) and using these results, the eigenvalues are determined. These eigenvalues finally allow to locate the pixels in the image for which the so-called cornerness strength is larger than a predefined threshold value. The final result is a list of corners, ordered according to their cornerness strength.

The corner locations can then be refined to determine the location with sub-pixel accuracy. OpenCV offers a C++ implemented version of the algorithm suggested by Weixing et al. (2009), named *cornerSubPix*, that refines a list of initial estimates of the corner positions.

The accuracy and robustness of the combination of these two methods mentioned above seemed however insufficient to use in our reconstruction algorithm. Accurate localization of image points **q** and **q'** has an importance influence on the performance of our algorithm because these determine the direction of the incident light rays $\overrightarrow{\boldsymbol{u}}$ as was explained in Chapter 4. Incorrect feature localization would therefore make an accurate reconstruction of the water surface impossible. Additionally, the adopted OpenCV algorithms appeared not robust when tested during early experiments. The corner detection algorithms implemented in the OpenCV library did not always locate every corner of the pattern and sometimes found a feature in the middle between two corners.

Because the previous two approaches did not perform well during initial tests, an alternative solution was searched that could be used during this thesis. After experimenting with several possible feature grids, it was decided to use a feature pattern that could be detected with non-corner based detection algorithms.

5.2.2 Adopted feature pattern and detection algorithm

Our finally implemented feature detection algorithm is not designed to locate corners. Due to lacking performance of the OpenCV corner algorithms, we adopted a pattern of dots as feature plane F. To that end, we use the OpenCV function *SimpleBlobDetector* that allows to locate ‘blobs’ (dots) in an image. Several parameters can be set to filter out the type of blobs that need to be detected. In our implementation, the blobs are located based on their color, minimal size and minimal distance from each other. The function allows finally to replace the detected but sometimes irregular blobs (due to thresholding of the image) by points located in the center of the detected blobs.

A well-thought choice for the density of the pattern must be made because this has an important influence on the performance of the reconstruction algorithm. A dense pattern results in a larger risk of feature elimination and separation of features, which occurs due to the refraction of these features in case of opposing normals.

'Elimination of feature points' means that the feature point becomes invisible to the camera due to the refraction limits. This is possible in case a local strong curvature of the water surface is combined with a camera that observes the water surface at a sharp angle. 'Separation of feature points' occurs when for two adjacent feature points on F the image points appear separated due to refraction. Additionally, a dense pattern might degrade the robustness in case template matching would be implemented in a further improvement of the algorithm. A too dense pattern could then lead to problems related to localization as the support region for the corners becomes smaller.

On the other hand, the pattern must be dense enough in order to have sufficient points to determine the entire water surface shape with reasonable accuracy. Because our surface function is fitted to the discrete surface points $\mathbf{p}$, a denser grid allows to reconstruct local shape effects with a higher accuracy.

The finally adopted feature pattern is depicted in Figure 5.9 and consists of circular dots with a diameter of 0.1 mm, located on a regular spacing of 10 mm from each other. Because the chosen feature points cannot be distinguished from each other after replacing them by markings at their centers of gravity using *SimpleBlobDetector*, sufficient distance between the features is necessary. In case of a large displacement of a tracked feature point on the water surface between two successive frames, the software could identify one of the neighboring feature points as its new location. With a combination of the chosen spacing and a fast frame rate of the camera, this problem is eliminated. Tracking of the feature points will be explained in detail in Chapter 8.

Figure 5.9: Adopted feature pattern during this thesis, consisting of regularly spaced dots with a diameter of 0.1 mm and a spacing of 10 mm.

Chapter 6

Detailed description of the developed method

6.1 Software used in the implementation of the algorithm

6.1.1 OpenCV

OpenCV (Open Source Computer Vision Library) is an open-source BSD-licensed library developed by Bradski (2000) that includes several hundreds of computer vision and machine learning algorithms (Bradski and Kaehler (2008)). The library is cross-platform and mainly written in C++ (although an older C interface exists) but contains bindings for other languages such as Python or JAVA. Some of the application areas include 2D and 3D feature toolkits, camera calibration, facial recognition systems, object identification and motion tracking, ...

In this thesis, the functions available from the OpenCV library will be used for several aspects of the water reconstruction procedure:

- Camera calibration (estimation of the distortion coefficients).
- Camera pose estimation (estimation of the camera position and orientation w.r.t. a predefined reference system).
- Feature and corner detection.
- Feature tracking (based on optical flow methods).

6.1.2 ImageJ

ImageJ is an open source JAVA-based image processing program for multidimensional image data with a focus on scientific imaging (Schneider et al. (2012) and Schindelin et al. (2015)). It has an open architecture which makes it extensible by using JAVA plugins. Custom acquisition, analysis and processing plugins can be developed using ImageJ's built in editor and Java compiler. A lot of these user-written plugins are free-available online.

During this thesis, the Fiji-distribution of ImageJ will be used to process the images before they are used in the surface reconstruction algorithm (Schindelin et al. (2012)). ImageJ provides some useful functions and plugins which make the detection of the feature points in the images much easier:

- Loading, renaming and converting images in bulk.

- Selecting and masking a part of the image. This allows to ‘cut out’ the grid in the image and set the rest of the image to a constant background color.
- Splitting of the color channels of the image (Red, Green and Blue: RGB).
- Enhancing the local contrast in the image.
- Hardening edges in the image.
- Removing noise in the image.
- Applying intensity-based thresholds: retaining only a specific range of the pixel intensity, and setting everything below or above this range to black or white respectively.
- . . .

Because it was noticed that the OpenCV functions sometimes have difficulties in detecting the feature points in the image, some of the ImageJ plugins are adopted to improve the quality of the input images. Random noise (see Chapter 7) is filtered out and accidental small particles in the water are removed. The finally obtained images contain a white background in which the feature points appear as black spots. In Figure 6.1a and Figure 6.1b, the same image before and after post-processing is depicted. This comparison shows that the contrast between the feature points and the background is much better after post-processing, which obviously facilitates feature detection.

6.2 Detailed description of the experimental setup

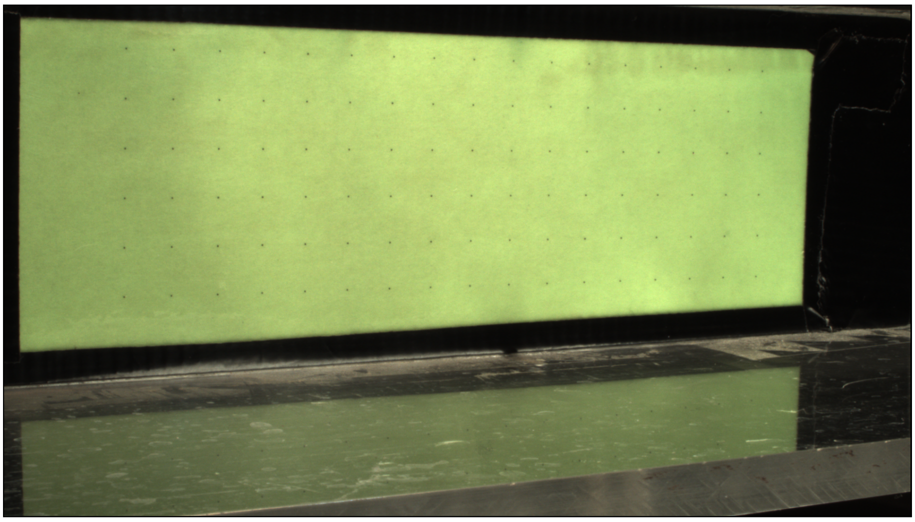
6.2.1 Imaging Equipment

6.2.1.1 General

The method developed within this thesis is image-based. This makes the technique easily implementable and does not require high-technological end expensive equipment such as lasers for example. It also offers the possibility to increase the accuracy of the reconstruction procedure by adopting more advanced imaging equipment if this would be required in further research.

Images of the water surface are taken with Basler ace GigE Vision cameras (Figure 6.2). Because these digital cameras are relatively small, they are easy to install in the small-scale test setup and easy to reposition if required. The cameras are area scan cameras containing a CCD sensor and are equipped with a wide angle, high-resolution Basler lens with a fixed focal length of 8 mm. They are able to capture both mono (grayscale) and color images. Their maximum frame rate is dependent on the image size and image format, limited by the maximum bandwidth of the data link (see further). The maximum resolution of a single image that can be reached is 1920 × 1080 pixels. In case the maximum resolution is used, this allows to capture the largest possible area in the scene and to obtain the most accurate details in the resulting image.

The cameras are equipped with a Gigabit Ethernet (GigE) cable. This data link transfers the captured images to store them on a nearby PC. It also offers a Power over Ethernet (PoE) functionality, which allows to supply the cameras with electricity using the same data cable. However, early experiments showed that the maximum bandwidth of such cables is the limiting factor in our test setup. A combination of the maximum resolution and maximum frame rate is therefore not possible and a compromise between both has to be made, depending on what needs to be captured in the scene.



(a) Before processing

(b) After processing

Figure 6.1: Image before and after processing in imageJ

Figure 6.2: Area scan Basler ace GigE camera, equipped with a Basler lens with focal length of 8 mm

6.2.1.2 Camera settings

a. Frame rate vs image resolution

The validation and error assessment (Chapter 7) is based on images of still water. Because in that case the position of the feature points does not change, motion blur or large movements of the image points are not an issue. For these tests, the maximal resolution is therefore used with a relatively small frame rate and shutter speed.

In the reconstruction of dynamically changing water surfaces though (Chapter 8), the feature points show significant displacements between successive frames. A faster frame rate is therefore necessary to obtain an accurate temporal description of the surface shape and to facilitate tracking of the feature points (Chapter 8). To that end, the size of a single image needs to be reduced in order not to exceed the maximum bandwidth of the data link. Three possible solutions for this problem are explained below.

Firstly, the image can simply be reduced in size by adjusting the acquisition window width. The camera then only transmits the detected electronic charges of the cells ('photosites') located in a specified rectangular zone of the CCD sensor. Although the accuracy and sharpness in the image remain the same, a smaller part of the 3D scene is in that case depicted in the final image. This makes the possible spatial extent of the reconstructed surface smaller.

A second possibility is to apply binning. CCD camera binning consists of combining information of neighboring detectors on the CCD sensor as input for a single pixel in the recorded image. When using 2×2 binning for example, the electronic charges from an array of four detectors is used to determine a single pixel intensity-value. This increases the contrast and image intensity of the finally obtained image. It also significantly improves the signal-to-noise ratio (SNR) of the camera. The spatial resolution is however decreased (for 2×2 binning by a factor of 2 in both the horizontal and vertical direction). The distance in the image (in pixels) between two separated points in the scene is in that case reduced by two compared to the full-resolution image. Our method is however based on the image disparity shifts (expressed in pixels) of the image points **q'**, which are used to compute the direction of the viewing rays $\overrightarrow{u}$. For small surface changes, the magnitude of the resulting image disparity shifts then becomes smaller compared to the inaccuracies related to feature detection. Binning therefore limits the accuracy of the detection algorithm and is preferably avoided for our application.

In Chapter 7, our method will be validated in experimental tests. In order to obtain optimal performance of the algorithm, the images of those tests are taken without binning. In Chapter 8, a dynamically changing surface will be reconstructed in which motion blur becomes a risk. Due to the limited bandwidth of the data cable, the higher required frame rate makes binning unavoidable.

A third possibility is to use monochrome (grayscale) images. Their reduced size (in bytes) allows a higher frame rate but monochrome images also have one major disadvantage, which will be explained in the next paragraph.

b. Image format: monochrome or color images

As was explained in Chapter 4, the wavelength of the viewing rays affects their change in direction due to refraction at the air-water interface. In order to take this into account, it is advisable to consider each color channel of the captured images separately. The digital Basler cameras are able to produce different image formats. Firstly, monochrome images are possible in which the different color channels are combined in a weighted sum to obtain grayscale images. The wavelength-dependent variation of the refractive index can then not be incorporated which makes monochrome images less suitable for our application.

In case of color images, two distinctive formats are possible: YUV and Bayer GB. The YUV color space consists of three components: a luminance component (Y) and two chrominance components (UV). The conversion from RGB to YUV is however also based on combining the RGB values as a different weighted sum for every component.

With the purpose of keeping the three color channels separate, we therefore use the Bayer GB-format. In order to split the three primary colors (red, green and blue) received from the scene, the camera sensor is equipped with a Bayer filter. This filter is constructed as an array of red, green and blue filters. Every type only transmits one particular color as shown in Figure 6.3. A combination of overlapping 2×2 arrays is then used to determine the color for every pixel in the final image.

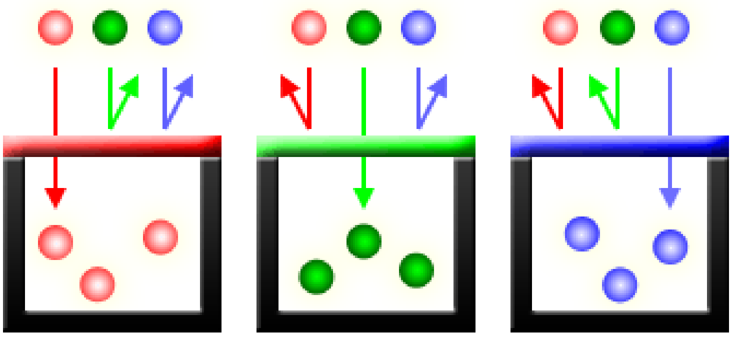


Figure 6.3: Photosites with color filters. From McHugh (2016).

The human eye is more sensitive to green light because it is more important in nature to distinguish small changes of green compared to the other two colors. In view of providing more contrast in the green channel, a Bayer array contains twice as many green filters as red or blue filters which is shown in Figure 6.4. The green channel shows therefore much finer details in the scene and also a lower sensitivity to noise (McHugh (2016)).

Using the Bayer GB images, a representative refractive index for every channel can be computed based on the average wavelength in that band of the light spectrum. In this first implementation of the algorithm, we simplified the reconstruction by only using the high-quality green channel with a corresponding refractive index of 1.336 according to Harvey et al. (1998).

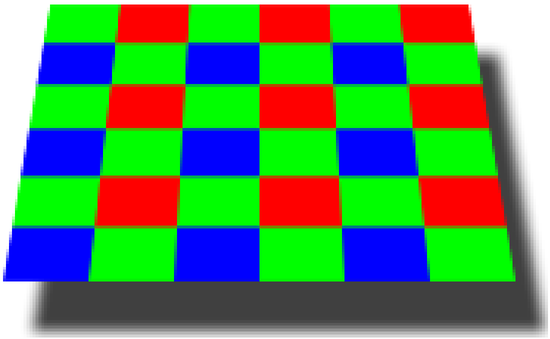


Figure 6.4: Bayer Array with reciprocal green-red and green-blue filter combinations. From McHugh (2016).

6.2.2 Lighting conditions

In order to obtain maximum contrast and high-quality images, sufficient light from the scene is required. Our initial lighting setup consists of several spots placed around the test tank, as shown in Figure 6.5. The lighting spots are directed towards the feature pattern on the bottom of the tank to make that area as bright as possible. In this way, the distinction between the feature pattern and the background in plane F is maximized.

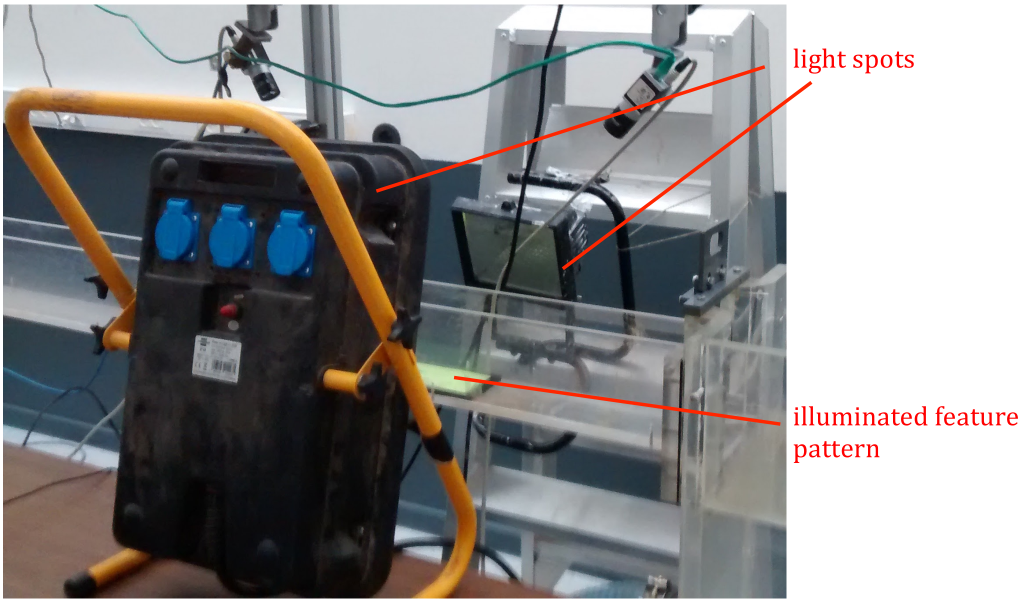


Figure 6.5: First possible lighting setup, consisting of regular lighting spots placed around the tank.

The brightness of the images remains however limited due to two reasons:

1 **Position of the lighting equipment above the water surface.**
The lighting equipment is ideally placed as close as possible to the captured pattern on plane F. In practice, a minimum distance between the lighting equipment and the tank needs to be retained. Because the cameras observe the water surface from above, lighting equipment located just above the pattern could hinder the view of the cameras. More importantly, light spots located close to the water surface could result in specular artifacts in the image. These local highlights could negatively influence feature detection.

2 **Minimal shutter speed of the camera.**
A smaller shutter speed results in more light on the image sensor and therefore brighter images. A minimal shutter speed is however required because of two reasons:

- Motion blur.
  Motion blur occurs in case the captured objects in the scene (in this case the surface points $\mathbf{p}$) move while the shutter of the camera is open. The image points of $\mathbf{p}$ are in that case blurry, making it impossible to locate them accurately.

- Image noise.
  Small shutter speeds also increase the amount of noise in the image, as they make the camera's sensor overheat faster. Noise makes feature localization more difficult and should therefore be avoided as much as possible. A more detailed explanation of noise and how to avoid it is given in Chapter 7.

It can be concluded that a trade-off between obtaining bright but on the other hand high-quality images is required. We therefore adapted the shutter speed according to the amount of light available in the scene.

Additionally, the lighting conditions across the feature pattern should be as uniform as possible.

A uniform brightness in the image facilitates the post-processing process and minimizes the risk that part of a feature point is lost due to thresholding (see Chapter 7).

In a second attempt to obtain images in which the feature points are more distinguishable from the background, a lighting setup with ultraviolet (UV) light was adopted. The feature pattern was printed on fluorescent paper and the traditional spots were replaced by an UV light tube ('black light') of 30 W as shown in Figure 6.6. UV offers the advantage that this part of the spectrum is undetectable for the cameras which eliminates the risk of specular highlights on the water surface. The UV lamp can therefore be placed as close as possible to the bottom of the tank, without the risk of bright reflections on the water surface. The lighting setup which is shown in Figure 6.6 could therefore still be optimized by lowering the UV spot closer to the illuminated fluorescent paper.

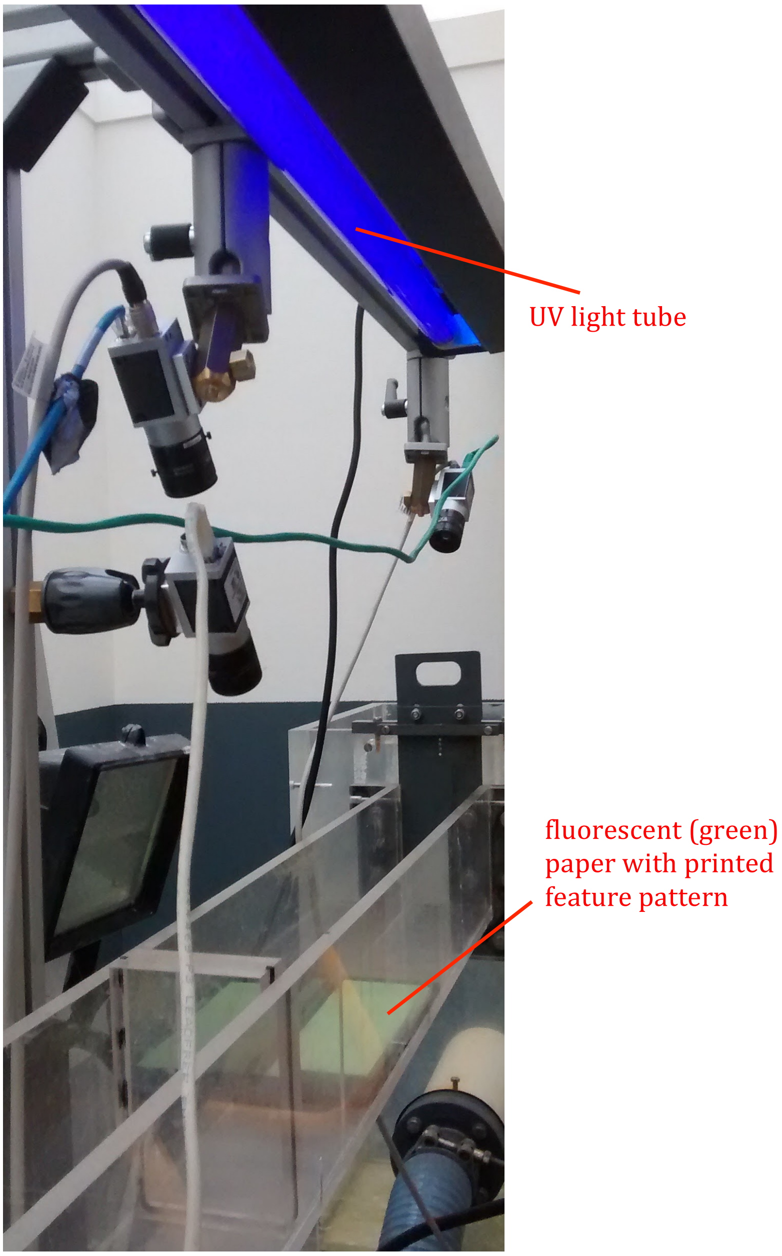


Figure 6.6: Lighting setup based on UV light. The feature pattern is printed on fluorescent paper that emits visible light in case it is illuminated by UV light.

The fluorescent paper absorbs this UV light and transforms the energy of the UV light rays into light within the visible range. The paper therefore 'glows' and serves as an apparent light source for the camera. The ink of the printed feature pattern locally blocks the emitted light and the feature points appear as dark spots in the image. Because absolutely no light is emitted at these locations, the contrast between the feature points and the rest of the paper is in this way maximized. Additionally, the room in which the test-tank is placed was darkened to increase the effect of fluorescence of the paper.

In case green paper is chosen, most of the emitted light is contained in the green channel of the camera. As explained in the previous section, the chosen cameras are much more sensitive to green light due to the Bayer array of the sensor. Choosing green paper allows to maximize the image quality and therefore accuracy of our image-based technique. A comparison between the performance of the algorithm with visible or UV light is given in Chapter 7.

6.2.3 Test tank

In order to study the self-induced sloshing phenomenon in navigation locks, a small-scale model of a navigation lock was build. The 'lock chamber' has a width of 8 cm and a height of 10 cm. The design incorporates a dividing wall to partition the entire test tank, which can be repositioned in case this is required. This allows to vary the length of the lock chamber in order to study the influence of this parameter in further stages of the research. The lock gates are simplified by vertical valves that can be lifted over a predefined distance to simulate the filling process of the lock chamber. Figure 6.7 depicts this scale model in which the most important parts are indicated.

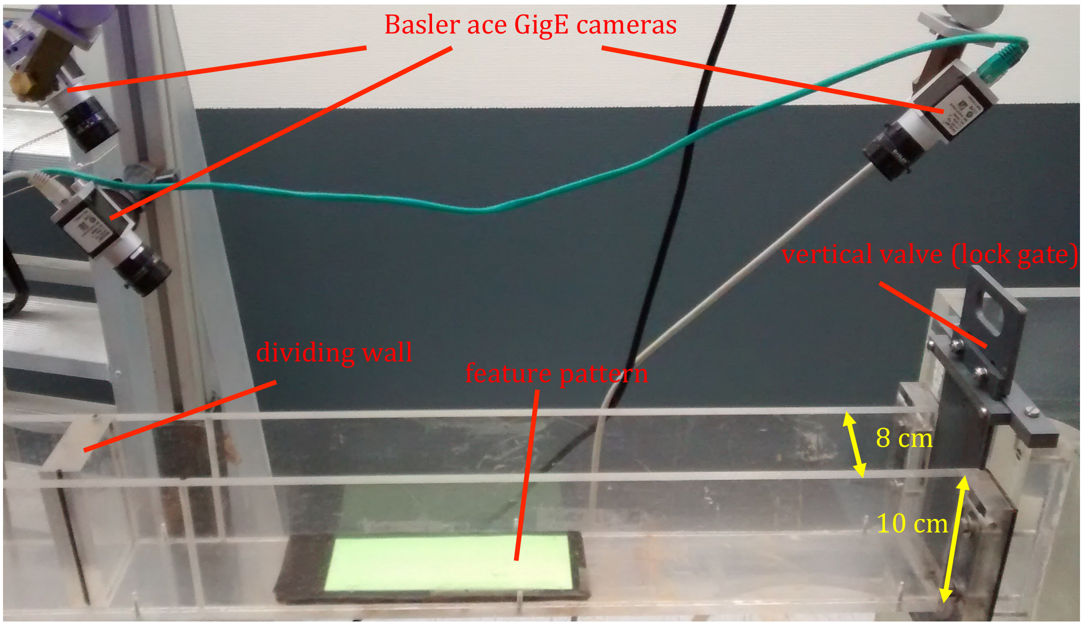


Figure 6.7: Illustration of the test tank and positioning of the cameras above the tank.

6.3 From pixel to world coordinates

6.3.1 Theoretical transformation from pixels to physical points in the scene

6.3.1.1 Distortion of the images

A first step in the reconstruction of the water surface consists of correcting the images for lens distortion. Ideally the images obtained from the video file or image sequence would result in a rectilinear projection, in which straight lines in a scene remain straight in the image. In reality lens distortion however always occurs. This form of optical aberration requires a correction of the images to obtain the undistorted pixel coordinates of the detected feature points.

Using OpenCV, the distortion coefficients according to Brown's distortion model, also known as the Brown-Conrady model (Brown (1966)), can be determined. This model takes both radial as tangential distortion into account. Radial distortion results in a radially symmetric distortion pattern, originating from the symmetry of a photographic lens. Although several types of radial distortion exist, the lens used in the experiments suffers from barrel distortion which is shown in Figure 6.8. In case of barrel distortion, the image magnification decreases as the distance with the optical axis decreases. This type of radial distortion usually occurs in wide angle lenses and zoom lenses.

Tangential distortion occurs because the image capturing lenses are not aligned with the image plane. This type of aberration is composed of decentering distortion and thin prism distortion. Decentering distortion arises due to the fact that the centers of lens elements are not perfectly collinear. Secondly, thin prism distortion originates from imperfection in lens design and manufacturing, as well as from camera assembly (e.g. slight tilt of some lens elements or the image sensing array).

In case every factor is taken into account, the relationship between the distorted and undistorted images points is given by following formula (Zhang (2000)):

$$\begin{aligned} x_d =& x_u \frac{1+k_1r^2+k_2r^4+k_3r^6}{1+k_4r^2+k_5r^4+k_6r^6} + 2p_1\, x_u\, y_u + p_2(r^2+2x_u^2) + s_1\, r^2 + s_2\, r^4 \\ y_d =& y_u \frac{1+k_1r^2+k_2r^4+k_3r^6}{1+k_4r^2+k_5r^4+k_6r^6} + p_1(r^2+2y_u^2) + 2p_2\, x_u\, y_u + s_3\, r^2 + s_4\, r^4 \end{aligned}$$

where:

(x_d,y_d) The distorted image point as projected on image plane using a specific lens.

It must be noted that these are not the same as the pixel coordinates obtained from the image. The relationship between the (distorted) pixel coordinates (u,v) and 2D coordinates (x_d,y_d) in the image plane is given by following equation, as will be explained in Section 6.3:

$$\begin{aligned} u &= f_x \cdot x_d + c_x \\ v &= f_y \cdot y_d + c_y \end{aligned}$$

(x_u,y_u) The undistorted image point as projected by an ideal pin-hole camera:

$$\begin{aligned} u &= f_x \cdot x_u + c_x \\ v &= f_y \cdot y_u + c_y \end{aligned}$$

k_n	The n^{th} radial distortion coefficient.
p_n	The n^{th} tangential distortion coefficient.
s_n	The n^{th} thin prism distortion coefficient.
r	$= \sqrt{x_u^2 + y_u^2}$ in case it is assumed that the distortion center is equal to the principal point.

Finally, the image sensor may be slightly tilted. This tilt causes a perspective distortion of x_d and y_d which can also be incorporated. This type of distortion can be modeled using the Scheimpflug camera model:

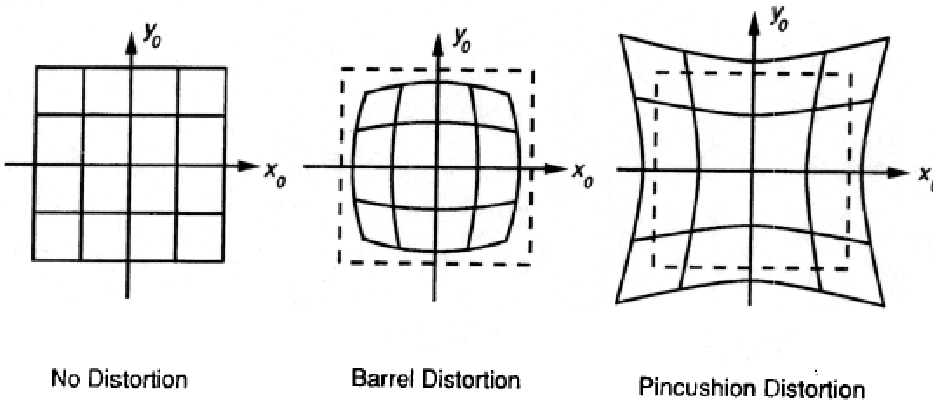
$$\begin{bmatrix} x'_d \\ y'_d \\ 1 \end{bmatrix} = \begin{bmatrix} R_{33}(\tau_x,\tau_y) & 0 & -R_{13}(\tau_x,\tau_y) \\ 0 & R_{33}(\tau_x,\tau_y) & -R_{23}(\tau_x,\tau_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix}$$

and taking these extra parameters into account:

$$u = f_x \cdot x'_d + c_x$$
$$v = f_y \cdot y'_d + c_y$$

where the matrix $R(\tau_x,\tau_y)$ is defined by two rotations with angular parameters τ_x and τ_y. A detailed comparison between the Scheimpflug model and the traditional pinhole camera is conducted in Zhang and Zhou (2015).

Because the most higher order terms as well as tilt of the image sensor are mostly negligible, the distortion coefficients k_1, k_2, k_3, p_1, p_2 are usually sufficient to undistort the images obtained from the camera. It was however observed that the accuracy of our reconstruction procedure is dependent on the distortion model that is used. Provided that a more extensive model allows a more accurate description of the distortion of the lens, the coefficients k_4, k_5, k_6 are also incorporated in the camera calibration to obtain a more accurate result. A more detailed elaboration of the influence of the adopted distortion model on the reconstruction accuracy is given in Chapter 7.



No Distortion Barrel Distortion Pincushion Distortion

Figure 6.8: Types of radial distortion. From Niemann (2006).

6.3.1.2 Intrinsic (or internal) parameters

The location of the feature points must be transformed from pixel positions to physical coordinates. The camera is considered as a pinhole camera, schematically represented in Figure 6.9. The sensor of the camera is made up a of a grid of discrete sensor elements (pixels), which are ordered from top to bottom and from left to right. The location of a point on this image plane is given by its pixel coordinates (u,v). The pixel-indexes start at (0,0) in the top left corner and increase going downwards or to the right of the image. In order to transform pixel positions into real physical locations in the scene, both a scaling and a translation has to be performed (Hartley and Zisserman (2003); Itseez (2015)).

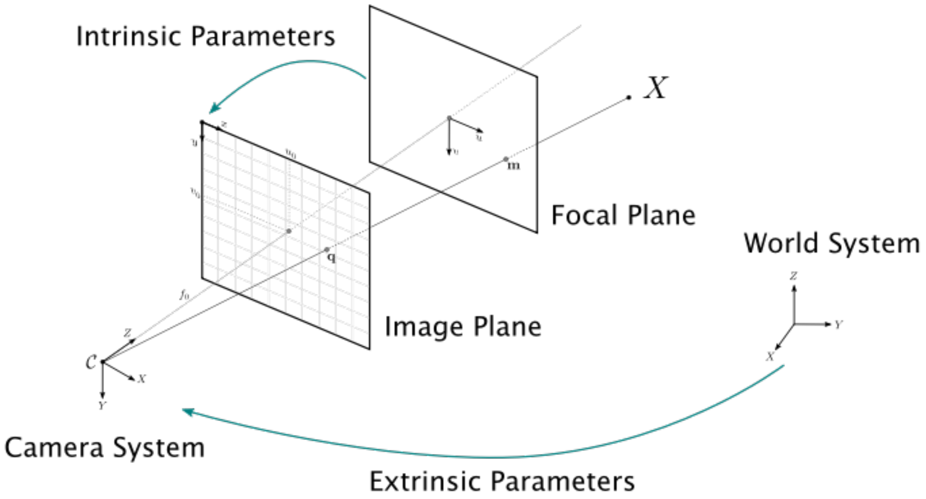


Figure 6.9: Schematic representation of pinhole camera and the transformation from pixel to world coordinates. From OpenMVG authors (2015).

A coordinate system in units of millimeters is defined, in which the origin is located at the principal point. This point marks the intersection of the optical axis with the image plane. The x- and y-axes of this Cartesian coordinate system are parallel with the u- and v-axes respectively. A scaling of the pixel coordinates is done by using the scale factors s_x and s_y. These represent the number of pixels in the horizontal and vertical direction per mm. For a through pinhole camera, s_x and s_y are the same. In reality however, these parameters can be slightly different due to several reasons as mentioned by Kyle (2013):

- Flaws in the digital camera sensor.
- Non-uniform scaling during post-processing.
- Unintentional distortion introduced by the camera's lens.
- An anamorphic format used by the camera, where the lens compresses a wide-screen scene into a standardized sensor.
- Errors in the camera calibration.

Some papers suggest a different approach in which a single focal length and an additional parameter named the 'aspect ratio' is adopted. This aspect ratio, defined as $\alpha = \frac{s_y}{s_x}$, describes the amount of deviation from a perfectly square pixel. Because the aspect ratio is not exactly equal to 1, the pixels are therefore not perfect squares. The resulting camera matrix (see further) obtained by both approaches remains however the same.

Secondly a translation is necessary because the origin of the physical coordinate system (principal point (c_x,c_y)) does not coincide with the top left corner of the image. Next a skew coefficient s_{xy} between the x- and the y-axis can be incorporated. Given that this parameter is always approximately zero, s_{xy} can normally be neglected and is therefore assumed equal to zero during this thesis.

Finally, the 2D coordinates in the image plane (x,y) need to be transformed to 3D coordinates. Our chosen 3D coordinate system has an x-and y-axis parallel with the 2D-case but the origin is located in the so-called center of projection O. The distance between the image plane and the center of projection O is denoted as the focal length f. This camera parameter is by definition measured in pixels and equal to the physical distance between the lens and the CCD array. The relationship between the 2D and 3D coordinates is given by the equations of perspective projection, in which the 2D points on the image plane are expressed in homogeneous coordinates:

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Taking into account the other intrinsic parameters, following relationship between the pixel coordinates and the position of the 3D coordinates in the camera reference frame is obtained:

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f \cdot s_x & s_{xy} & c_x \\ 0 & f \cdot s_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{6.1}$$

and with $s_{xy} \approx 0$:

$$\begin{bmatrix} f \cdot s_x & 0 & c_x \\ 0 & f \cdot s_y & c_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} = K \tag{6.2}$$

The matrix $K \in \mathbb{R}^{3\times3}$ is known as the intrinsic matrix or camera calibration matrix, and contains the 4 (or 5) intrinsic parameters.

6.3.1.3 Extrinsic (or external) parameters

In order to express the position of points (X,Y,Z) in a world coordinate system, independent of the camera, the extrinsic parameters of the camera have to be known. These can be used to apply a coordinate transformation between the camera's coordinate system and the world coordinate system.

We define a Cartesian world reference frame, with the origin in the upper left corner of the feature pattern. The horizontal x- and y-axes are chosen along the borders of the pattern and the z-axis normal to the bottom of the tank (positive upwards). The relationship between the points in the world coordinate system and the camera's coordinate system is given by:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} \tag{6.3}$$

or

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + T \tag{6.4}$$

in which $R \in \mathbb{R}^{3\times 3}$ is called the rotation matrix and $T \in \mathbb{R}^{3\times 1}$ the translation vector. This equation is often written in homogeneous coordinates:

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_3 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{6.5}$$

The joint rotation-translation matrix [R|t] is called the matrix of extrinsic parameters. Combining both the extrinsic and intrinsic parameters of the camera, the relationship between the (undistorted) pixel coordinates and coordinates of a point in the world reference frame is given by:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K(R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + T) \tag{6.6}$$

The entire transformation from pixel coordinates to world coordinates for a pinhole camera is visually represented in Figure 6.9. The equation is often written in homogeneous coordinates:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_3 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{6.7}$$

or

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = P \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{6.8}$$

where $P \in \mathbb{R}^{3\times 4}$ is called the projection matrix.

6.3.2 Practical implementation using OpenCV

6.3.2.1 Calibration: determination of the intrinsic camera parameters

The distortion coefficients of the camera are found together with its intrinsic parameters by using a set of calibration images. These images depict a chessboard-pattern under varying positions and orientations w.r.t. the camera. After loading in the calibration images, the chessboard pattern on every image is found with the function *findChessboardCorners*, which gives the pixel coordinates of the detected corners as result. The positions of these corners are subsequently refined with *cornerSubPix* to obtain sub-pixel accuracy. This list of corner coordinates is then compared with a list of physical corner coordinates. The theoretical location of these discrete points in the chessboard pattern is determined based on a coordinate system with the origin in the upper left corner of the chessboard pattern and both axes parallel with the chessboard borders. Based on the square size of the adopted calibration pattern, these physical coordinates can easily be computed.

These two lists of corner positions (in pixel-coordinates and physical coordinates (in mm's)) are subsequently used as input of the OpenCV function *calibrateCamera*. This optimized algorithm allows to determine the camera matrix and distortion coefficients, together with the rotational and translation vectors for every image. The algorithm is based on the work of Zhang (2000) and Bouguet (2014). A global Levenberg-Marquardt optimization algorithm is used to minimize the total reprojection error. This error is defined as the total sum of the squared distances between the observed corners in the image and the projections of the physical corner points in the image plane, using the current estimates of the camera parameters. The intrinsic parameters of the camera are finally stored in a .yml-file. The quality of the camera calibration is subsequently checked by evaluating the overall mean reprojection error. In case it would be observed that this error is too large, leaving out the images that result in a large reprojection error or even repeating the entire calibration is necessary to obtain a better calibration result.

A sufficient amount of calibration images is required, in which the chessboard pattern is viewed under different angles. In case too few images or images in which the camera position is not varied are used, a bad calibration result could be obtained. Because this has a negative influence on the final accuracy of the water surface reconstruction, this should be avoided as much as possible. In Zhang (2000), the error on the calibration is investigated with respect to the number of images. In this paper, they show that the calibration quality increases drastically for an initial increase of the number of images but does not improve significantly after 17 images as shown in Figure 6.10. Because points are never detected perfectly, too much images would even just add ‘noise’ to the calibration. Adding much more images to the training set would therefore result in lower calibration quality. For this reason, a total amount of twenty images is used in the camera calibration, which is also suggested by Bouguet (2014).

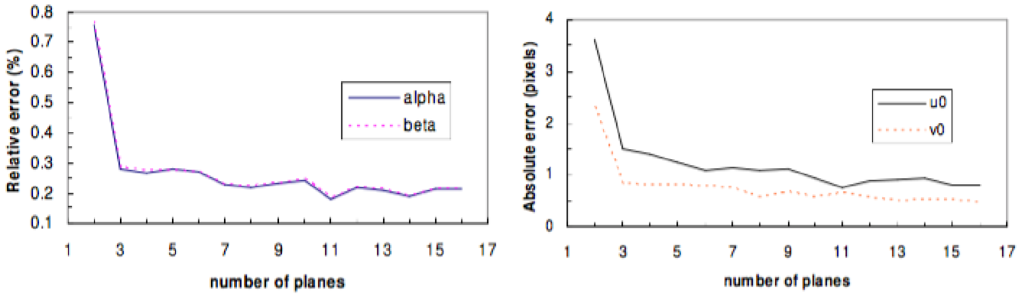


Figure 6.10: Error on calibration vs. the number of images of the model plane (chessboard pattern). From Zhang (2000).

A third important aspect of the camera calibration is the location of the chessboard in the calibration pictures. A sufficient amount of corner points should be located close to the edges and corners of the image to get a better estimate of the distortion coefficients. Because lens distortion is most pronounced in the corners of an image, the correction of the pixel coordinates is locally more important than in the center of the image. A correct distortion model can however only be determined in case enough information about the distortion in those regions of the image is provided.

Finally, the quality of the calibration images should be as good as possible. The calibration images are taken with the camera parameters (focal length, aperture size, ...) that will be used during capturing of the reconstructed image sequence. In order to facilitate feature detection, the camera is focused to capture the feature pattern as sharp as possible. Scenes that are located at a significantly different distance to the camera will as such appear blurry for those specific camera settings. For this reason, the calibration images should be taken with the position of the calibration pattern around the same depth range as the pattern during the measurements. This allows to obtain sharp calibration images in which the chessboard corners can unambiguously be located.

6.3.2.2 Initialization: determination of the extrinsic camera parameters

In the method described in Chapter 4, the entire geometric problem is expressed in an arbitrary but fixed coordinate system. Because we use multiple cameras in the practical implementation of the algorithm, it is advisable to define a coordinate system independently of the cameras used. As described in Section 6.3.1, the transformation between coordinates in a camera-fixed reference frame to 'world-coordinates' requires the extrinsic camera parameters to be known (stored in matrix R and vector T). A pose estimation of the cameras, i.e. position and orientation w.r.t. the physical grid, is therefore needed.

1. Solving the PnP-problem

Several methods exist to solve the pose estimation problem or non-linear Perspective-n-Point problem (PnP problem). All strategies are based on matching the observed image points and the physical coordinates of these points in the scene. The correspondence between both sets of points can finally be used to estimate the camera position and orientation w.r.t. to the object (in this case planar grid). In OpenCV, several of the most popular approaches have been implemented. One of the older methods that has been derived by Dementhon and Davis (1995) is implemented in OpenCV as *cvPosit*. The method assumes the scale factor s_x and s_y to be equal so only considers one focal length $f = f_x = f_y$. Although this is approximately true, this would result in an additional source of inaccuracy as already mentioned in Section 6.3.1.1. Additionally, this method fails for co-planar objects points and can therefore not be used.

A second and more recent OpenCV function *solvePnP* is therefore adopted. This function allows to estimate the camera pose given a set of object points, their corresponding image projections, as well as the camera matrix and the distortion coefficients. The implemented algorithm accepts several flags that allow us to determine which specific method has to be applied:

1. `SOLVEPNP_ITERATIVE`.
   An iterative method which incorporates a Levenberg-Marquardt optimization to find the pose that minimizes the total reprojection error, similarly as discussed in Section 6.3.2.1.

2. `SOLVEPNP_EPNP`.
   This method is based on the paper of Lepetit et al. (2009), in which they propose a non-iterative solution to the PnP-problem.

3. `SOLVEPNP_P3P`.
   This method, based on the paper of Gao et al. (2003), uses exactly four object and image points to solve the pose estimation problem.

4. `SOLVEPNP_DLS`.
   In Hesch and Roumeliotis (2011), a nonlinear least-squares cost function is proposed whose optimal conditions constitute a system of three third-order polynomials. The roots of this system are subsequently determined to find the minima of this cost function to obtain the position and orientation of the camera.

5. `SOLVEPNP_UPNP`.
   This method uses the approach derived by Penate-Sanchez et al. (2013), which is based on `EPNP` but in which the focal length is taken as an additional unknown.

The resulting rotation matrix and translation vector obtained with these different methods were compared to determine which method gives the best results. We determined the intersection points of the viewing rays from the camera's center through the observed image points with the horizontal plane at height zero, earlier defined as plane F. The physical coordinates of these intersection points, expressed in the world coordinate reference system, can be calculated by substitution of Z=0 in eq. (6.6):

$$s\begin{bmatrix}u\\v\\1\end{bmatrix}=K\left(R\begin{bmatrix}X\\Y\\0\end{bmatrix}+T\right)\tag{6.9}$$

In case the pixel coordinates that were used for pose estimation are substituted in eq. (6.9), the intersection points should in theory match the imposed object locations of the measurement grid. Because the camera pose estimation is however never perfect, these intersection points will deviate from the theoretical positions of the grid. The average of the Euclidean distances between the theoretical locations and projected image points on F is taken as quality measure for the pose estimation. This comparison showed that the first method is the far most accurate choice. Because the pose estimation has to be done only once during the initialization of the reconstruction algorithm, computational speed is not important to consider. Although some other methods allow a faster pose estimation, the iterative approach was therefore implemented.

2. Sorting of the feature points

The camera pose estimation requires a sorted list of pixel coordinates of feature points in the image plane. We therefore apply the function *SimpleBlobDetector* to detect the feature points in the post-processed images of grid without water in the tank.

Because the order in which the feature points are returned is initially unknown, the list of detected feature points needs to be rearranged. To determine the extrinsic camera parameters, these corners need to be sorted in a regular pattern that can be compared with the physical coordinates of the feature points expressed with respect to a world reference system. The origin of this reference system is located in the top left corner of the pattern and the x- and y-axes are defined parallel with the outer borders of the pattern. Their orientation is chosen in such a way that a right-handed Cartesian coordinate system is obtained with the z-axis perpendicular to the tank bottom and pointing upwards. Figure 6.11 depicts the feature grid in which the x-, y- and z-axis are indicated.

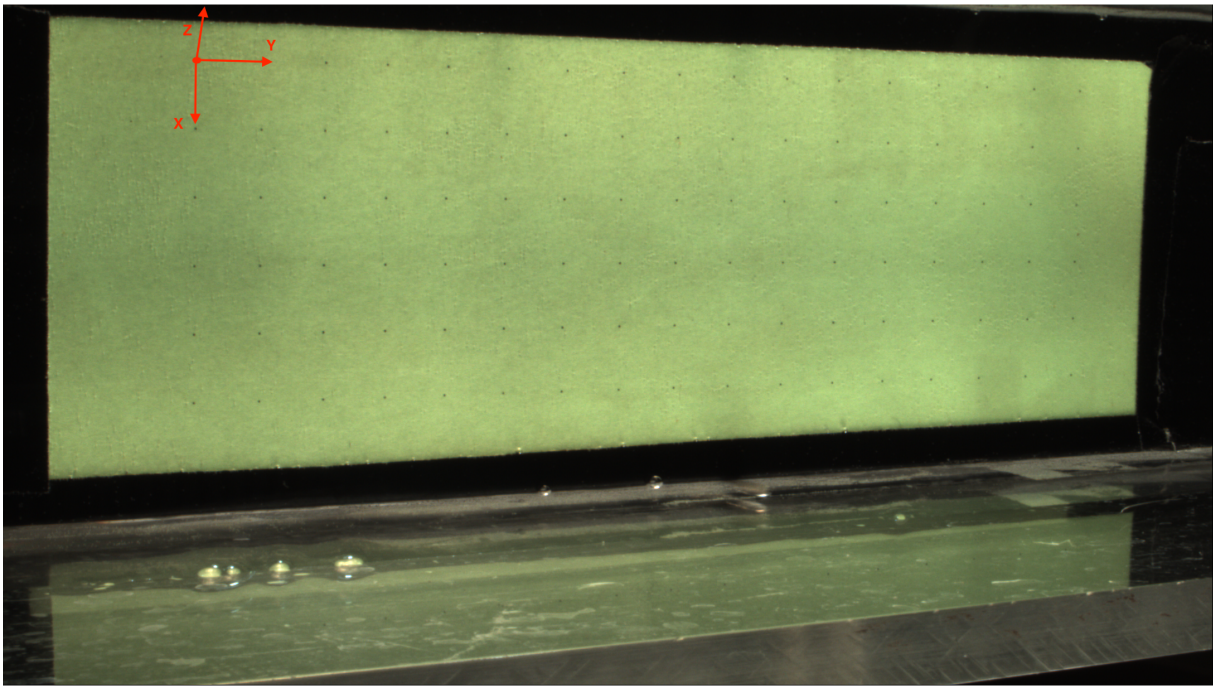


Figure 6.11: Feature pattern with our chosen world reference system

The theoretical corner positions are in that case a series of points with a spacing s equal to the physical distance between neighboring points (e.g. (0,0,0), (0,s,0), (0,2s,0),...). We list the features points starting from the origin, in which first all feature points with x-coordinate 0 are listed according to increasing y-coordinate. The following rows are indexed similarly, in which the x-coordinate is stepwise increased. As such, a sorted list of feature locations is obtained. To allow our pose estimation, the detected image points must be listed in such a way that they match with their corresponding coordinates in the scene. We implemented a self-developed sorting algorithm that will briefly be explained below.

The images are taken such that the physical y-axis, projected in the image, has approximately the same direction as the u-axis (horizontally to the right) and the physical x-direction approximates the v-direction (vertically downwards). Because it is difficult to perfectly align the y-axis with the u-axis (and x-axis with v-axis), just comparing the u and v coordinates of the image points is not possible. We therefore first locate the four outer corner points in the pattern based on visual detection of the user. Given those four corner points, the origin of our coordinate system is considered as the image point located closest to the top left corner of the image.

These four outer corners of the pattern are subsequently used to derive a mathematical formulation of lines that are parallel with the borders of the grid running in the y-direction. The regular spacing between these lines is calculated based on the distance between the extreme corner points and the number of rows in the pattern. By calculating the distance of the image points to each line (in pixel coordinates), points corresponding to the same row can be discerned. The points located on the same row are finally sorted according to increasing u-coordinate. The result of this algorithm is a list of pixel coordinates, ordered in the same way as our list of scene coordinates. A step-wise overview of the algorithm is given in Section 6.4.4 Algorithm 1.

Because we use different cameras to visualize a larger area of the tank, it is important to relate the cameras to the same world coordinate system. To allow the cameras to be positioned on the opposite side of the tank, an additional input parameter in the algorithm was therefore implemented. This allows to pass the position of the cameras w.r.t. the grid, i.e. the longitudinal side of the tank at which the camera is positioned. For cameras that view the pattern from opposite sides, the feature pattern is imaged differently. The origin of our world reference frame (outer corner of the pattern) is in that case depicted in the upper left image corner for one camera and the lower right image corner for the other camera. We therefore reverse the theoretical corner positions for one of the two opposing camera viewpoints so that the image points of every camera, all sorted in the way as explained before, correspond with the same physical feature points.

6.4 Surface reconstruction

6.4.1 Obtaining an estimate of the water surface shape

After the initialization, the global reconstruction algorithm is followed: for every time frame (image) the coefficients of the surface shape-model are determined using a multivariate optimization algorithm which minimizes the total error over all the detected feature points as defined in Chapter 4. A certain set of coefficients of our surface function is hypothesized, which results in an initial estimation of the shape of the water surface. In the following, the notation $''$ corresponds with a parameter obtained using this hypothesized surface.

In case the observed image points are sorted similarly as in the initialization phase, this allows to determine the location of every surface point $\mathbf{p}''$. These points are the projections of the corresponding feature point $\mathbf{f}$ on the hypothesized water surface. The location of these points can be computed in our world reference frame by solving the three equations given by eq. (6.6) that describe all points located on the viewing ray through image point **q'** (u,v). The three unknown parameters s, X and Y are found for a given value Z=z$''$ as follows:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \left(R \begin{bmatrix} X \\ Y \\ z'' \end{bmatrix} + T \right) \tag{6.10}$$

The value of z$''$ corresponding with surface point $\mathbf{p}''$ can be calculated using expression (4.30) in combination with a hypothesized set of coefficients (A''_{00}, A''_{10}, A''_{01}, A''_{11}, A''_{20}, A''_{02}, B'' and C''):

$$\begin{aligned} z(x,y) = & A''_{00} + A''_{10}\, cos(\frac{\pi x}{L_x}) + A''_{01}\, cos(\frac{\pi y}{L_y}) + A''_{11}\, cos(\frac{\pi x}{L_x})\, cos(\frac{\pi y}{L_y}) + \\ & A''_{20}\, cos(\frac{2\pi x}{L_x}) + A''_{02}\, cos(\frac{2\pi y}{L_y}) + B''\frac{x}{L_x} + C''\frac{y}{L_y} \end{aligned} \tag{6.11}$$

In order to compute $\overrightarrow{n_1}$ at point $\mathbf{p}''$, eq. (4.27) requires knowledge about the incident and refracted light rays at point $\mathbf{p}''$. To obtain a mathematical formulation for this incident ray $\overrightarrow{u}$, the position of the camera center and image point **q'** in the image plane must be expressed w.r.t. the world reference frame. The world coordinates of the camera center can be obtained by solving following set of three equations:

$$\boldsymbol{c_{world}} = -R^{-1}T \tag{6.12}$$

The physical coordinates of the intersection of ray $\overrightarrow{u}$ with the image plane is computed as:

$$\boldsymbol{q_{world}} = R^{-1}(K^{-1}\boldsymbol{q'} - T) \tag{6.13}$$

Because the location of **f'** can be found by replacing z$''$ with 0 in eq. (6.10), eq. (4.21) allows to determine the direction of the incident light ray between the camera center and point $\mathbf{p}''$. The direction of $\overrightarrow{v}$ is then derived by using the location of feature point **f** that is projected in $\mathbf{p}''$. The normal direction $\overrightarrow{n_1}$ that accounts for the refraction of $\overrightarrow{u}$ towards **f** can then finally be calculated with eq. (4.27).

6.4.2 Verification of the hypothesized model: computing the error metric

To verify the hypothesized description of the water surface, the second normal set $\overrightarrow{n_2}$ obtained with eq. (4.32) is computed in the discrete (x,y)-positions of points $\mathbf{p}''$. Our verification algorithm comprises a comparison between corresponding normals $\overrightarrow{n_1}$ and $\overrightarrow{n_2}$. In case the normal collinearity metric E_{col} is used, the error related to one specific feature point is easily calculated as:

$$E_{col} = cos^{-1}(\overrightarrow{n_1} \cdot \overrightarrow{n_2}) \tag{6.14}$$

The second error metric that can be used in the verification algorithm is the disparity difference metric E_{disp}. This involves some additional steps to compute the error related to one specific surface point $\mathbf{p}''$. The input parameters in this verification are point $\mathbf{p}''$, **q'** in physical world coordinates and the normalized normal vector $\overrightarrow{n_2}$. The direction of vector $\hat{\boldsymbol{u}}$ remains unchanged because this is fixed by $\overrightarrow{cq'}$. We recall the angle θ_δ from (4.20), and compute $\theta_{\delta 2}$ based on Figure 6.12:

$$\theta_{\delta 2} = \theta_{i2} - \theta_{r2} \tag{6.15}$$

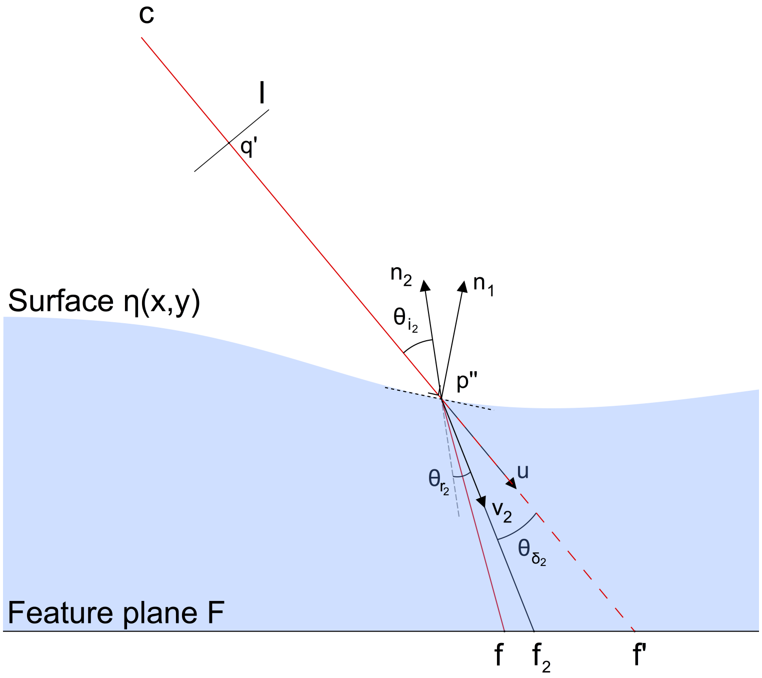


Figure 6.12: Illustration of refractive displacement in case $\overrightarrow{\boldsymbol{n}}_{\mathbf{1}}$ is swapped with $\overrightarrow{\boldsymbol{n}}_{\mathbf{2}}$. The refracted ray $\overrightarrow{\boldsymbol{v}}_{\mathbf{2}}$ according to $\overrightarrow{\boldsymbol{n}}_{\mathbf{2}}$ intersects the feature plane F at $\boldsymbol{f_2}$. Based on Morris (2004).

Because the vector $\hat{\boldsymbol{v}}_{\mathbf{2}}$ is still unknown, $\theta_{\delta 2}$ cannot be calculated with eq. (4.25). Using the two known vectors $\overrightarrow{\boldsymbol{n_2}}$ and $\hat{\boldsymbol{u}}$, the angle of the incident ray θ_{i2} is in contrast directly found by:

$$\theta_{i2} = cos^{-1}(-\hat{\boldsymbol{u}} \cdot \overrightarrow{\boldsymbol{n_2}}) \tag{6.16}$$

The angle $\theta_{\delta 2}$ can then be found using Snell's law:

$$sin(\theta_{i2}) = r_w sin(\theta_{r2})$$

and by substitution of (6.15) this becomes:

$$sin(\theta_{i2}) = r_w sin(\theta_{i2} - \theta_{\delta 2}))$$

$$\theta_{\delta 2} = \theta_{i2} - sin^{-1}(\frac{sin(\theta_{i2})}{r_w}) \tag{6.17}$$

This allows to calculate the vector $\hat{\boldsymbol{v}}_{\mathbf{2}}$ by rotating the vector $\hat{\boldsymbol{u}}$ over the angle $\theta_{\delta 2}$:

$$\hat{\boldsymbol{v}}_{\mathbf{2}} = R(\theta_{\delta 2}, \overrightarrow{\boldsymbol{n_2}} \times \hat{\boldsymbol{u}})\hat{\boldsymbol{u}} \tag{6.18}$$

The location of point $\boldsymbol{f_2}$ is subsequently found as the intersection of the ray from $\mathbf{p''}$ with direction $\hat{\boldsymbol{v}}_{\mathbf{2}}$ and plane F (defined by z-coordinate=0):

$$a = \frac{-\boldsymbol{p''_z}}{\hat{\boldsymbol{v}}_{\mathbf{2},\boldsymbol{z}}}$$

$$\boldsymbol{f_2} = \boldsymbol{p''} + a\,\hat{\boldsymbol{v}}_{\mathbf{2}}$$

The error associated with the refractive displacements of the feature points on plane F is finally computed for every feature point as:

$$E_{disp} = |\boldsymbol{f} - \boldsymbol{f_2}| \tag{6.19}$$

An overview of the entire verification algorithm (Algorithm 2) is given in Section 6.4.4.

6.4.3 Global optimization

The entire water surface is finally reconstructed by finding the coefficients in eq. (6.11) for which the dissimilarity between both normal sets is as small as possible. A multivariate optimization is done to minimize a global error function, expressed as the sum of squares of the errors related to every feature point:

$$E_{tot} = \sum_{f} E_f^2$$

where $E_f = E_{col}$ or E_{disp} depending on which error metric is adopted.

Because of the limited time available during this thesis, a direct implementation of a pre-programmed optimization algorithm was pursued. The presented method was initially programmed in the language and software environment **R**. **R** is used a lot for statistical computing and graphics because it is open source and provides a wide variety of statistical and graphical techniques/functions. The computational efficiency of this language seemed however insufficient for our reconstruction algorithm. The time necessary for the multivariate optimization of the coefficients of our surface function was very long, making it not suitable for practical use.

Since our developed **R** script was insufficient for practical applications, the entire reconstruction algorithm was rewritten in C++. This programming language allows to develop a stand-alone reconstruction script with a much higher computational efficiency. Both the detection of features as well as the optimization of the surface function are incorporated in the same program, making it much more user-friendly.

We make use of the C++ version of the ALGLIB library (ALGLIB (2016)), which includes an implementation of the Levenberg-Marquardt algorithm (LMA). This wide-known procedure for non-linear optimization combines the steepest descent or gradient method with the Newton-Raphson method. The first approach searches the minimum of a function in the direction of the gradient, which gives it a good operational stability. The second approach is an iterative method and uses a quadratic model to find a minimum faster. LMA combines both by finding a first approximation of the minimum with the steepest descent method, after which an accelerated convergence to the minimum is obtained with Newton's method. The algorithm finally stops its search in case one of the stopping conditions is reached:

- Gradient-based stopping condition: when the gradient norm is smaller than a certain threshold.
- Step-size-based stopping condition: when the step that is taken (change in one the coefficients) is smaller than a certain threshold.
- Function-based stopping condition: when the change in function value becomes smaller than a certain threshold.
- Iteration-based stopping condition: when the maximum number of iterations is reached.

Several possible optimization settings were tested. For the results given in Chapter 7 and Chapter 8, the second stopping criterion was chosen. The conducted iteration procedure uses a step-size equal 0.01 mm for every coefficient and continues until the required step size becomes smaller than 0.01 mm. The final result is a set of coefficients that best explain the refractive disparity of the entire feature pattern.

6.4.4 Overview of the algorithm used

Algorithm 1: sortingCorners

```
Input  : List of unsorted image points (pixel coordinates), order in which corners need
         to be converted
Output: List of sorted image point (pixel coordinates)
  1. Show the image in which every feature point is highlighted to the user
  2. Find the four features in the image located closest to the indicated points by the user.
     These four points are the edge points of the feature pattern.
  3. Define the edge point located closest to the top left of the image as origin and first point
     in the feature list.
  4. Compute difference in u- and v-coordinates between two edge points on the left of the
     grid and divide by necessary steps in v-direction (# rows in v-direction ($n_v$) -1):
         $step\,u_1 = \frac{\Delta u_1}{n_v - 1} \qquad stepv_1 = \frac{\Delta\, v_1}{n_v - 1}$
     Do the same for the two edge points on the right of the grid:
         $step\,u_2 = \frac{\Delta u_2}{n_v - 1} \qquad step\,v_2 = \frac{\Delta v_2}{n_v - 1}$
  5. for $i$ from $0 \rightarrow n_v - 1$ do
         (a) Compute two points on a line connecting the two x-borders:
                 $p_1 = origin + i \cdot [step\,u_1, step\,v_1] \qquad p_2 = edgePoint2 + i \cdot [step\,u_2, step\,v_2]$
         (b) Compute distance to this line for every point
         (c) Sort points according to distance from line
         (d) Append the first $n_u$ points (amount of columns)
     end
```

Algorithm 2: errorComputation

```
Input  : Hypothesized coefficients $A''_{00}, A''_{10}, A''_{01}, A''_{11}, A''_{20}, A''_{02}, B''$ and $C''$ , feature
         positions on plane F = $\mathbf{f}$, pixel coordinates of the the image points q', camera
         center of projection $\mathbf{c}$, refractive index of water $r_w$.
Output: Error $E_f$
  1. Compute $\mathbf{p}''$ from the water surface function with hypothesized coefficients by
     combination of eq. (6.10) and eq. (6.11)
  2. Compute $\overrightarrow{u}$ and $\overrightarrow{v}$ using eq. (4.21) - (4.23) or eq. (4.22) - (4.24) respectively
  3. Find the angle $\theta_\delta$ from eq. (4.25)
  4. Using a correct estimate of the refractive index $r_w$, compute $\theta_i$ with eq. (4.26)
  5. Based on the calculated vectors $\overrightarrow{u}$ and $\overrightarrow{v}$, compute the axis of rotation and
     corresponding rotation matrix R with eq. (4.28)
  6. Compute for every point $\mathbf{p}''$ the corresponding surface normal $\overrightarrow{n_1}$ with eq. (4.27)
  7. Compute the surface normal $\overrightarrow{n_2}$ based on the surface function evaluated at the (x,y)
     position of points $\mathbf{p}''$ using eq. (4.31) and eq. (4.32)
  8. Compute error $E_f$ from the normal collinearity metric or disparity difference metric
     using eq. (6.14) or eq. (6.19) respectively
  9. Return $E_f$
```

Chapter 7

Validation and error assessment of the measurement technique

In this chapter, the technique that was described in detail in Chapter 6 is validated using two approaches. First of all, experimental tests are done in which the reconstructed surfaces are compared with other possible measurement techniques. A dynamic changing water surface is however difficult to validate, as traditional measurement techniques only allow discrete, vertical changes in water level to be quantified. In order to have ground truth measurements, our experimental validation is based on the reconstruction of still and flat water surfaces.

Because the core of our reconstruction algorithm considers each time frame (image) separately, the difference between the reconstruction of still or dynamically changing water surfaces remains methodologically small. For both cases, the detected feature points in the image are used to fit a surface model that is chosen based on prior knowledge about the expected surface shape. More complex surface shapes are reconstructed similarly as a flat water surface by adjusting the fitted surface function accordingly.

Moreover, most sources of errors in the reconstruction (simplifications in the reconstruction methodology, optimization, camera calibration, feature detection, ...) are not related to the movement of the water surface. The limited amount of additional complications inherent to moving waters surfaces (i.e. motion blur and larger displacement of features points) can be minimized by optimizing the technical aspects of the recording, such as shorter shutter times, more illumination by stronger lighting and more powerful hardware.

A detailed study with still and flat water surfaces offers some interesting aspects:

1. **Still water remains stationary.**
   Because the surface shape does not change in time, it is possible to take images of the water surface under different recording conditions. This allows to vary several parameters to study their influence on the obtained accuracy, such as the lighting setup, possible image formats and the position of the camera's with respect to the grid.

2. **Accurate alternative methods are available to ground-truth the image-based reconstructed surfaces.**
   Still water remains perfectly horizontal in the entire test tank. This allows us to obtain an accurate description of the water surface by means of traditional measurement techniques. These measured water level differences can then be used to ground-truth the image-based reconstructed surfaces.

3. **Low-parameterized surface models can describe the surface near-perfectly.**
   The approach suggested in this thesis uses a theoretical parameterized model to describe

the surface shape of the water, for which the coefficients are fitted to the observations. Very irregular water surfaces (e.g. a steep and irregular wave traveling across the water surface) are however difficult (or impossible) to describe with an explicit function rule. Given that the surface cannot be approximated with a limited amount of parameters, a reliable reconstruction of such phenomena is in the current implementation of the algorithm (still) not possible. In contrast, a flat surface can theoretically perfectly be described by the surface function derived in Section 4.2.

4. **No loss of feature points.**
   Flat and still water refracts the individual viewing rays in a uniform way. The observed pattern on the water surface shows in that case only a very small deformation or distortion compared to the original pattern on plane F. Additionally, motion blur does not occur because the observed feature points on the water surface do not move while capturing. These two aspects facilitate feature detection and allow us to detect all feature points in every processed image. The uniformly distributed surface points **p** can then be used to run the optimization algorithm in optimal conditions and obtain maximum accuracy.

A second evaluation of the algorithm's performance is done with numerical simulations. Firstly, we determine for a chosen set of image points **q'** the location of feature points **f** by refracting the viewing rays at a known, simulated water surface. Subsequently, we assume a Gaussian distribution $N(0, \sigma)$ for the localization error in the image plane. This random noise is added to the image points, resulting in $\boldsymbol{q'} + \delta \boldsymbol{q}$ that are used as input for our reconstruction algorithm. By comparison of the obtained best-fitting surface with the original simulated surface, the dissimilarity between both can be used as indication for the sensitivity of the algorithm w.r.t. inaccurate feature localization.

Numerical simulations are not only a supplementary validation of the developed methodology, but also offer several advantages w.r.t. experimental tests:

1. They can be used to quantify the effect of some parameters for which the amount of required experimental tests would be too large.
2. In experimental tests, inaccuracies related to several aspects of the reconstruction (see Section 7.1) are always combined. This makes it difficult to estimate the influence of one parameter w.r.t. the robustness and accuracy of our method. In contrast, numerical simulations allow to focus on one particular aspect of the reconstruction.
3. Because in numerical simulations the perfect solution is always known, it is possible to eliminate the finite accuracy of the ground truth. Additionally, it allows to study more complex surface shapes which are difficult to validate in experimental tests due to the lack of alternative measurement techniques.

7.1 Possible errors on the reconstructed water surface

In ideal circumstances, the final result of the optimization procedure is a set of water surface coefficients that define a surface shape which matches with the established or simulated water surface in the tank. In reality, the reconstruction slightly deviates from the actual surface due to several reasons listed hereafter.

7.1.1 Errors related to multivariate optimization

A first error is related to the optimization procedure that is followed to obtain the coefficients that best describe the water surface. As explained in the previous chapter, the Levenberg-Marquardt algorithm (LMA) is used to obtain a multivariate optimization result of the surface shape coefficients. As mentioned in Chapter 6, the results discussed in this chapter are obtained with a step-size based stopping condition with a step-size equal to 0.01 mm.

The solution of LMA is in most cases a good approximation of the minimum E_{tot} but has inherent error, characteristic to non-linear optimization procedures.

7.1.2 Errors related to feature detection

The proposed methodology relies on the detection of features points in the image plane as was explained in Chapter 4. The pixel coordinates of the image points **q'** determine the direction of the incident viewing rays $\vec{u}$ and a reliable reconstruction requires them to be detected as accurately as possible.

7.1.2.1 Inaccuracy related to optical techniques

Accurate feature localization is difficult due to several aspects that are inherent to optical, image-based techniques:

1. **Finite depth of field.**
   Because the distance between the camera and the feature points on the water surface is different for every feature point, an ideal focus of the camera for the entire water surface is difficult to obtain. As such, certain regions that are located closer or further than the focus point will be slightly blurred. The magnitude of depth of field blurring increases in case the camera views the surface at a sharper angle because the relative difference in distance between two opposite sides of the pattern then becomes larger.

   This can be mitigated by increasing the depth of field of the camera which is function of several controllable parameters. Possible solutions are reducing the width of the aperture, increasing the focus distance or increasing the focal length (using a wider lens). The depth range over which the camera captures sharp images can in this way be increased and a larger surface area will be depicted sharply.

2. **Shutter speed of the camera.**
   Secondly, insufficient lighting in the scene causes images with sub-optimal contrast. By reducing the shutter speed, which determines the time the camera's shutter is open, the amount of light on the image sensor is increased which results in brighter images. A small shutter speed can however cause blurry images due to motion blur. Additionally, it affects the amount of noise which will be discussed in the next paragraph. In this chapter, risk of

motion is avoided because stationary surfaces are captured. The shutter speed is therefore taken low enough to obtain bright images in which the feature points are easily discerned.

In practical applications with moving water, the shutter speed should be increased to avoid that captured objects (surface points **p**) move while the camera's shutter is open. If that occurs, the objects become smeared in the image which makes accurate localization of such points impossible. To avoid this motion blur, a higher shutter speed is therefore taken in the test case described in Chapter 8.

3. **Image noise.**
   A final adverse effect is the presence of image noise. Image noise is defined as the random variation of the brightness (color intensity) in images and usually appears as little dots or speckles in regions that should be smooth. Noise can be reduced by both fine-tuning the camera parameters as well as by post-processing of the obtained images.

   A first influencing parameter is the ISO (International Organization for Standardization) setting of the camera, which determines the camera's sensitivity to light. A higher ISO setting results in more light captured by the camera but also increases noise in the signal. Secondly, the camera sensor heats up during recording which results in more risk of image noise. That is why the camera should not work for extended periods and breaks between different recordings is advisable. Finally, a longer exposure time (i.e. a smaller shutter speed) has a negative influence because it increases the risk of electrical noise during image recording.

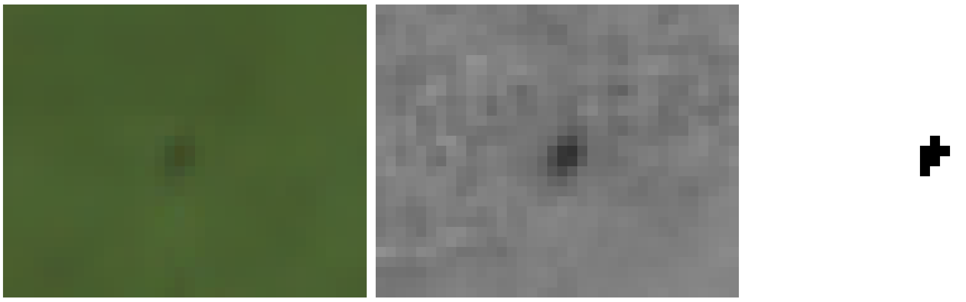
   Different methods have been developed to reduce noise in a post-processing stage. Krig (2014) gives a large overview of possible techniques that can be used to increase the image quality. In all methods, the goal is to separate the pixel intensity originating from the real surface detail and the noise contribution. Most of these methods apply a smoothening filter that computes the average pixel intensity in the neighborhood of the considered pixel. The value for every pixel is then replaced by the average of the intensity within 'the filter mask'. This however reduces contrast in the image and therefore negatively affects feature detection. An ideal balance between maintaining sharp transitions but on the other hand removing noise needs therefore to be searched. Because the amount of noise depends on the chosen camera settings (which can be different for different recordings), the degree of noise removal in this thesis was fitted to each specific image sequence.

7.1.2.2 Inaccuracy related to the feature detector

Due to the limited time available during this thesis, a suboptimal combination of feature pattern and feature detection method was implemented which suffers from a limited accuracy. As explained in Chapter 5, the feature detection algorithm used in this thesis (OpenCV's function *SimpleBlobDetector*) detects little blobs (dots) in the image plane and labels their centers of gravity as image points **q'**. In case the entire feature point is depicted in the image, the center of gravity of such a blob corresponds with the actual center of the corresponding feature point.

In practice, some post-processing of the captured images is necessary to filter out noise in the image and remove accidental small particles present in the water. By applying thresholds and local contrast enhancement methods, the most distinguishable spots in the image can finally be retrieved which is shown in Figure 6.1. This facilitates feature detection but makes the location of the detected image points **q'** also influenced by the original image quality and the processing adaptations.

Both the lighting conditions as well as the focus of the camera are difficult to optimize for the entire spatial extent of the feature pattern. This causes that some feature points are not clearly distinguishable from their surroundings. By applying intensity-based thresholds, part of the dark feature points in the image can in that case accidentally be set to 'background'. In the final post-processed image, this part of the feature point is in that case no longer visible. A comparison of an inaccurate detected image point before and after thresholding is depicted in Figure 7.1. The blob after post-processing then only represents a portion of the actual surface area of point **p** on the water. The center of gravity of the blob will therefore be slightly different (order of 1-3 pixels in our setup) than the actual center of point **p**.



(a) Original point in color image (visible light)

(b) After applying local contrast enhancement methods

(c) After applying intensity-based thresholds

Figure 7.1: Illustration of thresholding process in ImageJ: the exact location of the center of the image points is somewhat ambiguous. The center of the finally detected blob might be deviating from the actual center of the scene point

7.1.3 Errors related to camera calibration and initialization

The calibration procedure explained in Chapter 6 is used to determine the distortion coefficients and intrinsic camera parameters. Next, the extrinsic camera parameters (expressing the camera's position w.r.t. the grid) are obtained by solving the PnP-problem based on an image of the feature pattern without water. Both the camera calibration and initialization step require the detection of image points that correspond with particular points in the scene (corners of a chessboard pattern or features **f** on plane F). For this reason, errors related to inaccurate detection of image points (discussed in the previous section) have a indirect influence on the estimated camera parameters.

Additionally, the intrinsic parameters (stored in matrix K) and distortion coefficients of the camera are a solution obtained with the LMA, in which the total average reprojection error over all images is minimized. As explained in Chapter 6, also the estimated matrix R and vector T are the result of a Levenberg-Marquardt optimization based on the reprojection error of the initialization image. For both, the solution therefore inhibits an inherent error that is characteristic to optimization algorithms.

7.2 Description of the validation procedure

7.2.1 Experimental validation

In our experimental tests, image sequences are taken with multiple cameras in which the (still) water level is adjusted in discrete steps. This change in water level is measured simultaneously with a level gauge, located at the side of the water tank. Comparison between the results obtained with these two possible approaches allows to verify if the reconstructed water surfaces are correct.

7.2.1.1 Reconstruction of the water surface based on refractive disparity

We start the reconstruction by the initialization step as explained in Chapter 6. An image of the feature pattern without water present in the tank is used to estimate the camera's extrinsic camera parameters, given the intrinsic parameters based on an earlier camera calibration. For every established horizontal water surface, the image points **q'** are located in the corresponding frame of every camera. Using this set of pixel coordinates, the global optimization algorithm is applied to compute the surface coefficients that best explain the refractive disparities of the feature points.

The feature grid itself is printed on a metal plate that is placed on the bottom of the tank. This plate has a certain thickness that can vary slightly between the extreme points of the grid (fractions of mm's). Additionally, the bottom of our test-tank is not perfectly horizontal. These two considerations cause that a small inclination of plane F on the bottom of the tank is possible in both the x- and y-direction. Still water on the other hand remains always perfectly horizontal, as its surface shape is governed by the earth's gravity force. Although the reconstructed water surfaces are horizontal, the local water depth can therefore vary linearly across plane F.

Given the inclined feature plane F, every surface can in theory be described using only three coefficients: A_{00}, B and C. These final two are necessary in order to describe the linearly varying water depth. This resulting surface model then becomes:

$$\eta(x,y) = A_{00} + B\frac{x}{L_x} + C\frac{y}{L_y} \tag{7.1}$$

Additionally, the optimization can be conducted using all 8 surface coefficients in eq. (4.19). It can be predicted that these higher order terms will be used to fit the local surface shape to inaccuracies related to the camera pose estimation and feature localization.

A second important aspect of our surface reconstruction is the error metric that is used as error measure that needs to be minimized in the optimization of the surface coefficients. In Chapter 4, we suggested two possible metrics: the 'normal collinearity metric' and the 'disparity difference metric'. These will further be referred to as metric 1 (M1) and metric 2 (M2) respectively. In the following, the combination of a chosen error metric (M) and surface model (i.e. amount of optimized parameters in the surface function P) will be denoted as the chosen 'reconstruction configuration' (M,P).

The optimization algorithm uses the physical coordinates of surface points **p** to find the best fitting surface through these points. As such, the coefficients are only optimized to describe the shape of the water in the area where surface points are located. This limits the spatial extent in which the optimization result is valid. Depending on the reconstructed water depths, the location of these surface points changes and the coefficients are therefore optimized to describe a slightly varying surface area. Additionally, the different cameras are positioned at opposite sides of the feature grid. Their views overlap in that case only in the middle part of the grid as shown in Figure 7.2. In Section 7.3.5, the advantage of combining multiple cameras will however be investigated.

Because of both aspects, we limit the spatial extent over which ‘the error on the actual surface’ will be evaluated. A spatial domain W(x,y) is chosen above the middle part of the feature pattern on the bottom of the test tank. For every camera and surface height, the optimized coefficients are valid within W which allows to derive an unbiased quality measure.

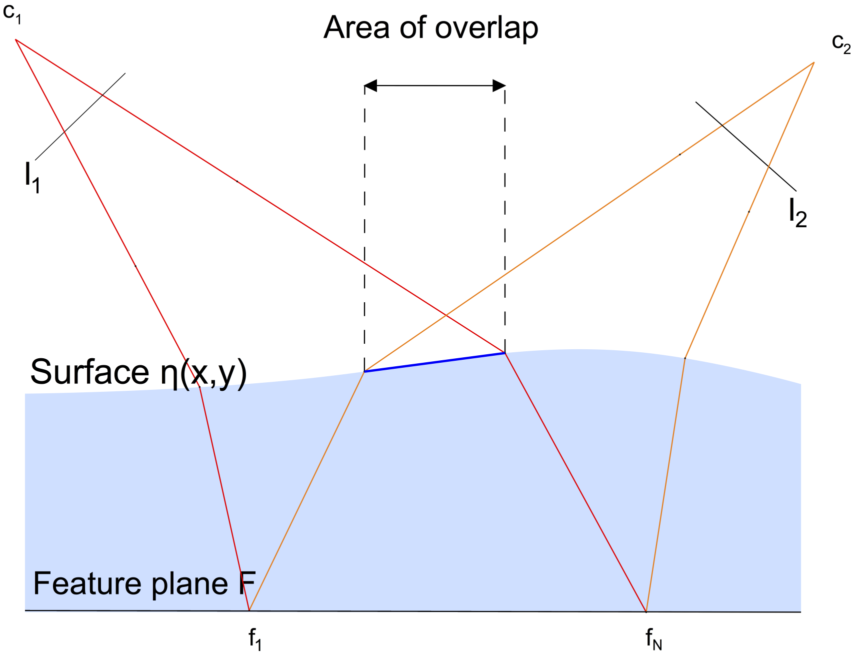
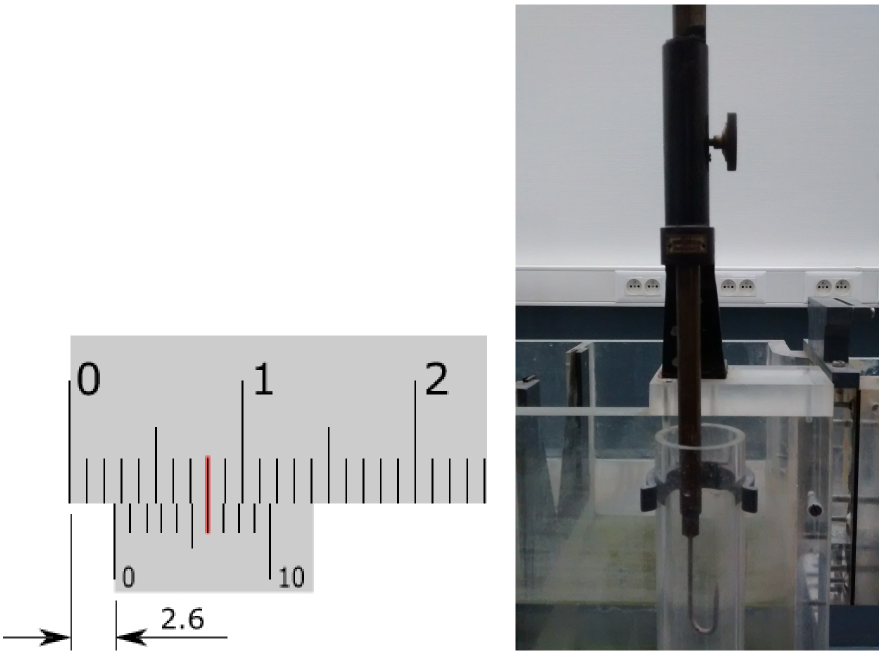


Figure 7.2: Area in which surface points detected by both cameras and for all water height levels overlap determines spatial domain W used during the validation. Based on Morris (2004).

7.2.1.2 Obtaining a ground truth water level with a level gauge

Most traditional measurement techniques, such as the level gauge used in this thesis, only allow to measure water level changes in a fixed point. For this reason, a linearly varying surface height is difficult to validate using a direct comparison of the reconstructed surface shape and such point measurements. The change in water level between two horizontal surfaces is however the same at every point in the tank. A comparison between the computed water level changes within W(x,y) and the level gauge measurements is therefore possible.

The accuracy that can be obtained with a level gauge is however limited. Using a vernier scale, the available level gauge allows to obtain an accuracy of 0.1 mm on the individual water levels. The principle of a vernier scale is depicted in Figure 7.3, together with the level gauge used during the experiments. Because we compare water level differences, each verification combines two measurements with the level gauge. This doubles the error made with the level gauge, resulting in an accuracy of 0.2 mm for every water level change.



(a) Vernier scale

(b) Level gauge positioned at the side of the water tank

Figure 7.3: Principle of a vernier scale.
Firstly, the fixed scale is first used to determine the water level with 1 mm accuracy. Because this measure is usually located between two mm-markings, the finer vernier scale is required to obtain 0.1 mm accuracy. The position where the markings of the finer vernier scale and the fixed scale coincide is used as sub-millimeter indication. In case this e.g. occurs at the sixth marking of the vernier scale, 0.6 mm should be added to the fixed scale marking.

7.2.1.3 Averaging procedure to mitigate inaccurate feature detection & definition of reconstruction error measure

As was already mentioned earlier, incorrect feature localization is one the most determining factors in the accuracy of the algorithm. This issue is not related to the reconstruction methodology and could partly be solved by improving the feature detector in further development of our method. In order to validate the core of our reconstruction algorithm, a sequence of ten images is taken for each camera at the same established water depth. Averaging the results obtained with these different images allows to partly mitigate incorrect pixel coordinates **q'**. For each reconstruction configuration, this results in a single averaged surface at every reconstructed water depth.

Next, two averaged surfaces found with the same choice of (M,P) are compared and the difference in surface height for the N points within W is computed, resulting in N water level changes Δh. These are compared with 'the actual water level change', obtained from the point measurements using the level gauge Δh_{gauge} to obtain an error measure $|E_q|$ for every corresponding image point:

$$|E_q| = |\Delta h - \Delta h_{gauge}| \quad \text{with q=1, ..., N} \tag{7.2}$$

These N errors are finally spatially averaged within W to obtain a global error measure $|E_{mean}|$:

$$|E_{mean}| = \frac{\sum_N |E_q|}{N} \tag{7.3}$$

7.2.2 Numerical validation

A second validation is based on Monte-Carlo simulations in order to evaluate the sensitivity of the developed method to incorrect feature localization. Using these simulations, we attempt to quantify the influence of different parameters within the methodology on the performance of our algorithm. As explained in the beginning of this chapter, simulations allow to study a large number of combinations of parameters, which is not feasible experimentally.

We start each simulation by assuming a camera position and camera parameters (stored in matrices K and R and vector T) that are representative for the experimental tests that will be conducted. Because in numerical simulations no images are available, we generate the image points **q'** as a regular pattern in the image plane. The chosen regular pattern consists of 50 features points in the longitudinal direction of the tank (y-direction in Chapter 6) and 15 feature points along the width of the tank (x-direction).

The 'actual water surface' is simulated based on the theoretical description $\eta(x,y)$ with a predefined set of surface coefficients. The refraction of the viewing rays through the generated points **q'** is then determined based on the mentioned simulation input parameters. These refracted rays, starting at the surface points towards plane F, are finally used to obtain a list of feature points **f** by computing their intersection with the plane at a height z=0.

The error made in feature detection is incorporated by assuming that feature localization errors follow a Gaussian distribution $N(0,\sigma)$ in both the u- and v-direction. By adding this random noise to the coordinates of the original chosen image points, a new set of 'shifted points' $\boldsymbol{q'}+\delta\boldsymbol{q}$ is found in every iteration step. Given these incorrect image points $\boldsymbol{q'}+\delta\boldsymbol{q}$ and feature points **f**, the global reconstruction algorithm is then applied with a chosen configuration (M,P). This results in a reconstructed surface $\eta'(x,y)$ (different for every iteration step) that will be deviating from what originally was simulated as is shown in Figure 7.4.

The error made w.r.t. the original water surface is evaluated over a certain area of the reconstructed surface. Similarly as was discussed in the previous section, the spatial domain W is chosen in which the optimization results for the various simulated water surface remain valid. Taking the mean over the entire domain W and over all iterations, this finally gives a mean error $|E_{mean}|$:

$$|E_{mean}| = \frac{\sum_{iterations}\sum_{N}|E_{q,it}|}{\#\,iterations \cdot N} \tag{7.4}$$

Because this numerical study is based on Monte-Carlo simulations, a reasonable choice for the amount of iterations has to be made. To that end, we evaluate the convergence rate of the average error for one specific test case. The original surface is simulated at a constant water depth of 40 mm and 500 simulations are conducted assuming the localization inaccuracy $\delta\boldsymbol{q}$ follows a $N(0,\sigma)$-distribution with $\sigma = 2$. Figure 7.5 shows the average of $|E_{mean}|$ over all iterations in function of the amount of iterations performed. Because the mean error does not longer change after 70-80 iterations, the numerical simulations discussed in this chapter are conducted with 100 iterations.

Because the numerical simulations do not suffer from an inclination of feature plane F, a horizontal water surface can theoretical be described using only one constant term A_{00}. For this reason, we not only adopt a 3- and 8-parameter surface model but also reconstruct the surface as $\eta(x,y) = A_{00}$ in the simulations with horizontal water surfaces (Section 7.3.3.3).

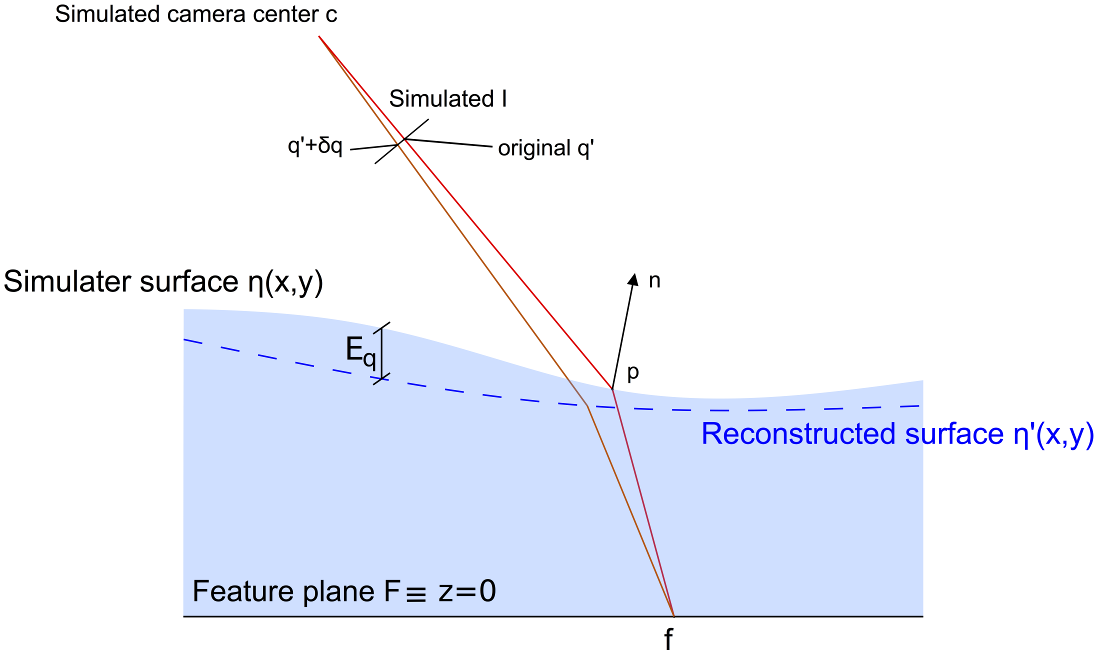


Figure 7.4: Due to the random noise $\delta \boldsymbol{q}$ added to the original image points $\boldsymbol{q'}$, the reconstructed surface $\eta'(x,y)$ will not be the same as the original simulated surface $\eta(x,y)$. The difference in water level between these two surfaces defines the reconstruction error E_q.

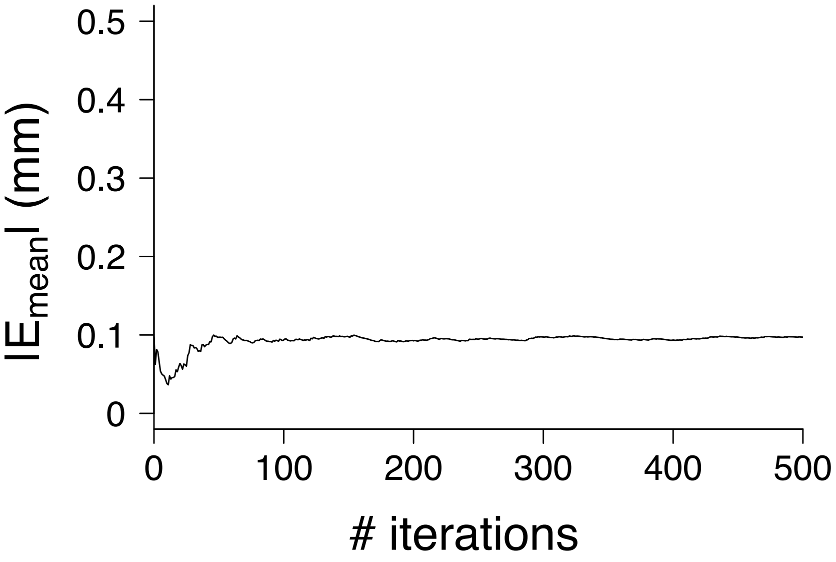


Figure 7.5: Convergence rate of $|E_{mean}|$ over all iterations, in function of amount of iterations conducted. The mean error $|E_{mean}|$ stabilizes after 70-80 iterations.

7.3 Evaluation of the influencing parameters

The reconstruction of the water surface uses a certain choice of three characterizing parameters. Firstly, the optimization of the water surface shape coefficients can be done based on the detected image points of one or a combination of multiple cameras located at different positions w.r.t. the feature grid. Secondly, two different error metrics can be used: the normal collinearity metric (M1) and the disparity difference metric (M2) as explained in Chapter 4. A third parameter is the amount of theoretical coefficients that is used to describe the water surface. The influence of these parameters on the reconstruction accuracy and robustness is studied in the following sections. Correct and accurate feature detection plays however a primordial role in the reconstruction procedure. That is why in Section 7.3.1, we first investigate how the best image quality can be obtained in order to facilitate feature detection.

7.3.1 Influence of the image quality: lighting conditions & image format

7.3.1.1 Theoretical considerations

An important requirement for accurate surface reconstruction is to obtain images in which the feature points are easily detectable. Because the color of the printed feature pattern is black, the background on plane F should be as bright as possible. As was discussed in Section 6.2.2 of Chapter 6, two different lighting setups are tested to obtain maximum contrast between the feature points and their surroundings.

Based on traditional optical measurement setups, we first use visible light to illuminate the feature plane. Although the lighting setup is fine-tuned to obtain the best possible images, it proves difficult to obtain a uniform lighting over the entire feature pattern. Combined with depth of field-blurring, this makes the exact position of some points in the image plane somewhat ambiguous and largely dependent on how the images are post-processed.

For this reason, a second method with UV light is adopted. The fluorescent paper serves in that case as apparent light source when it is illuminated with UV light. Because the emissivity of the paper is the same over the entire feature pattern, a uniform brightness is in this way ensured. By blocking the incident light with ink of the printed feature pattern, the contrast between the feature points and the rest of the paper is additionally maximized. Based on the reasoning above, it can be assumed that a more accurate feature localization can be obtained with UV light. A more reliable and detailed reconstruction of the water surface is therefore expected.

For dynamic water surfaces, the frame rate of the camera comes important in order to avoid motion blur and maximize the temporal resolution of the reconstruction procedure. However, the data cable used to transmit the captured images has a maximum throughput as was explained in Chapter 6. This limits the maximum frame rate and the resulting temporal resolution that can be obtained. Because grayscale images have a smaller size (in bytes) compared to color images, they allow a higher frame rate without exceeding the maximum bandwidth. Grayscale image are one the other hand a weighted sum of the three color channels: red, green and blue. As was explained in Chapter 6, the refractive index of water depends on the wavelength of the light rays and should be adjusted according to the processed color channel. Grascale images can on the contrary only be combined with a single refractive index, representative for all three channels. The range over which the refractive index r_w changes in the visible light spectrum is however small. The nonlinearity of Snell's law makes the influence of this simplification nevertheless difficult to predict.

7.3.1.2 Experimental validation

We conducted experimental tests by using two cameras at opposite sides of the feature pattern (in the longitudinal y-direction). Color images with both UV and visible light were split in their three color channels and the green channel was retained as input for the reconstruction. In case of visible light, we additionally took grayscale images to investigate the influence of the varying refractive index of the water. Because green light is situated in the middle of the visible light spectrum, the same refractive index of 1.336 (green light) was used as a representative value for the weighted sum of the three color channels.

Figure 7.6 shows the spatially averaged error $|E_{mean}|$ for the various types of images. The error metric and the amount of coefficients, previously denoted as reconstruction configuration (M,P), are indicated in the top left corner of these graphs. On the x-axis, the initial and final water height are indicated as $h_i - h_f$.

For most cases, the mean error made on the reconstructed water level differences is situated within the accuracy-range of the level gauge (0.2 mm), indicated by the hatched area. Although the images with UV light seem to give slightly better results, no significant difference between the different lighting conditions can be discerned. It can be concluded that for still water, both lighting sets allow to capture high-quality images.

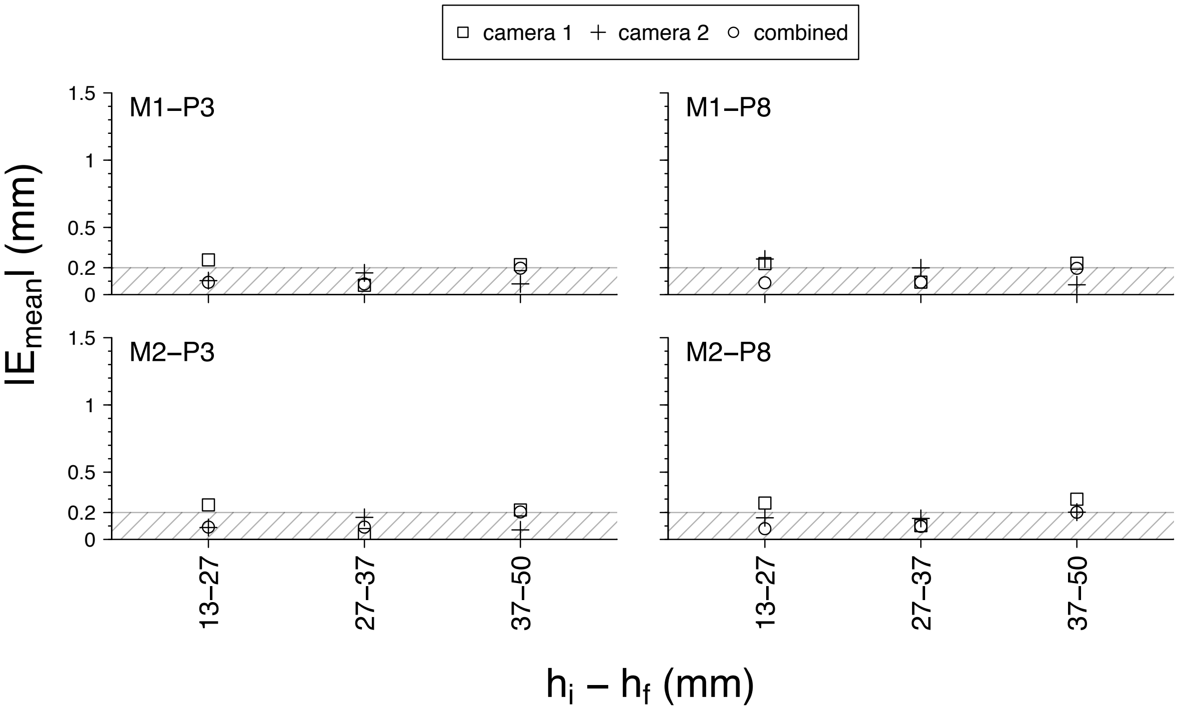
Secondly, the influence of the variability in the refractive index within the visible light spectrum seems to be negligible. This is most likely caused by the narrow range in which the index of visible light is situated. Although this depends on which author is consulted (Harvey et al. (1998); Schiebener et al. (1990); Thormählen et al. (1985)), this range is approximately 1.331-1.344 for water at 20°C. The chosen refractive index of green light is situated in the middle of this range and therefore considered as a good representative value for the entire visible light spectrum. Additionally, green light (with r_w=1.336) contributes for a larger part in the final monochrome image compared to the other two channels. The weighing factors depend on the transformation method that is used, but as a finger rule $2/3^{th}$ green in the final grayscale image can be assumed.

It must be mentioned that in case larger water depths are reconstructed, we expect that a correct value of r_w will become more critical for an accurate surface reconstruction. The refracted part of each ray then travels a larger distance through the water, making a correct estimation of the change in direction more important.

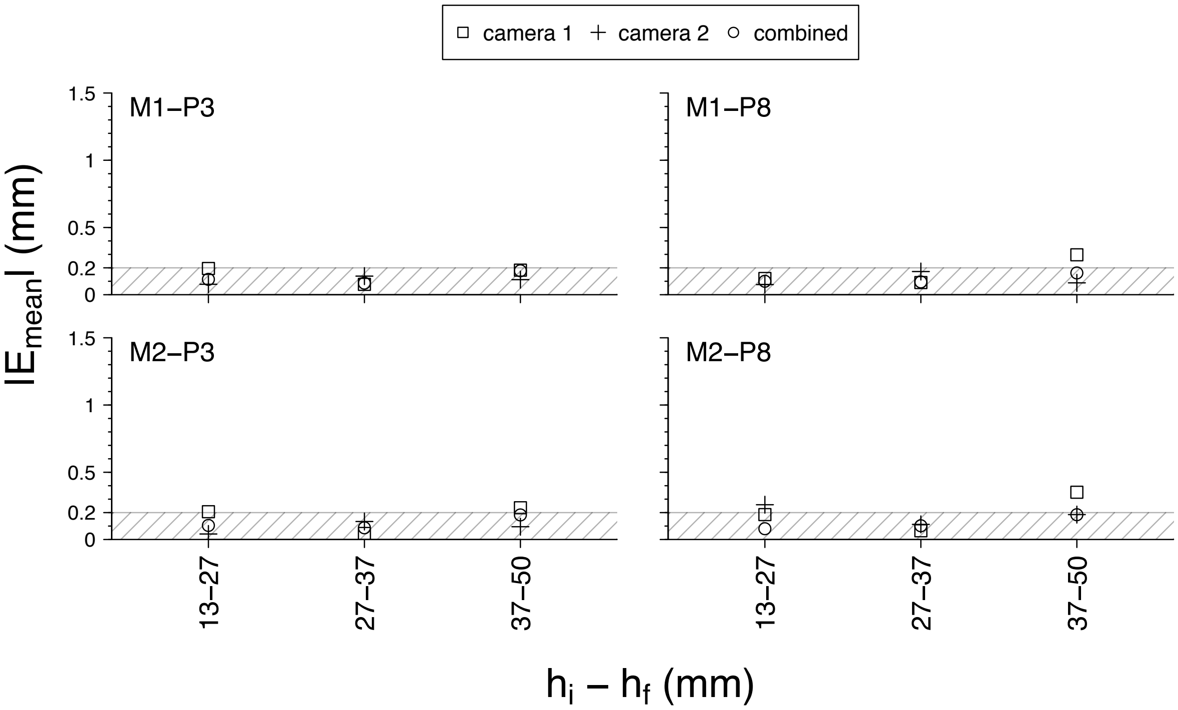
This comparison shows that the reconstruction algorithm is able to reconstruct water surfaces with high accuracy. In case sufficient care is taken to obtain strong and uniform illumination of the feature plane, the difference between UV and visible light seems small. Additionally, the simplification by using monochrome images does not significantly decrease the reconstruction accuracy. Based on the reasons mentioned earlier, we assume nonetheless that UV offers the highest possible contrast between the feature points and their surroundings. For this reason, a second set of measurements with UV light will be adopted to study the influence of several parameters in the reconstruction algorithm.

7.3.1.3 Numerical validation

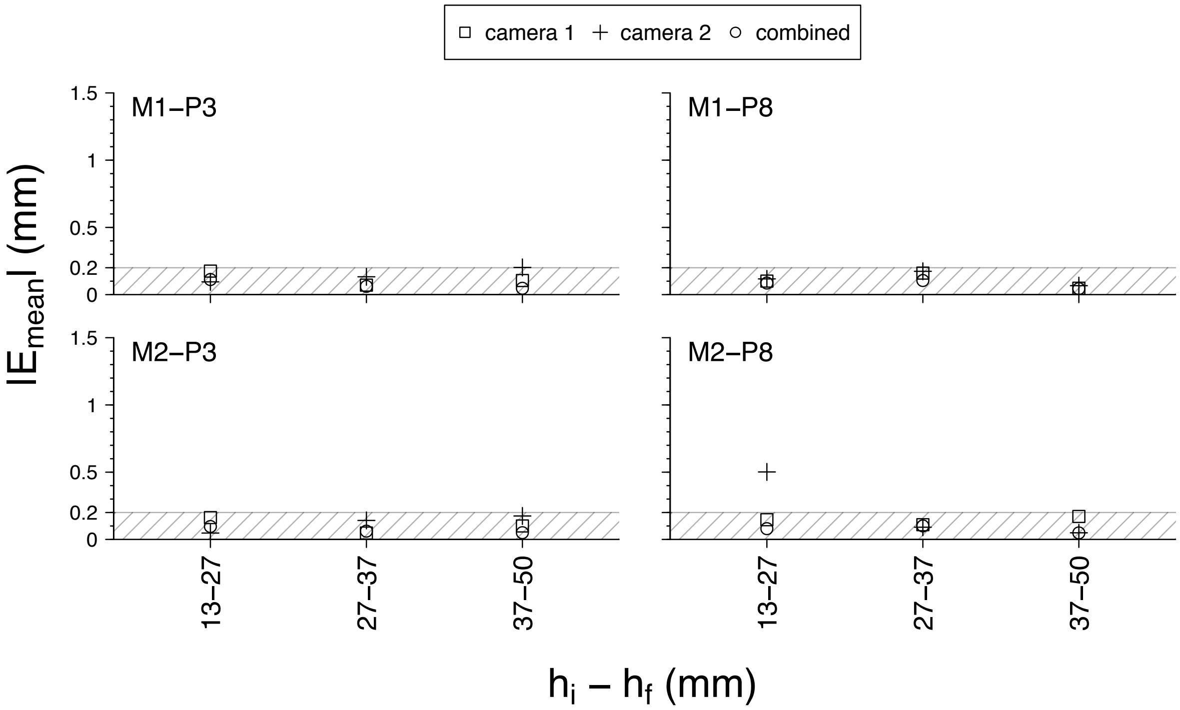
Our numerical simulations use a set of generated image points instead of detecting them in images. As a result, the numerical simulations did not allow to study the influence of the image quality on the performance of our algorithm.



(a) Visible light illumination and grayscale images.



(b) Visible light illumination and color images (green channel).



(c) Ultraviolet light illumination (green channel).

Figure 7.6: Comparison of $|E_{mean}|$ between visible light (monochrome or color) and UV light.

7.3.2 Influence of the camera calibration and initialization

7.3.2.1 Experimental validation

Camera calibration

Different distortion models are possible to obtain the undistorted pixel coordinates of the image points of the feature pattern, each incorporating a different set of distortion coefficients. As explained in Section 6.3.2, the cameras are calibrated with the OpenCV function *calibrateCamera*, which contains several possible distortion models. An overview of these distortion models and the incorporated distortion coefficients is given in Table 7.1. For a definition of the corresponding transformation functions, we refer to Chapter 6.

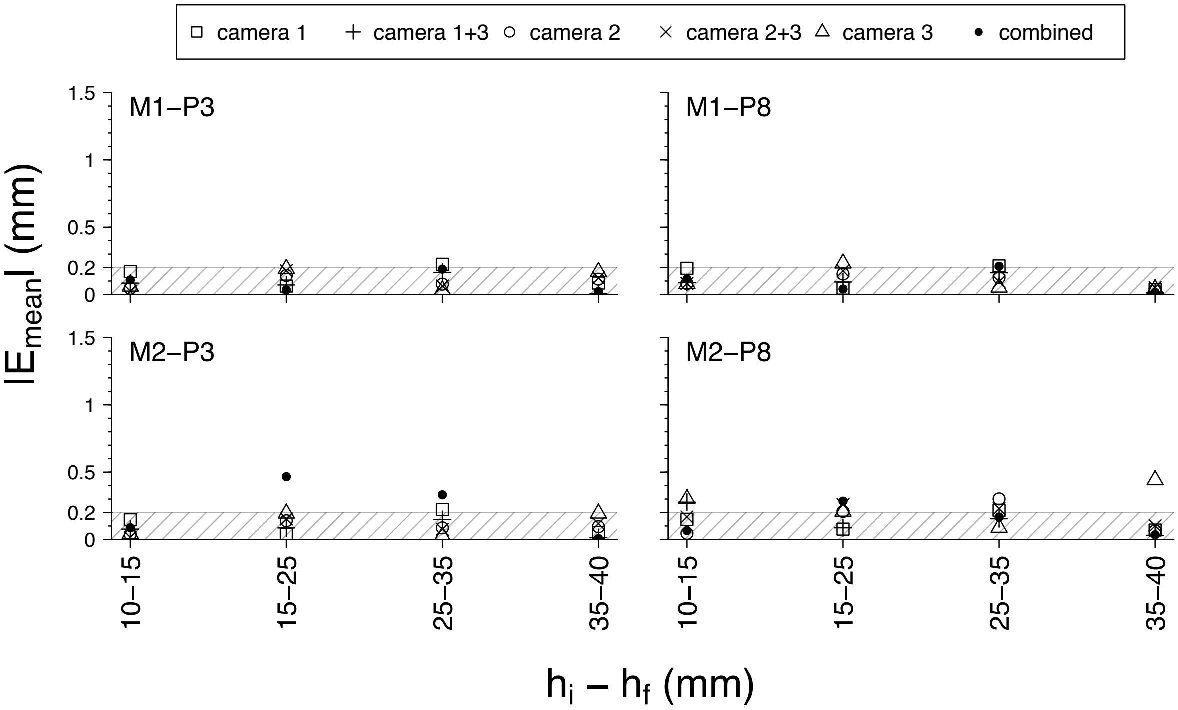
To verify the influence of the adopted distortion model, we reconstructed the surface using the undistorted image points obtained with the different distortion transformation functions. Figure 7.7 shows that the difference between the possible distortion models is very limited and almost all errors lie within the accuracy range of the level gauge. Nonetheless, we expect the most accurate results with the rotational model. This model takes eight distortion coefficients into

Table 7.1: Overview of the possible distortion models with OpenCV.

Distortion model	**Distortion coefficients**
Basic Brown-Conrady model	$\boldsymbol{k_1, k_2, k_3, p_1, p_2}$
Rotational model	$\boldsymbol{k_1, k_2, k_3, k_4, k_5, k_6, p_1, p_2}$
Tilted model	$\boldsymbol{k_1, k_2, k_3, p_1, p_2, \tau_x, \tau_y}$

consideration, providing a better description of the distortion in the image plane. Because no adaptations to the camera sensor were made, the tilt of the image sensor can only be caused by manufacturing defects. The tilted distortion coefficients are therefore approximately zero, making the third model less appropriate. Based on these considerations, we will therefore use the second distortion model during this thesis. Because Figure 7.7 shows that the difference between the three models is small, another choice could also be possible.

As explained in Chapter 6, the quality of camera calibration is usually expressed as the total average reprojection error. Because this could give an indication which calibration gives better reconstruction results, we compared the reprojection errors corresponding with the different camera calibrations and the error $|E_{mean}|$ of the reconstructed surfaces. No clear correlation between the average reprojection error and the reconstruction result seemed however present. The average reprojection error can as such not be used to predict which camera has the highest potential w.r.t. the accuracy of surface reconstruction.

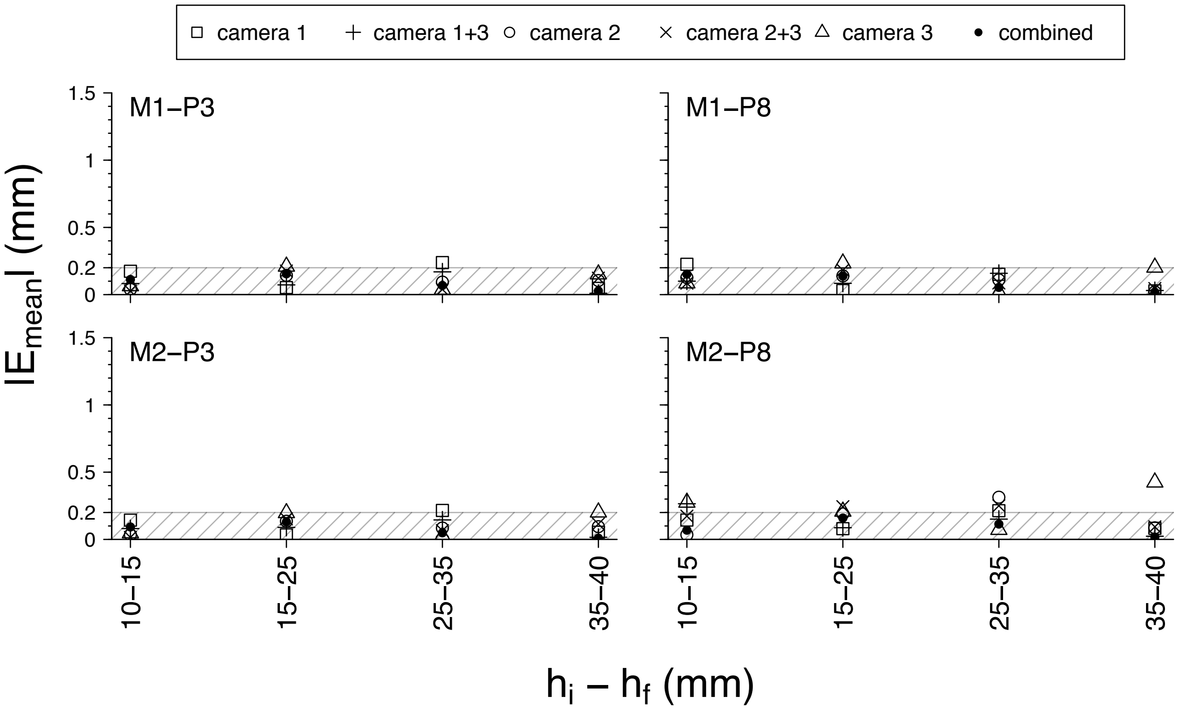


(a) $|E_{mean}|$ with basic Brown-Conrady model.

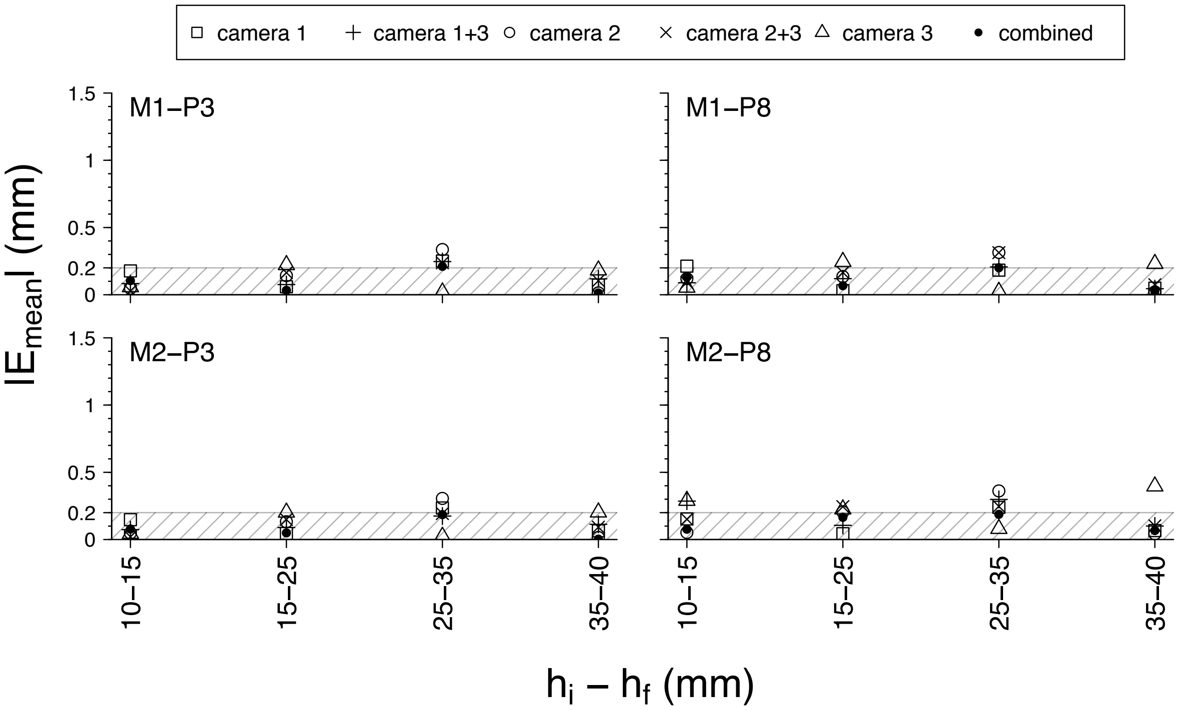
Initialization

In the initialization step, the extrinsic camera parameters (stored in matrix R and vector T) are estimated. To evaluate the influence of this pose estimation, i.e. position and orientation w.r.t. the chosen reference system, four initialization images without water were taken. The resulting extrinsic camera parameters varied due to the error made in every solution of the PnP-problem. Because it is not important to determine which initialization image gives the most accurate results, a direct comparison of the average error $|E_{mean}|$ using one of the four initialization images is not given. Instead, we determine in which conditions the final result is more sensitive to the initialization step.

We computed four water surfaces (each an averaged surface over images 1-10) based on the four different initializations. For every point in W, the difference between the resulting highest and lowest surface height is used as indication for the sensitivity of the algorithm w.r.t. the camera pose estimation. The spatially averaged values of these deviations, denoted as Δh_{mean},



(b) $|E_{mean}|$ with rotational model.



(c) $|E_{mean}|$ with tilted model.

Figure 7.7: Comparison of $|E_{mean}|$ for different distortion models.

are plotted in Figure 7.8 for different reconstruction configurations at two specific water heights. This shows that the difference between the four initializations remains small and is not related to the average water depth.

On average, the deviations can become higher in case 8 distortion coefficients are optimized.

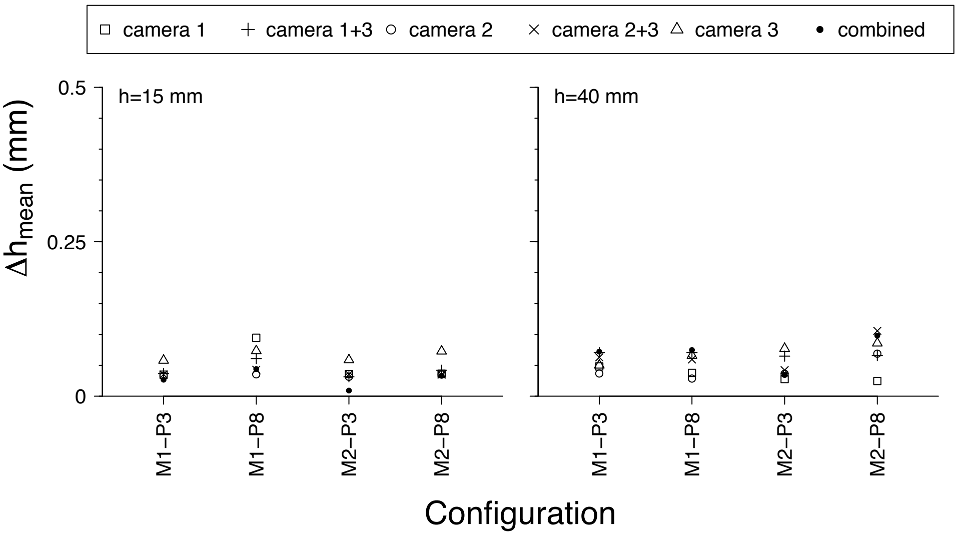


Figure 7.8: Spatially averaged deviations Δh_{mean} on reconstructed water surface based on four different initialization images. Note that this is not a comparison with the measurements obtained with a level gauge.

This seems reasonable because in that case a more complex surface is reconstructed to account for the refractive disparities. A small change in the estimation of the extrinsic camera parameters can in that case lead to a larger variation of the reconstructed water surface. The error metric that is used appears not to affect the impact of pose estimation on the obtained result.

Finally, we conclude from Figure 7.8 that a combination of two or three cameras makes the sensitivity of the algorithm w.r.t. the initialization of the cameras lower. In most cases, the scatter on the obtained surfaces is (especially for three cameras) significantly lower than in case a single camera is used. Deviations related to one camera are then attenuated by the other cameras, making the reconstructed surfaces more consistent.

7.3.2.2 Numerical validation

Undistorted image points are retrieved by a transformation of the pixel coordinates, i.e. shifting them in the image plane according to the distortion model that is chosen. The methodology used in our numerical validation is based on the same principle in which we shift the original chosen image points over a distance $\delta \boldsymbol{q}$. The influence of a incorrect estimation of the distortion coefficients is as such incorporated in every test discussed in this chapter.

Due to the limited time available during this thesis, a numerical evaluation of the initialization step was not not more possible. Hence, we suggest a methodology that could be used in further studies on the algorithm's performance.

The estimation of the camera position is usually based on matching detected image points with known locations of the corresponding points in the scene. In our numerical simulations, we reversed this methodology by computing points **f** for a given calibrated camera. Inaccuracies related to pose estimation could therefore be modeled by shifting points **f** similarly as was done for points **q'**. Even in case the image points **q'** remain the same, the reconstructed surface will in that case change due to the incorrect camera position w.r.t. this deformed feature pattern $\boldsymbol{f} + \delta \boldsymbol{f}$.

7.3.3 Influence of the water level and camera position

7.3.3.1 Theoretical considerations

The refractive-based geometry of our methodology is defined by the location of points **p**, **f** and camera center **c**, which determine the direction of the incident and refracted rays $\vec{u}$ and $\vec{v}$. As a result, these two correlated parameters (water level and camera position) influence the magnitude of the refractive disparities and the accuracy that can be reached.

Firstly, the position of the camera determines the overall quality of the captured images. The risk of depth of field-blurring (explained in Section 7.1) increases when the relative difference in distance between the camera and two opposite sides of the pattern increases. As a result, part of the projected feature pattern on the surface becomes in that case more difficult to locate accurately.

Secondly, the camera position determines the angle of the viewing rays between the camera and the observed feature points on the water surface. Snell's law expresses the change in direction of the refracted viewing rays at the air-water interface. Because of the nonlinearity of Snell's law, the position of the camera has a nonlinear effect on the observed 'image disparity shifts' in the image plane. For cameras that observe the water surface at a grazing angle, the 'refractive disparity' becomes larger compared to cameras positioned more above the reconstructed area. As a result, the corresponding image disparity shift from **q** to **q'** increases as is shown in Figure 7.9. It can be expected that in case the camera is positioned more directly above the feature grid, a high-precision surface reconstruction is more difficult to obtain. A small change in water level might in that case be insufficient to cause large movements of the detected image points. The image disparity shifts then become too small relatively to inaccuracies due to feature detection.

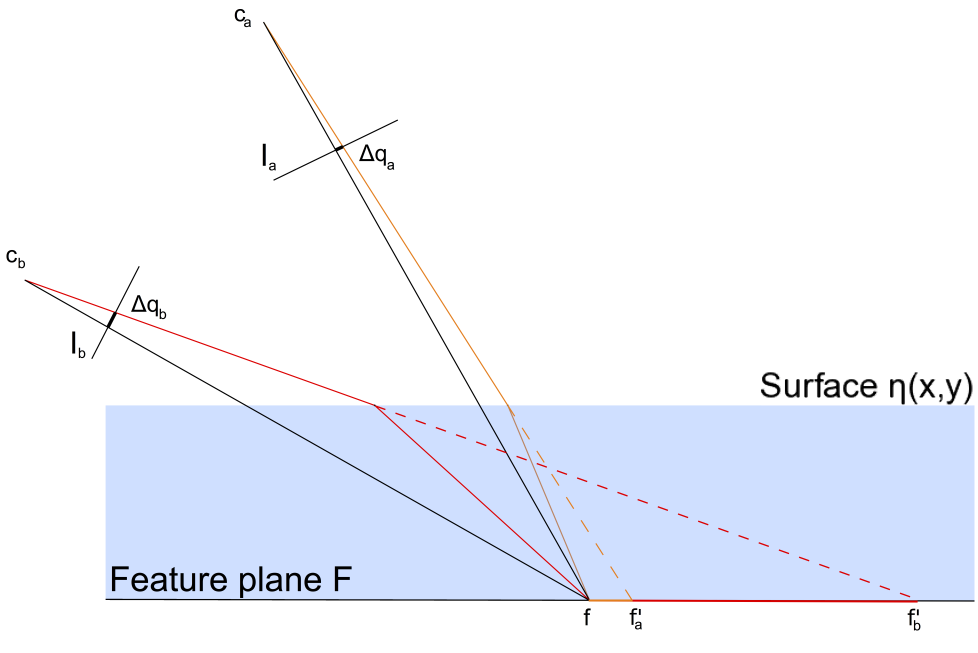


Figure 7.9: The refractive disparity on feature plane F is larger for camera $\boldsymbol{c_b}$, which views the water surface at a grazing angle, compared to camera $\boldsymbol{c_a}$, which observes the surface more perpendicular. This causes a larger image disparity shift $\Delta \boldsymbol{q_b}$ from **q** to **q'** compared to $\Delta \boldsymbol{q_a}$.

Similar considerations can be made w.r.t. the (average) water depth. A larger water depth results in a larger change in pixel coordinates because the movement of points located closer to the camera corresponds with a larger movement in the image plane. Considering that the cameras observe the water surface from above, larger surface heights move the points **p** towards the camera. Additionally, the refracted rays travel a larger distance in the water between points **p** and **f**. For a constant change in water level, this results in larger changes of the refractive disparities and larger movements of the surface points as can be seen in Figure 7.10. We therefore assume that the accuracy of the reconstructed surfaces increases with increasing (average) water depth.

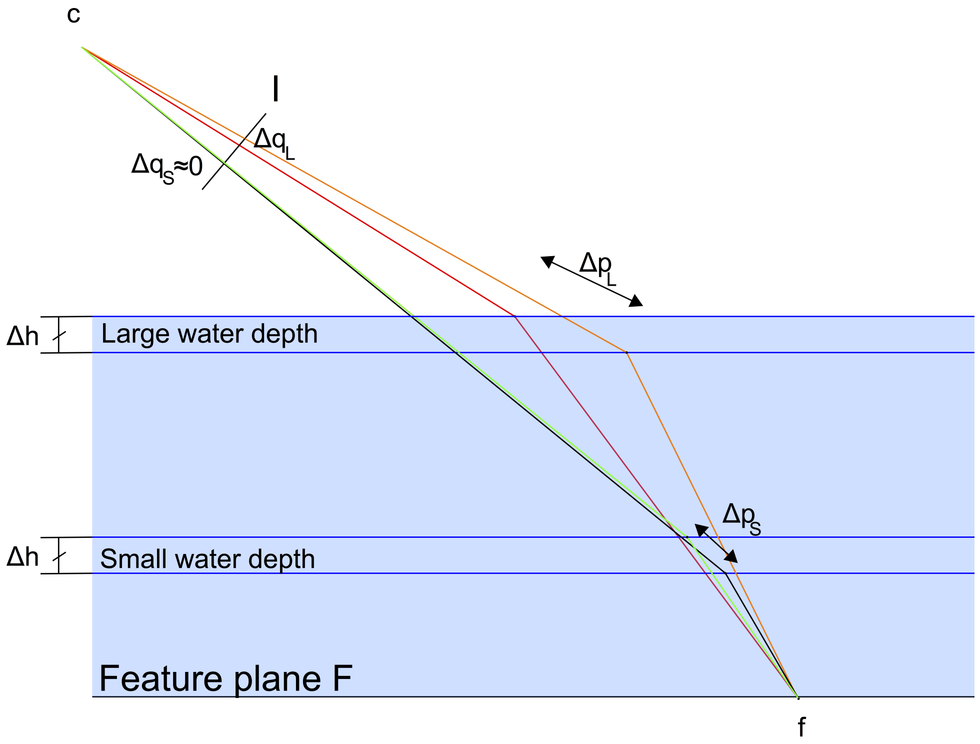


Figure 7.10: The same water level change causes larger changes in the refractive disparities (not indicated) and larger movements of the surface points $\Delta \boldsymbol{p_L}$ at a high water level. The resulting image disparity shifts $\Delta \boldsymbol{q_L}$ become more pronounced compared to those at small water depths ($\Delta \boldsymbol{p_S}$,$\Delta \boldsymbol{q_S}$).

7.3.3.2 Experimental validation

We reconstructed the same water surface with two cameras that view the feature pattern at a different angle and distance from the water surface. Both cameras were positioned at the same horizontal position w.r.t. the feature grid but with a varying height above the bottom of the tank. Camera 1, positioned at a height of 27 cm, viewed the water surface under a sharper angle than camera 2 at a height of 38 cm. Figure 7.11 shows that the average error for both cameras usually falls within the measurement accuracy of the level gauge. We assume that the disadvantage of smaller movements in the image plane of camera 2 is mitigated by a more accurate feature localization. Based on the average error, it proves difficult to determine ‘the best camera pose’ to obtain an accurate reconstruction result.

Hence, we applied a different methodology by analyzing the variation of $|E_q|$ over the 10 processed images in the entire domain W. Figure 7.12 shows these box plots for different choices of (M,P), which indicates that the mean and variation are smaller for camera 1 which views the surface at a more grazing angle. Although more pronounced for the disparity difference

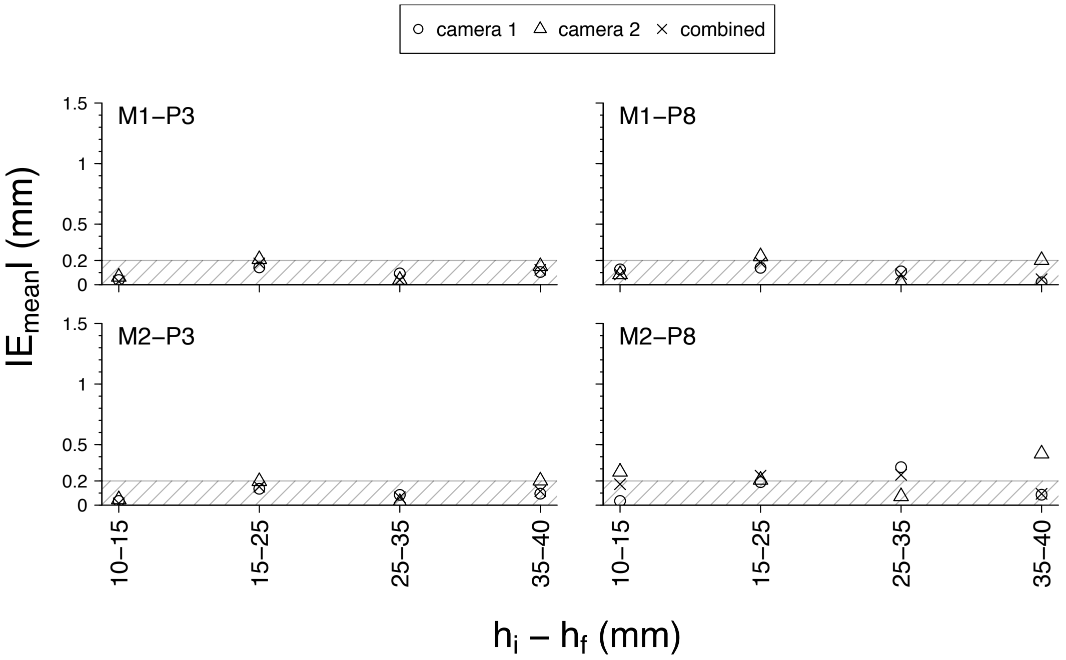


Figure 7.11: Comparison of $|E_{mean}|$ between two different camera positions.

metric (metric 2), we additionally observe a trend in which the variation becomes larger in case 8 coefficients are optimized. This will further be discussed in Section 7.3.4.

Because the image disparity shifts are smaller for camera 2, the relative importance of incorrect feature localization becomes larger. In case the individual optimization results are plotted, camera 2 proves to result in more inclined or irregular surfaces. These are partly optimized to fit the apparent image disparity shifts due to incorrect feature localization. Taking this into consideration, the larger variation (and average error) for 8 optimized coefficients is easily explained. The higher order terms in such a model allow to change the local orientation of the surface more drastically, depending on which image is processed. For camera 1, the relative magnitude of incorrect feature localization decreases compared to the disparity shifts in the image plane. This makes the algorithm more robust and results in more regular surface shapes.

The influence of the average surface height on the accuracy of our method is evaluated by running the reconstruction algorithm for a horizontal water surface at different water levels. The graphs presented earlier in this section show that the influence of the average water level is small and no clear trend can be discerned. We assume that the change between the different water levels is too small to be significant compared to the distance between the water surface and the camera.

7.3.3.3 Numerical validation

The results obtained with the experimental tests are in this section verified using numerical simulations by considering three different surface shapes that will be discussed hereafter.

a. Horizontal water surfaces

We simulated several horizontal water surfaces for which the ‘correct’ water depth was varied between 1 and 10 cm. The camera position was kept constant, with the coordinates of the camera center **c** equaling (5,-215,365) (expressed in mm’s) w.r.t. a world reference frame shown in Figure 7.13 (only a part of the simulated feature pattern is depicted).

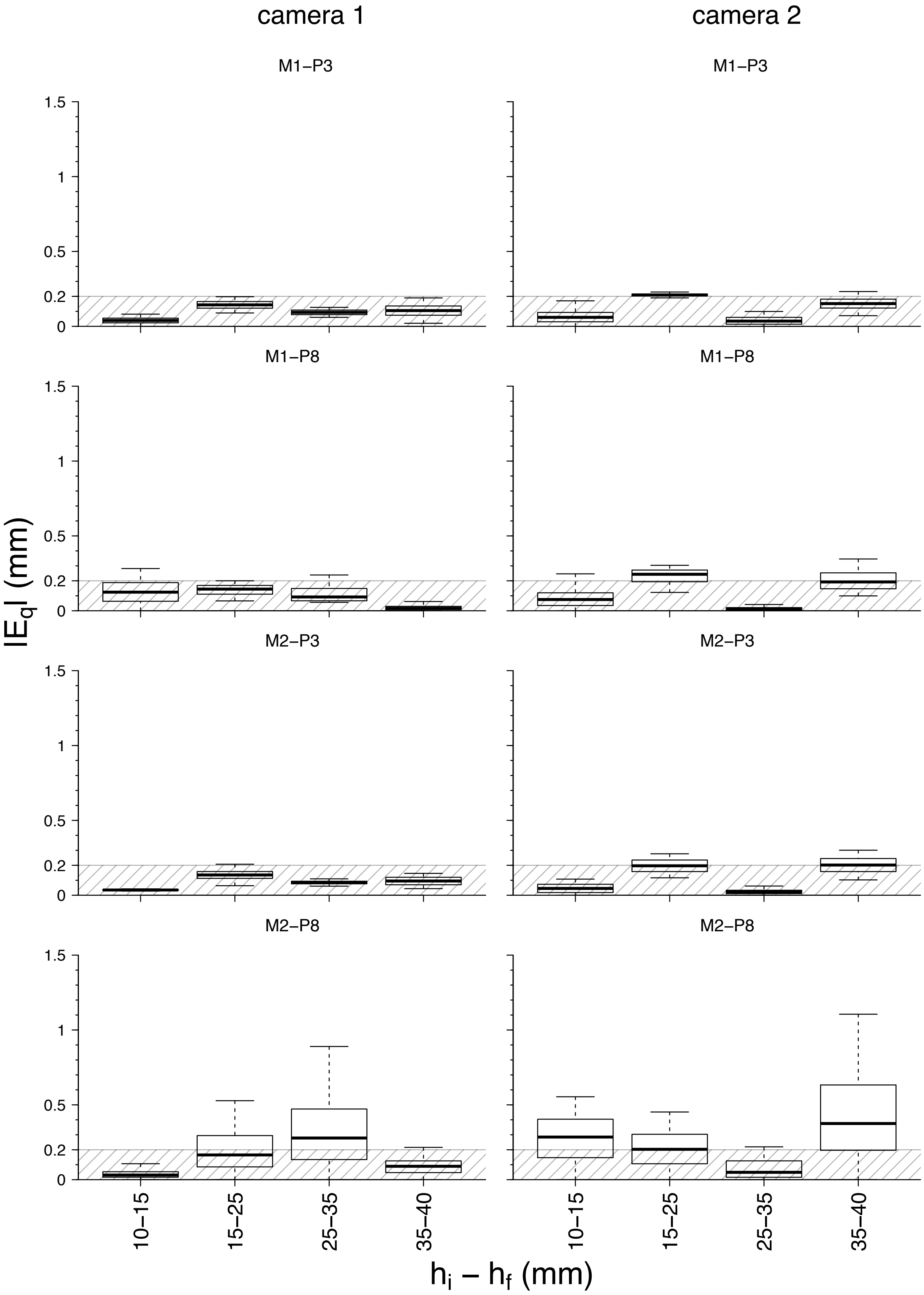


Figure 7.12: Box plots of error $|E_q|$ in domain W for varying camera positions: camera 1 - at height of 27 cm above tank bottom; camera 2 - at height of 38 cm above tank bottom.

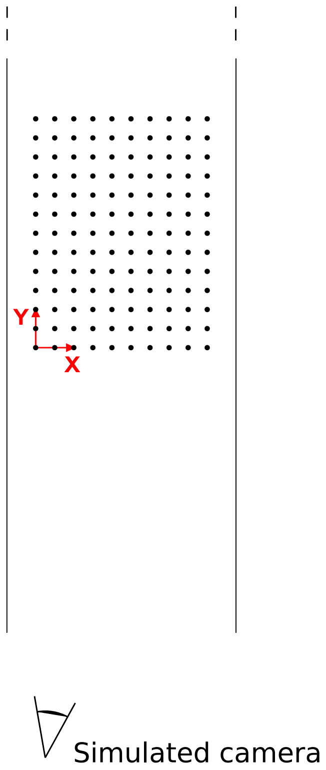


Figure 7.13: Illustration of the simulated camera position w.r.t. the chosen reference frame with origin in the outer corner of the feature pattern on plane F.

Two series of simulations were conducted, in which we shifted the original pixel coordinates based on a $N(0, \sigma)$ distribution with $\sigma = 1$ or $\sigma = 2$. The results of these simulations for the various 'actual' water depths ($A_{00,orig}$) and choices of (M,P) are depicted in Figure 7.14.

For low water depths, incorrect feature detection seems to have an important effect on the accuracy of the reconstruction in case the localization error is high (i.e. $\sigma = 2$). A clear trend is observed in which the magnitude of $|E_{mean}|$ then initially decreases with increasing water depth. This can be expected as the error made in the image plane ($\delta \boldsymbol{q}$) results in an incorrect prediction of the direction of the viewing rays $\overrightarrow{\boldsymbol{u}}$. The corresponding errors on the predicted surface points $\mathbf{p}$ are then linearly related to the distance along $\overrightarrow{\boldsymbol{u}}$ between the camera center and the surface.

At larger water depths, a significant different behavior is noticed between the different surface models. For models containing only 1 or 3 coefficients, the performance of the algorithm does not change much above 30 mm water depth. In contrary, the average error of the P8-models increases after reaching a minimum between 30 and 50 mm water depth. Similarly as was explained earlier, the algorithm uses the 8 parameter surface model to fit a more irregular water surface to the incorrect image disparity shifts. The difference between such models and P1- or P3-models seems however more pronounced at more extreme water depths. We assume that there exists an optimal surface-camera position combination at which the robustness of the algorithm is the largest. In that case, the adopted surface model has only a small influence on the finally obtained result.

In order to gain further insight, we plotted the error (averaged over all iterations) over the entire domain W:

$$|E_q| = \frac{\sum_{iterations} |E_{q,it}|}{100} \tag{7.5}$$

Figure 7.15a and Figure 7.15b show the results for an original water height of 10 and 90 mm respectively, obtained with $\sigma = 2$ in case (M,P)=(1,8) is used. The error increases progressively with increasing y-coordinate. This can be explained by the linear scaling of the simulated localization errors in the image plane with increasing distance along $\overrightarrow{\boldsymbol{u}}$. As is shown on Figure 7.13, surface points with a small y-coordinate are located closer to the camera which results in a smaller error on the locally reconstructed surface shape. It remains however difficult to explain why the reconstruction with 8 coefficients becomes less robust at larger water depths.

Although the absolute error is larger for larger water depths, the relative error compared to the surface height remains approximately the same or even becomes smaller compared to at small water depths. This is particularly noticed in the area located further from the camera (large y-values). For surface points in those areas, the scaling of the error made in the image plane decreases significantly at larger water depths due to the smaller distance of **p** along ray $\overrightarrow{\boldsymbol{u}}$ (more than for the areas located close to the camera). In case we divide $|E_q|$ in that zone by the original chosen water depth (h), this results in the following:

$$\text{h=10:} \quad \frac{|E_q|}{h} = \frac{0.45}{10} = 0.045 \qquad\qquad \text{h=90:} \quad \frac{|E_q|}{h} = \frac{1.55}{90} = 0.017$$

This partly confirms our assumption that at larger water depths, the reconstructed surface can more accurately be reconstructed. For applications in which the absolute water level is important, e.g. actual water depth measurements (instead of oscillations of the water surface), larger water depths can therefore be determined with a larger accuracy.

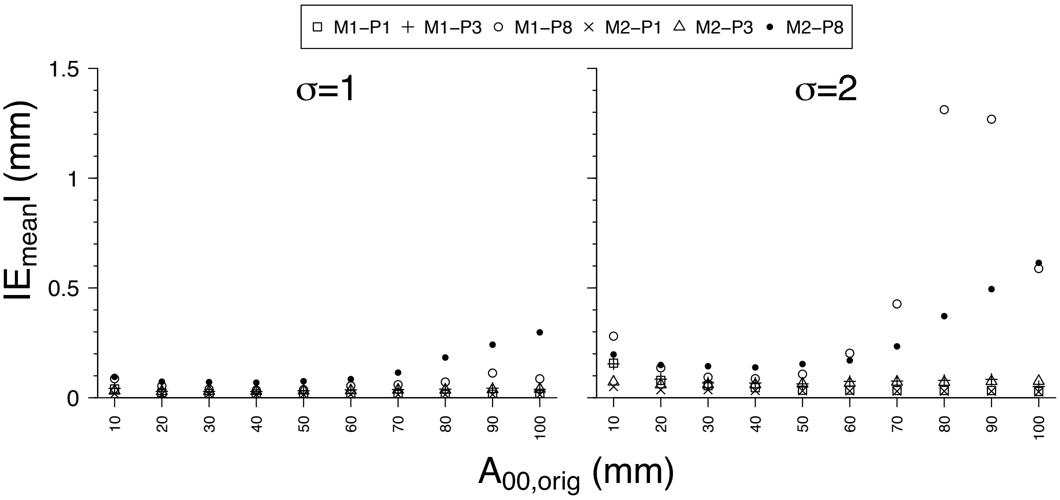
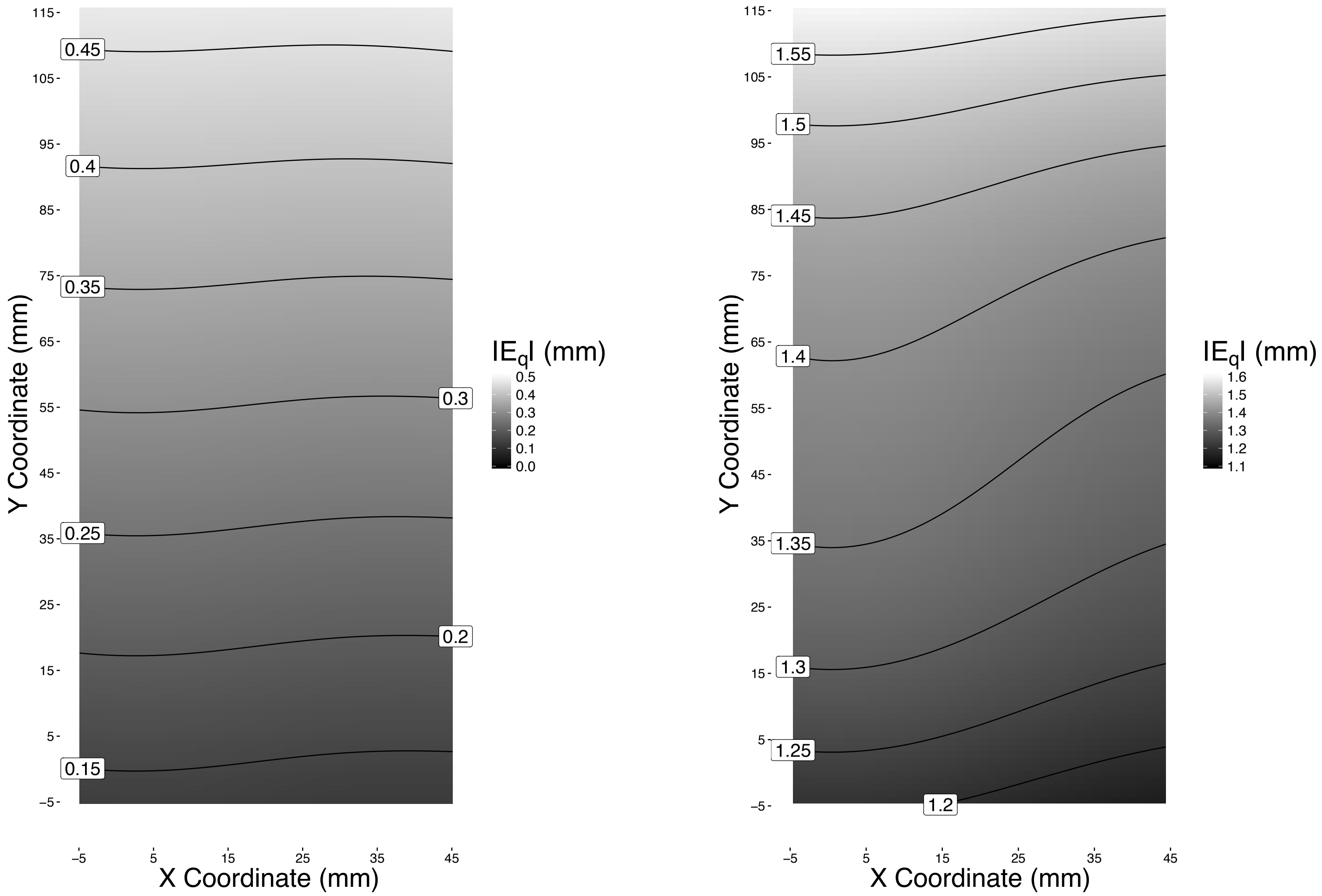


Figure 7.14: $|E_{mean}|$ in function of water depth for a simulation with $\delta\boldsymbol{q} \sim N(0, \sigma)$.



(a) $A_{00,orig} = 10$ mm

(b) $A_{00,orig} = 90$ mm

Figure 7.15: Comparison of error $|E_q|$ within W for (M,P)=(1,8) and $\delta \boldsymbol{q} \sim N(0,2)$. Note that the graphs have a different scale.

b. Inclined water surface

The second series of numerical simulations comprised an original water surface using a constant term $A_{00,orig} = 40$ mm but with a varying linear inclination in the y-direction (coefficient C). The reconstruction algorithm was then applied with two different surface models: a surface model with 3 coefficients (A_{00}, B and C) and a surface model incorporating all 8 surface coefficients.

Figure 7.16 shows that although the mean error increases with larger values of $\delta\boldsymbol{q}$, no clear trend can be discerned. It is assumed that this is related to the small changes in inclination, for which the resulting change of points **p** remains too small to have a significant effect. Such inclinations are however in the range of what can be expected during experiments of self-induced sloshing. In practical applications of the algorithm, smoothly inclined surface shapes should therefore not impose a problem.

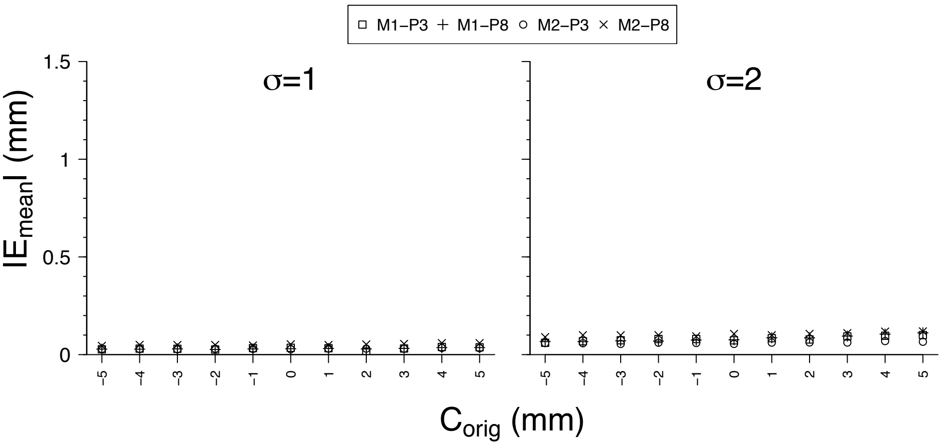


Figure 7.16: $|E_{mean}|$ in function of inclination of the water surface for a simulation with $\delta\boldsymbol{q} \sim N(0, \sigma)$.

c. Curved water surface

A final group of simulations was conducted in which the first order cosine term was varied while maintaining the constant term $A_{00,orig} = 40$ mm. Figure 7.18 shows $|E_{mean}|$ in which the coefficient A_{01} (corresponding to the longitudinal y-direction of the tank) was varied between -10 mm and 10 mm and a surface model with 8 coefficients was adopted. For large negative values of A_{01}, the error increases significantly. This can be explained in case we consider the camera position w.r.t. the reconstructed surface.

Figure 7.17 shows a schematic representation of the simulated test case. A large negative value of A_{01} causes that the angle between the viewing rays and the surface normal decreases. As was explained in Section 7.3.3.1, changes of the water surface in the same direction of the viewing ray result in smaller disparity shifts in the image plane. Similarly, a certain movement of the image points can only be explained by a larger change of the local surface inclination or water depth. As a result, the deviation of the reconstruction $\eta'(x,y)$ (based on $\boldsymbol{q'} + \delta\boldsymbol{q}$) w.r.t. the original surface $\eta(x,y)$ then rapidly increases.

This clearly shows that choosing an appropriate location of the imaging equipment, depending on the surface that needs to be reconstructed, can significantly increase the performance of the algorithm. Multiple cameras could therefore even further improve the reconstruction accuracy. Each zone of the surface can in that case be reconstructed based on the camera for which the viewing angle w.r.t. the local surface shape is optimal. Due to the limited time available in this

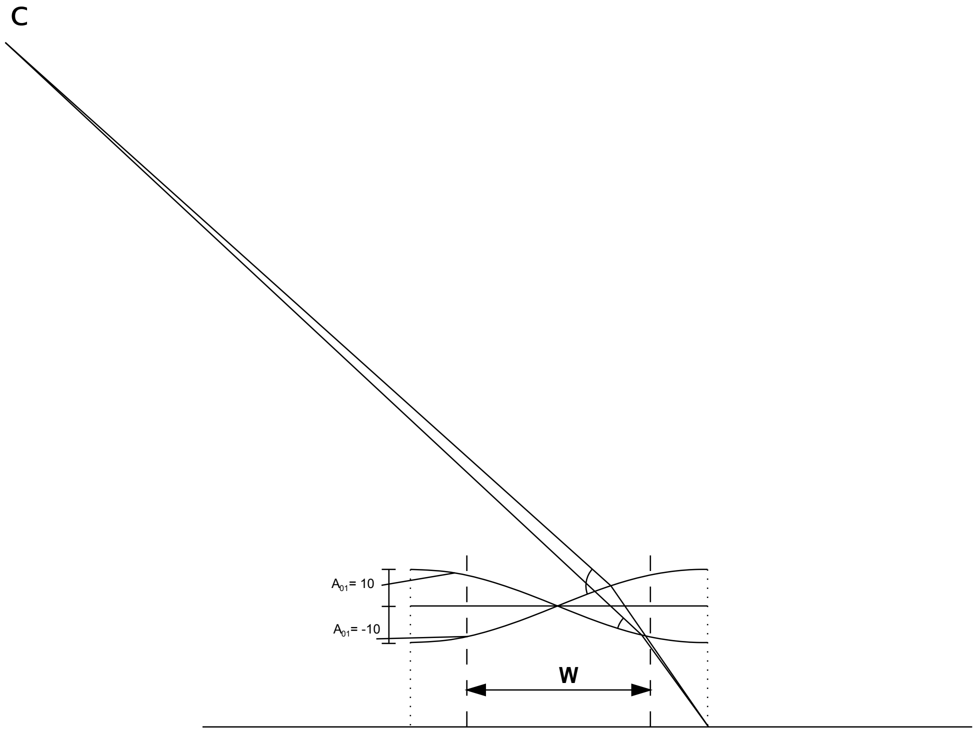


Figure 7.17: Illustration of the simulation of a curved surface. Viewing rays intersect the surface at more grazing angles in case of $A_{01}\,(mm) > 0$, which makes the algorithm more robust and accurate.

thesis, this is however not yet incorporated in the current developed algorithm.

7.3.4 Influence of the reconstruction configuration: error metric & surface model

7.3.4.1 Theoretical considerations

As explained in Chapter 6, two different error metrics can be used in our optimization procedure: the normal collinearity metric and disparity difference metric. Both use the dissimilarity between normal sets $\overrightarrow{\boldsymbol{n_1}}$ and $\overrightarrow{\boldsymbol{n_2}}$ as error measure for the assumed water depth, corresponding with the current estimate of the surface function coefficients. Morris (2004) states that the second metric gives better results in case of very small water depths. For shallow water depths ($O(0.3\,mm)$), the effect of the surface normal on refraction becomes insignificant. He suggests that the disparity difference metric models this by relating the error to the water depth. Large normal differences at low depths cause in that case a smaller error E_{disp} than the same normal difference at larger water depths. For larger water level differences, he states that the difference between the performance of both metrics becomes small. Because we use the algorithm for significantly larger water depths than 0.3 mm, it remains difficult to predict which error metric has the largest potential to obtain accurate and robust results.

Additionally, different surface models are used to describe the actual water surface. As explained in Section 7.2.1, a model which contains three coefficients is in theory perfectly able to describe the water surfaces established during the experiments. In case additional terms are incorporated, the higher order terms are used to locally adjust the surface shape to mitigate the errors related to feature localization, camera calibration and pose estimation. Higher order models have therefore too many degrees of freedom, making them more sensitive for the optimization input. Although

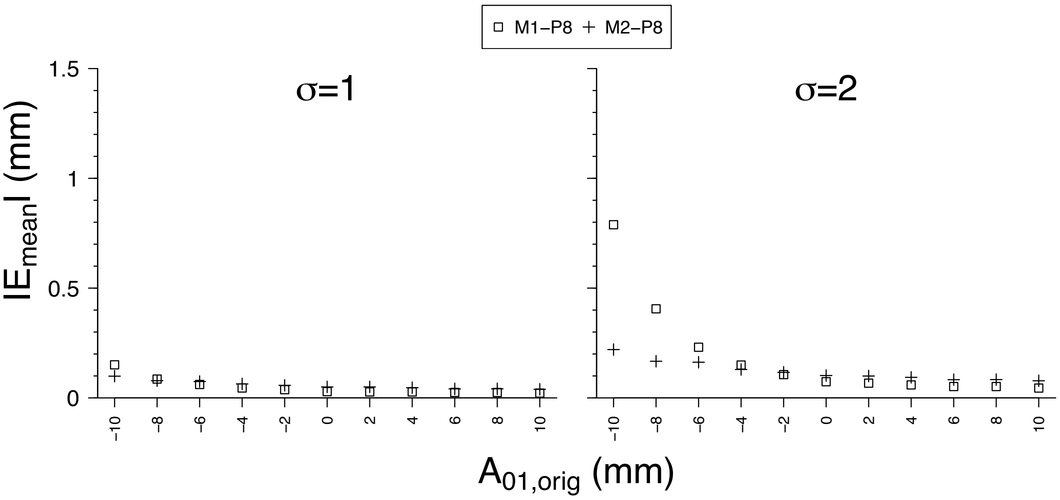


Figure 7.18: $|E_{mean}|$ in function of curvature of the water surface for a simulation with $\delta\boldsymbol{q} \sim N(0,\sigma)$.

a more complex and locally varying surface might give a smaller total error E_{tot}, the obtained surface shape is in that case a suboptimal representation of reality.

In the numerical simulations, we will additionally study more complex surface shapes. Similarly as in the experiments, we expect that adding redundant terms will make the reconstruction less robust to the Gaussian noise $\delta\boldsymbol{q}$.

7.3.4.2 Experimental validation

The graphs depicted earlier in Figure 7.7 show that no clear correlation can be found between $|E_{mean}|$ and the reconstruction configuration that is used. We therefore apply our second methodology in which box plots are considered. Figure 7.19 shows these box plots using different configurations for three separate cameras. Although the mean error usually falls within our measurement accuracy range of 0.2 mm, two significant trends can be discerned.

Firstly, the surface model incorporating 8 coefficients shows usually a much larger variation and also higher peak values of $|E_q|$. Our assumption that a surface model with too many degrees of freedom becomes less robust is therefore experimentally confirmed. Hence, initial knowledge about the phenomenon or water surface that needs to be reconstructed can significantly improve the performance of our method.

Secondly, no clear difference between both metrics is present in case only three coefficients are optimized. The disparity difference metric (M2) seems however more sensitive for the optimized surface model. In case 8 surface coefficients are optimized, metric 2 results in some cases in less reliable and more irregular surface shapes.

7.3.4.3 Numerical validation

A first evaluation based on numerical simulations is done by considering Figure 7.14 in which horizontal water surfaces are simulated. In case the error on the pixel coordinates remains small (i.e. $\sigma = 1$), the influence of the chosen surface model is only significant at larger water depths. In that case, surface models with 8 coefficients deviate more from the original horizontal simulated

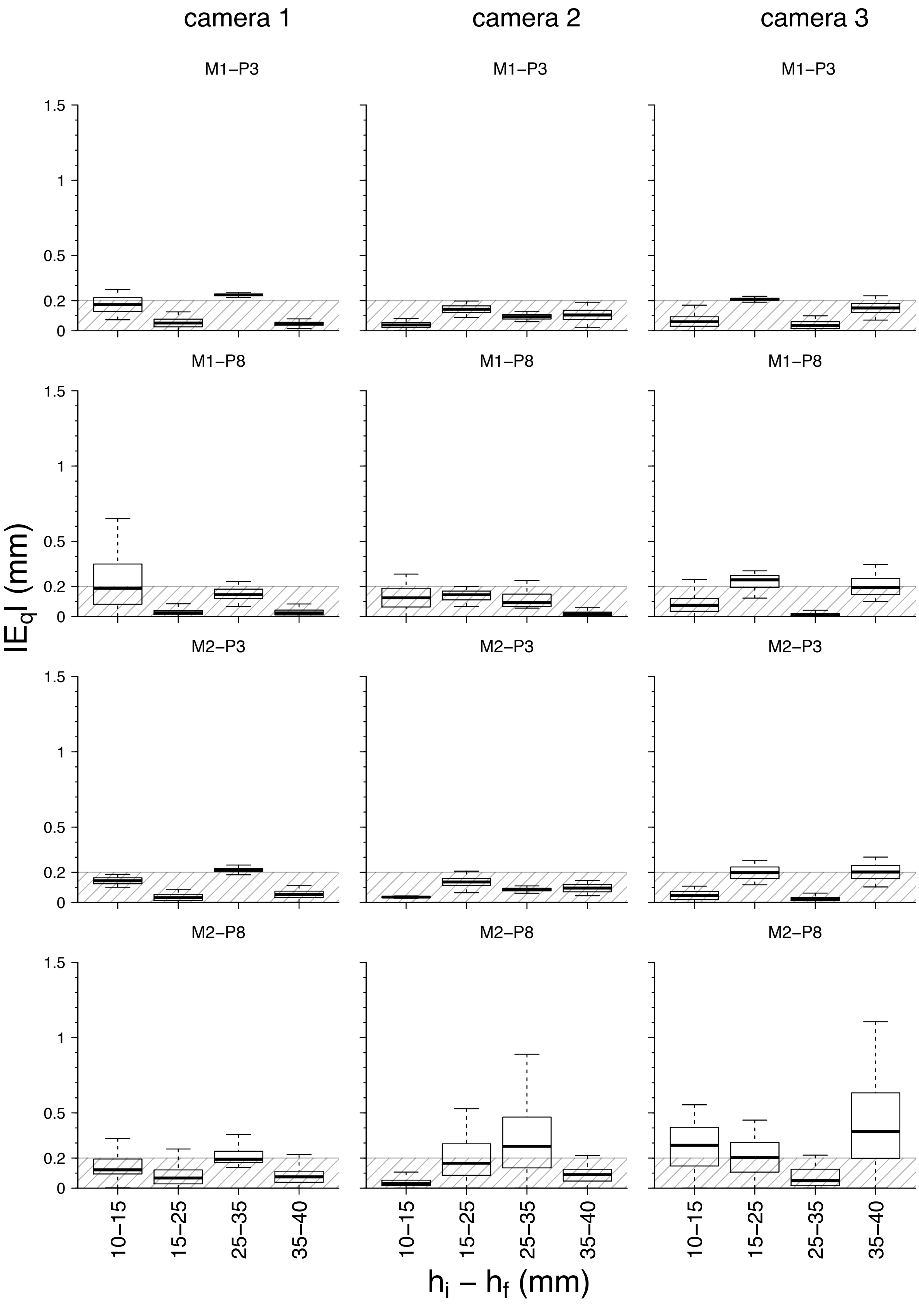


Figure 7.19: Box plots of error $|E_q|$ in domain W for varying reconstruction configurations.

surface. In case of larger shifts $\delta \boldsymbol{q}$, this effect becomes more pronounced. Inaccurate feature localization can therefore be mitigated by reducing the amount of coefficients in the surface function. Prior knowledge about the expected surface shape is then obviously required.

Additionally, the normal collinearity metric (M1) seems to be more robust compared to the disparity difference metric in case the localization errors remain limited. For higher localization errors, the opposite is observed and the first metric then sometimes diverges from the actual solution. A reason for these local peaks is however difficult to derive. Using a more advanced optimization algorithm could probably solve this issue in later stages of the algorithm's development.

If we consider Figure 7.18, corresponding to a more complex surface shape, no real difference seems present between the two possible error metrics in case the localization error remains small. For larger deviations, metric 2 seems to be more accurate and robust in case the viewing rays intersect the surface at larger angles ($A_{01} \ll 0$). For a more grazing angle ($A_{01} > 0$), the difference decreases rapidly. In case of accurate feature localization ($\sigma = 1$), the collinearity metric even performs better than the disparity difference metric. Because the range in which $|E_{mean}|$ is in that case similar as what was observed during the experiments, the numerical simulations confirm what was concluded in Section 7.3.4.2.

7.3.5 Influence of the amount of cameras used

7.3.5.1 Theoretical considerations

Although the cameras that are used are equipped with a wide angle lens, their view remains limited. In case full resolution images are taken (without reducing the acquisition window width), the cameras' horizontal and vertical angle of view are respectively $\alpha_h = 29.6°$ and $\alpha_v = 16.9°$. In practice, this means that to depict an object completely in the image, the camera needs to be positioned sufficiently far from the object. In Section 7.3.3, we showed however that the reconstruction accuracy decreases in case the camera is positioned further from the water surface. For this reason, we limit the height at which the camera is positioned (for the tests discussed in this thesis $\approx$ 30-40 cm above the feature plane). Figure 7.20 shows that combining multiple cameras in that case allows to reconstruct a larger surface area by optimizing their views. Each zone is then covered by at least one camera and can be incorporated in the optimization of the surface coefficients.

In view of accurate and reliable surface reconstruction, combining two or more cameras offers additional advantages:

- The number of feature points is doubled.
  The minimization procedure is applied to fit a theoretical surface model (see Chapter 4) to twice as many projections of feature points on the water surface ($\mathbf{p}$ in Chapter 6). Errors in the detection of some feature points contribute therefore relatively less to the total error $E_{tot} = \sum_f E_f^2$. Inaccuracy due to post-processing, noise and the limited depth of field is by this means partly mitigated.

- Inaccuracy related to a single camera can be mitigated.
  Inaccuracies related to calibration and initialization of one camera can partly be corrected by using the other camera. Even if the individual errors for every camera are the same, the combination of both usually improves the final result.

- Loss of feature points can more easily be handled.
  Loss of feature points during tracking over the captured image sequence can be dealt with

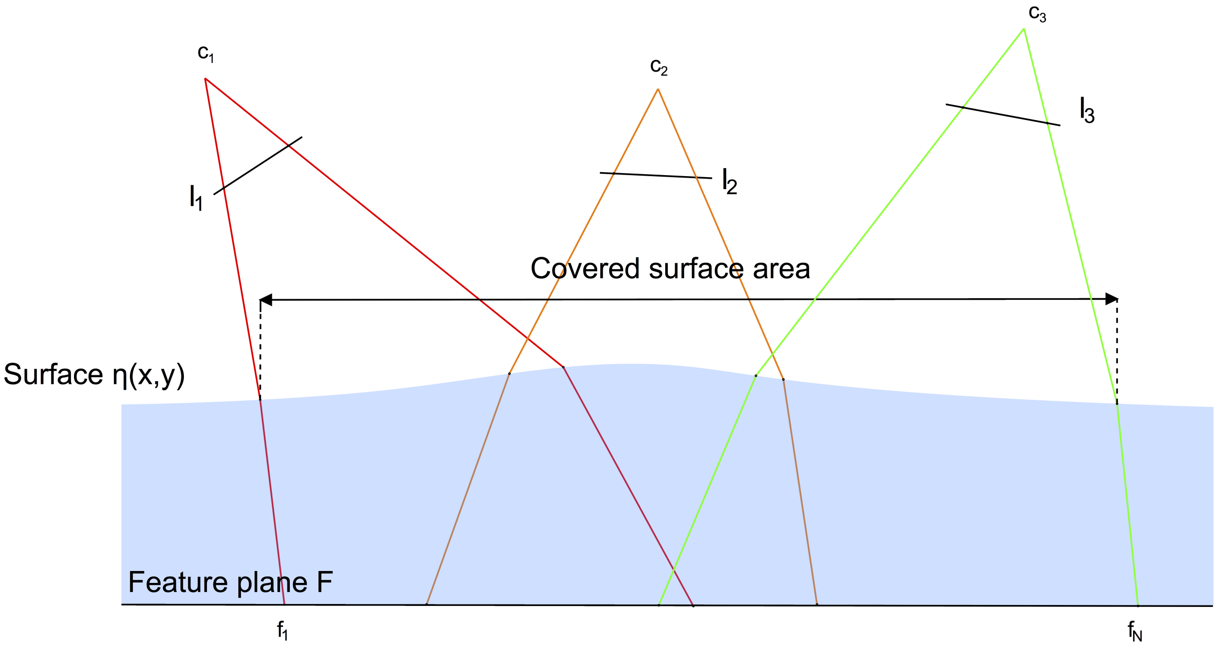


Figure 7.20: The spatial extent of the surface reconstruction can be enlarged by using multiple cameras.

more easily. Loss of feature points is caused by motion blur or large movements of features between two successive frames. This problem is only relevant for actual time-dependent surface reconstructions and will be discussed in Chapter 8.

- Several zones of the water surface can be optimized separately.
  As was explained in Section 7.3.3.3, there exists an ideal position of the camera w.r.t. the surface for which the accuracy and robustness of the algorithm is maximized. In case multiple cameras are used, each camera view can reconstruct that area for which the distance and angle of the viewing rays are optimal. This is however difficult to implement in practice and therefore not yet incorporated in the current developed algorithm.

7.3.5.2 Experimental validation

In order to quantify the advantage of multiple cameras, the results obtained with an optimization in which a single, two or three cameras is used are compared. Because previous sections showed that no conclusions can be drawn based $|E_{mean}|$, we use the same methodology of box plots to quantify the robustness and accuracy of the algorithm.

In Figure 7.21, the variability of the individual errors for a varying amount of cameras is depicted. The first column corresponds to an average of three optimizations, each using only a single camera. In the second column, the scatter obtained by adopting (two times) two cameras is shown. Finally, an optimization with all three cameras combined is given in the third column.

Comparison between these three cases indicates that the scatter decreases and the robustness of the surface reconstruction increases by using multiple cameras. Although no conclusions can be drawn based on the average error (located within the accuracy range of the level gauge), an optimization with three cameras causes much less variation on the individual errors $|E_q|$. Combining several cameras allows in most cases to mitigate the errors made in feature localization and errors related to the camera calibration and initialization. We observed that the reconstructed surfaces become more flat with smaller (incorrect) higher order terms. The surface shape is less locally adapted because it should fit the image disparity shifts in multiple images. Because feature localization errors are therefore more difficult to mitigate with higher order terms, E_{tot} might in that case even become larger. Even so, the obtained result is a much better representation of reality.

Additionally, Figure 7.21 confirms that the maximal error and variability of the individual errors are larger in case 8 surface coefficients are used. It can also be observed that the performance of the disparity difference metric (M2) is lower for most of the tested cases. This aspect is however less pronounced for the combination of three cameras, which can be explained by the mitigation-effect as was explained above.

7.3.5.3 Numerical validation

No time was available to study the effect of multiple cameras using a numerical approach. This aspect can however easily be investigated in further research by incorporating a second camera in the numerical simulations. Similarly as was done earlier, random noise can be added to the pixel coordinates of the second camera using a sample out of $N(0, \sigma)$. We expect that the combination of two cameras will allow to leverage the shifts $\boldsymbol{\delta q}$ of a single camera and make the reconstruction more robust.

7.4 Conclusion

In this chapter, we validated our developed C++ program in both experimental and numerical tests. The developed algorithm proved highly accurate and was for most cases able to reconstruct the surface within the accuracy that can be obtained with e.g. a level gauge.

We investigated the most crucial aspects w.r.t. the performance of the algorithm. This showed that feature localization is the most crucial step in order to obtain high accurate results. In case sufficient care is taken to obtain strong and uniform illumination of the captured feature plane, images with both UV and visible light gave good results. Camera calibration and camera pose estimation seemed in contrast to have a small influence on the reconstruction result.

A second important aspect of the reconstruction is the position of the camera w.r.t. the reconstructed surface. For cameras that view the surface at grazing angles, higher accuracy was possible in case accurate feature localization was obtained. In case the angle of the viewing rays with the surface decreased, a surface model with redundant coefficients became however less robust to incorrect pixel coordinates of the feature points. The local surface shape was then fitted to explain the apparent movements in the image plane that are related to the localization errors.

We finally concluded that combining multiple camera significantly improves the accuracy and robustness of the reconstruction. Inaccuracies related to a single camera can then be mitigated by the other cameras. The obtained surface reconstruction proved less affected by inaccurate feature localization and was a better approximation of reality.

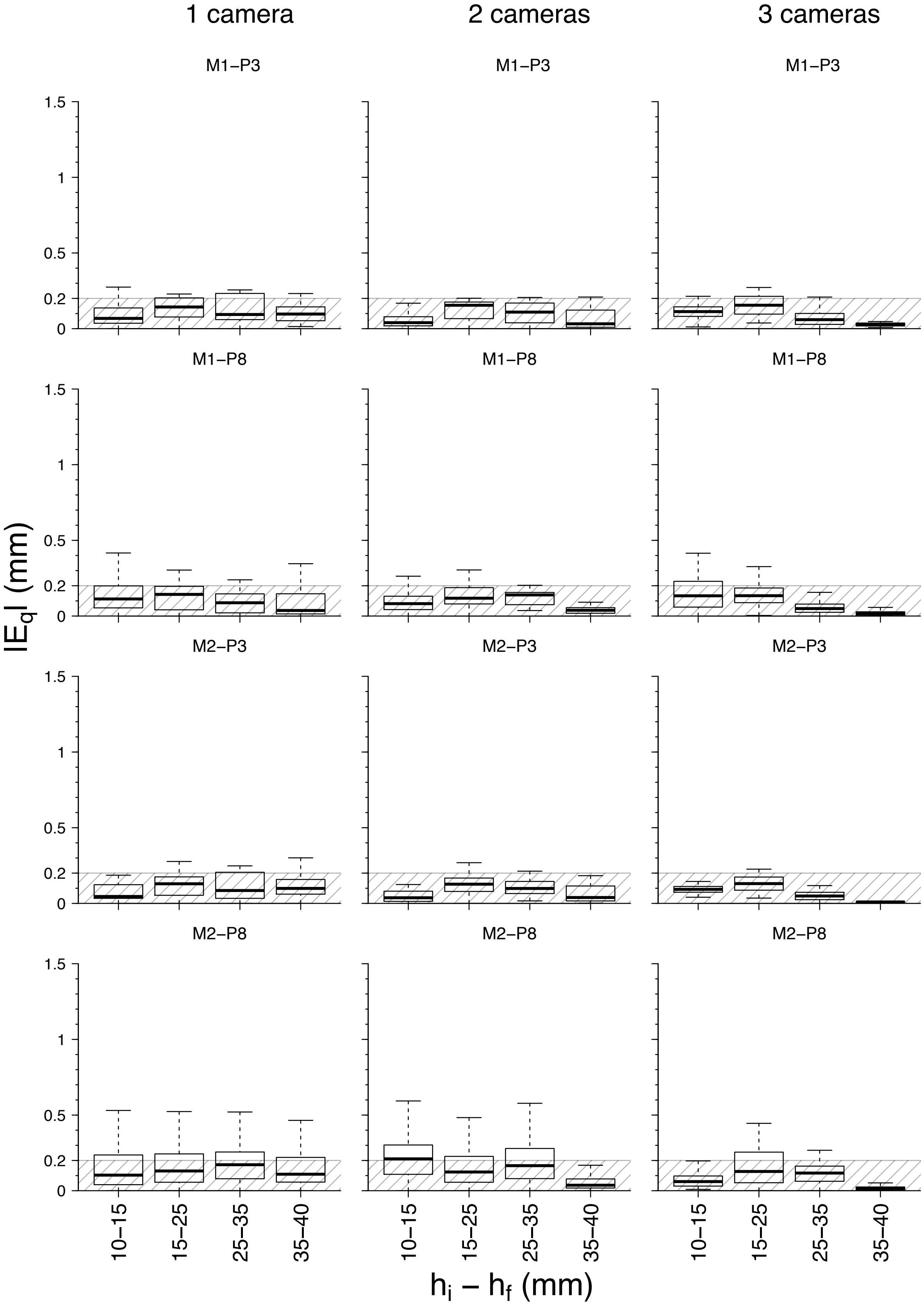


Figure 7.21: Box plots of error $|E_q|$ in domain W. In each column, a different amount of cameras is used as input for the reconstruction algorithm.

Chapter 8

Spatio-temporal reconstruction of the water surface shape

In Chapter 5, we presented a method to detect feature points in an image. The suggested algorithm allows to detect ‘blobs’ in the image and obtain the pixel coordinates of the centers of gravity of each of these dots. Using our sorting algorithm described in Chapter 6, the detected image points could be ordered in a systematical way. In the previous chapter, this method was applied for all ten images that depict the same water surface. In case of dynamically changing water surfaces, the same method is applied for an image of the feature pattern without water as well as for the first processed image of the image sequence. The feature detection algorithm that is used during initialization is however not suited to use for the entire image sequence because of several reasons.

Firstly, both the OpenCV feature detection algorithms and our developed sorting algorithm are not time efficient. Additionally, sorting of the corners is currently not yet automatized and requires the user to manually indicate the four outer corners of the pattern. Further improvements of the developed algorithm could partly mitigate this last problem. The time needed for a single image becomes therefore not longer acceptable in case an entire sequence of images needs to be processed. For this reason, we searched different solutions to detect feature points in image sequences that can contain several hundreds or even thousands of images.

Secondly, it proved difficult to handle images in which the feature pattern is largely distorted or in which a subset of the feature points is even no longer visible. The latter can be caused by motion blur, unsatisfactory lighting conditions or physical limits related to refraction of light. The latter can occur in case an initial sharp viewing angle of the camera is combined with large curvatures of the surface. The critical angle of the air-water interface (as explained in Chapter 4) might then be exceeded, causing that some rays originating from the feature points do not reach the camera center. We therefore need a method to handle occlusions of features wherein knowledge about which points are not detected is important. This aspect will further be discussed in Section 8.2.

Similarly as was proposed in the original method of Morris (2004), we incorporated feature tracking methods that are based on the principle of optical flow. In Section 8.1, the theory behind the optical flow method that will be used in this thesis is given. In Section 8.2, a detailed description of the implementation in our C++ script will then be elaborated. We finally test this implementation in a first experiment of the filling process of the test tank.

8.1 Theoretical background

8.1.1 Optical flow

Optical flow has been studied extensively in literature and many different approaches to compute optical flow have been developed (Beauchemin and Barron (1995)). In this chapter, the basics of optical flow are elaborated to obtain a general understanding necessary for the application of feature tracking methods.

Two main approaches have been developed to track points across a video stream or sequence of images (Bradski and Kaehler (2008)): dense and sparse optical flow methods. Both approaches rest on the same basic principle to analyze an image sequence of a three-dimensional scene in which objects are moving w.r.t. the camera. Each point of the 3D surface then moves along a 3D path $\mathbf{X}(t)$ in the camera reference system. In case this motion is projected on the image plane, this results in a two-dimensional path $\mathbf{q}(t) \equiv [x(t),y(t)]^T$. The 2D velocities are subsequently derived by taking the derivative w.r.t. time: $\frac{d\mathbf{q}(t)}{dt}$. These discrete velocity vectors are then used to obtain a global 2D motion field. Optical flow is used to compute an approximation of this motion field based on the information contained in the time-varying image intensity.

The first approach, denoted as 'dense optical flow methods' or 'global methods', considers the motion (velocity) of every pixel in the images. These velocities can be computed based on the distance a pixel has moved between successive frames, which results in a velocity field for all pixels in the image. Two of the mostly used dense optical flow algorithms are 'The Horn-Schunck method' (Horn and Schunck (1981)) and the 'Block matching method' (Huang and Zhuang (1995)).

The first method uses a brightness constancy assumption to derive the brightness constancy equations. A solution for these equations is subsequently found by iteration and imposing a hypothesized smoothness constraint on the pixel velocities in both directions. This means that the neighboring pixels are assumed to have similar velocities, which results in a progressive optical flow variation. The second method considers (usually overlapping) blocks of pixels and computes the motion of these blocks. By matching these windows between consecutive frames, a velocity vector for every central pixel in the window/block can be obtained.

Although dense optical flow methods provide more detailed information about the image, a problem arises because some points in the image are difficult to track (e.g. a point on a white piece of paper). Interpolation between distinguishable points is therefore required to solve for those points that are ambiguous. This however severely increases the computational cost. Additionally, it has been observed that these methods have a low efficiency in case of small motions.

For this reason, most applications adopt sparse optical flow methods. These 'local methods' specify beforehand a subset of distinguishable points in the image that are easy to track and for which the algorithm is applied. Tracking of distinguishable points make these methods not only more robust and reliable but also much less computationally demanding compared to dense optical flow methods. In this thesis, we are only interested in the image points of the discrete feature pattern in every frame of the image sequence. Bearing this in mind, a sparse optical flow method seems more suitable for our application.

In literature, several sparse optical flow methods have been derived. The Lucas-Kanade (LK) method, proposed by Lucas et al. (1981), is the most widely used variant. As already explained, a subset of points with certain desirable properties is specified beforehand and tracked across the image sequence. Local information in a small window surrounding each point of interest is then used to solve the optical flow equation. Using information in a local neighborhood of the pixel creates on the other hand a risk that points move out of these small, local windows in case of large motions. This makes them impossible to find in the next image that is processed. In order to handle these large motions, 'a pyramidal LK algorithm' has been developed which will be explained in Section 8.1.3. Because this relies on the method presented by Lucas et al. (1981), we start our theoretical framework with the derivation of the basic LK algorithm.

8.1.2 The traditional Lucas-Kanade algorithm

The LK algorithm is based on three assumptions:

- **Color/Brightness constancy constraint.**
  This is the basic assumption for most optical flow methods. It means that a pixel corresponding to a particular point of a 3D object does not change in appearance between different frames. In case of grayscale images, this implies that the brightness value of the pixel corresponding to one 3D point does not change when this point is tracked across the image sequence. For simplicity's sake, we will only consider grayscale images in the rest of this theoretical derivation. The algorithm can however easily be adapted to color images by applying the same methodology for every color channel of the image.
- **Temporal persistence.**
  This assumption requires the image motions of 3D points to remain small between consecutive frames. In practice, this requirement of 'small movements' is usually fulfilled by making the temporal increments small enough relative to the scale of the image motion.
- **Spatial coherence.**
  Spatial coherence implies that neighboring 3D points, belonging to the same surface in the scene, have a similar motion and are projected to nearby points in the image plane.

Based on assumption 1, the optical flow equation can be derived:

$$I(x,y,t) = I(x + \Delta x, y + \Delta y, t + \Delta t) \tag{8.1}$$

where:

$\Delta x, \Delta y$	The spatial displacements of the image pixel in the horizontal and vertical direction.
Δt	The time difference between two consecutive frames.
$I(x,y,t)$	The intensity/gray level of pixel (x,y) at moment t.

In case it is assumed that the movements are small, the image intensity $I(x+\Delta x, y+\Delta y, t+\Delta t)$ can be approximated by following Taylor series:

$$I(x + \Delta x, y + \Delta y, t + \Delta t) = I(x,y,t) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t + H.O.T \tag{8.2}$$

By combining equations 8.1 and 8.2 and neglecting the higher order terms (H.O.T), following expression is obtained:

$$\frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t = 0 \tag{8.3}$$

or

$$\frac{\partial I}{\partial x}\frac{\Delta x}{\Delta t} + \frac{\partial I}{\partial y}\frac{\Delta y}{\Delta t} + \frac{\partial I}{\partial t}\frac{\Delta t}{\Delta t} = 0 \tag{8.4}$$

This can be expressed more shortly as:

$$\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} = 0 \tag{8.5}$$

$$I_x u + I_y v = -I_t \tag{8.6}$$

$$\nabla I^T \cdot \boldsymbol{d} = -I_t \tag{8.7}$$

where:

$\boldsymbol{d}$	The optical flow or 2D velocity, equal to the shift of the pixel considered ($= [\frac{dx}{dt},\frac{dy}{dt}]$) in pixels per frame.
u,v	The x- and y-components of the 2D velocity $\boldsymbol{d}$.
I_x, I_y	The derivatives of the intensity I at (x,y,t) in the x- and y-direction.
I_t	The time derivative of the intensity I at (x,y,t).
∇I	The gradient vector of I at (x,y,t) ($= [\frac{\partial I}{\partial x},\frac{\partial I}{\partial y}]$).

This equation presents the optical flow constraint w.r.t. movement. It contains however two unknowns, u and v, and can thus not be solved directly. Measurements at single-pixel level are therefore underconstrained and a unique solution for the 2D-motion seems not possible. This ambiguity is also known as the aperture problem, which arises when the aperture or window in which motion is measured becomes too small. For large motions that are evaluated in small apertures, edges are observed instead of corners as shown in Figure 8.1. Only motion perpendicular ('normal') to these lines can then be discerned.

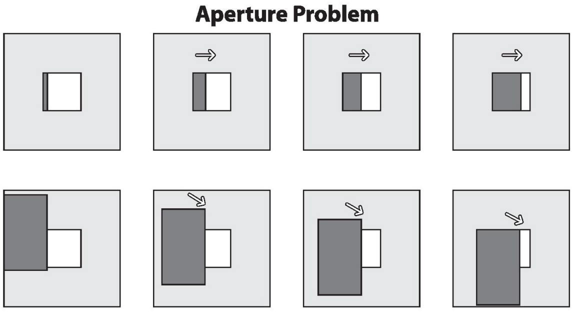


Figure 8.1: Aperture problem: through the aperture window (upper row) we see an edge moving to the right but cannot detect the downward part of the motion (lower row). From Bradski and Kaehler (2008).

The same is valid for optical flow algorithms, in which only the motion normal to the line described by eq. (8.6) can be determined. Figure 8.2 gives a graphical representation of the aperture problem applied to optical flow. To solve this issue, all optical flow methods make certain hypotheses to derive additional constraints that allow to derive the optical flow of the pixel points.

The LK method assumes that the optical flow is constant in a local neighborhood/window around the pixel $\boldsymbol{x} = [x,y]^T$ under consideration. The choice of the adopted window determines the performance of the algorithm. In case the window size is too large, the assumption of spatial coherence will be violated. Additionally, a small window is required in order not to smooth out the details in the image. One the other hand, a window size that is too small does not solve the ambiguity problem. In particular, a large window would be preferable to obtain a robust method that is able to handle large motions. A compromise between both requirements needs therefore

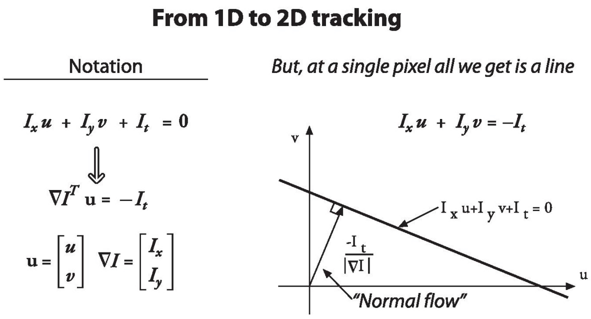


Figure 8.2: Aperture problem for the two-dimensional optical flow. From Bradski and Kaehler (2008).

to be made. In case the search interval in both the positive and negative x- and y-direction is denoted ω_x and ω_y, this results in a total window size $(2\,\omega_x + 1 \times 2\,\omega_y + 1)$. Typical values for ω_x and ω_y are 2, 3, 4, 5, 6, 7 pixels. This allows to set up $(2\,\omega_x + 1 \times 2\,\omega_y + 1)$ equations to solve eq. (8.7) for all pixels in the window centered around the point $\mathbf{x}$:

$$
\begin{aligned}
I_x(q_1)u + I_y(q_1)v &= -I_t(q_1) \\
I_x(q_2)u + I_y(q_2)v &= -I_t(q_2) \\
&\vdots \qquad (8.8) \\
I_x(q_n)u + I_y(q_n)v &= -I_t(q_n) \qquad (8.9)
\end{aligned}
$$

in which:

q_1,q_2, … ,q_n	The pixels inside the window $(2\,\omega_x + 1 \times 2\,\omega_y + 1)$.
$I_x(q_i)$,$I_y(q_i)$,$I_t(q_i)$	The partial derivatives of the image I with respect to x- and y-direction and time t, evaluated at the point q_i and at the current time.

These equations can also be written in matrix form:

$$A\boldsymbol{d} = \boldsymbol{b} \tag{8.10}$$

where:

$$
A = \begin{bmatrix} I_x(q_1) & I_y(q_1) \\ I_x(q_2) & I_y(q_2) \\ \vdots & \vdots \\ I_x(q_n) & I_y(q_n) \end{bmatrix}, \quad \boldsymbol{d} = \begin{bmatrix} u \\ v \end{bmatrix}, \quad \text{and} \quad \boldsymbol{b} = \begin{bmatrix} -I_t(q_1) \\ -I_t(q_2) \\ \vdots \\ -I_t(q_n) \end{bmatrix}
$$

This system has more equations than unknowns and thus it is usually overdetermined (in case the window contains more than 1 edge). The LK algorithm therefore finds a least-squares solution of eq. (8.10), in which a 2×2 system is solved to find $min(A\boldsymbol{d} - \boldsymbol{b})^2$:

$$A^T A\boldsymbol{d} = A^T\boldsymbol{b} \quad \text{or} \tag{8.11}$$

$$\boldsymbol{d} = (A^T A)^{-1} A^T \boldsymbol{b} \tag{8.12}$$

Writing this in full, the obtained expression becomes:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum_i I_x(q_i)^2 & \sum_i I_x(q_i)I_y(q_i) \\ \sum_i I_y(q_i)I_x(q_i) & \sum_i I_y(q_i)^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i I_x(q_i)I_t(q_i) \\ -\sum_i I_y(q_i)I_t(q_i) \end{bmatrix} \tag{8.13}$$

in which the sums are running from i=1 to n. In this expression, the first term resembles the structure tensor of the image at point **x** (with a window function =1 for all n points), which was already explained in Chapter 5. This matrix is further denoted as G:

$$G = \begin{bmatrix} \sum_i I_x(q_i)^2 & \sum_i I_x(q_i)I_y(q_i) \\ \sum_i I_y(q_i)I_x(q_i) & \sum_i I_y(q_i)^2 \end{bmatrix} \tag{8.14}$$

which is computed for every window/patch of the image and contains 4 elements that are calculated based on the spatial derivatives I_x and I_y. That is why this matrix is also known as the matrix of spatial gradient or gradient matrix.

A solution for this expression is however only possible in case $G = A^T A$ is invertible. This is not the case if the pixel $[x,y]^T$ is located in a region with no structure (e.g. if I_x, I_y and I_t are all zero for every pixel in the neighborhood). This corresponds with a window in which no two edges are present. Even if G is invertible, the matrix can still be ill-conditioned if its elements are very small and close to zero. A method to test the structure tensor G, for which a full rank of 2 is required, is based on the eigenvalues of this matrix: $\lambda_1 \geq \lambda_2 > 0$. To avoid noise issues and an ill-conditioned tensor, λ_2 is usually required not to be too small. For points located on an edge, the method will also not be able to estimate the optical flow components in two directions. This requires that λ_1/λ_2 is not too large as was explained in Chapter 5.

It can be concluded that the LK-algorithm only works well in case λ_1 and λ_2 are large enough and have similar magnitude. We recall that this requirement is also used by the Harris corner detector. In case the tracked feature points are initialized using a corner detector based on this principle, problems related to the rank of the structure tensor G will be avoided. It must be mentioned that the adopted feature detector in this thesis does not rely on the principle stated above. Further improvements of the algorithm to incorporate a better feature detector should therefore be chosen to facilitate feature tracking after initialization.

8.1.3 The Pyramidal Lucas-Kanade feature tracking algorithm

One major assumption of the LK method is that the image flow vector between two consecutive frames remains small. Only in that case, the displacements are relatively small compared to the pixel spacing and the differential eq. (8.10) still holds. For larger flow vectors, the LK method can only be used to refine coarse estimates which are determined using other methods. Extrapolation of the flow vectors computed from previous frames can for this purpose also be used. However, one of the most popular approaches was developed by Bouguet (2001) and uses the LK algorithm on a reduced-scale version of the image sequence. The method additionally allows to refine to smaller window sizes in case the accuracy of the traditional LK method is insufficient. A brief explanation of this feature tracking algorithm will be given hereafter, based on the work of Bouguet (2001) and Mahmoudi et al. (2014).

The goal of a local optical flow method is to track a sparse set of image points between two 2D grayscale images I and J. The quantities I(**x**) and J(**x**) represent the corresponding grayscale values (pixel intensities) on the location $\boldsymbol{x} = [x,y]^T$ in the image plane. It is assumed that the image point corresponding to a particular point of the 3D scene, $\boldsymbol{q_1} = [q_{1x},q_{1y}]^T$, in the first image I is known. The optical flow problem consists of subsequently locating its position $\boldsymbol{q_2} = \boldsymbol{q_1} + \boldsymbol{d} = [q_{1x} + d_x,q_{1y} + d_y]^T$ in the second image, based on the similarity between I($\boldsymbol{q_1}$) and J($\boldsymbol{q_2}$).

Similarly as stated before, the optical flow is assumed to be constant in a rectangular window of size $(2\,\omega_x + 1 \times 2\,\omega_y + 1)$. The image velocity (optical flow) $\mathbf{d}$ is then computed as the vector that minimizes the residual function ϵ. This function expresses the residual difference in pixel intensities between the rectangular window in image I and the shifted window in image J:

$$\epsilon(\boldsymbol{d}) = \epsilon(d_x,d_y) = \sum_{x=q_{1x}-\omega_x}^{q_{1x}+\omega_x} \sum_{y=q_{1y}-\omega_y}^{q_{1y}+\omega_y} (I(x,y) - J(x + d_x,y + d_y))^2 \tag{8.15}$$

The window in which the similarity function is measured is in this context also denoted as the integration window. The different steps in the iterative and pyramidal feature tracking algorithm are briefly explained below.

Step 1: Pyramid construction

A pyramid representation of an image I of size $n_x \times n_y$ is shown on Figure 8.3. Initially, the image has a high resolution (raw image). This high resolution image $I^0 = I$ is the ‘zero-level’ with image width and height $n_x^0 = n_x$ and $n_y^0 = n_y$. The pyramid of the image is then computed recursively, by computing level I^1 from I^0, I^2 from I^1, ..This results in a number of levels $L = 0,1,\ldots,m$ with corresponding images I^L.

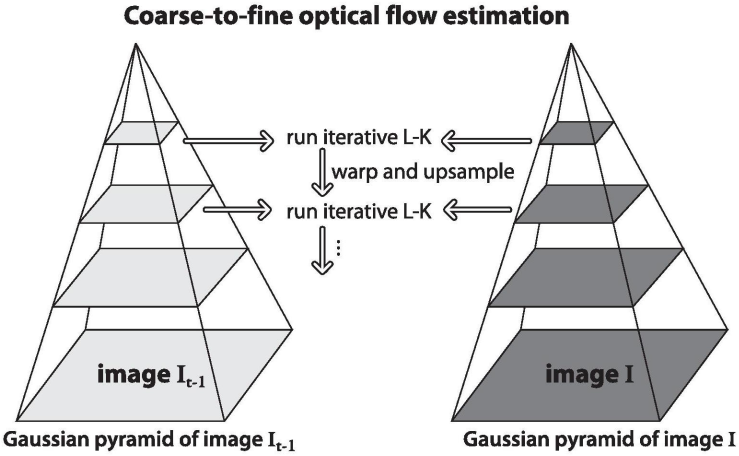


Figure 8.3: Pyramid representation of image I_{m-1} (denoted I in the derivation) and I_m (denoted J in the derivation) to compute optical flow. From Li et al. (2015).

For each level, image I^{L-1} is used to compute I^L. The image values are first copied for every two pixels in the image I^{L-1} (for $0 \leq x \leq n_x^{L-1} - 1$ and $0 \leq y \leq n_y^{L-1} - 1$):

$$\begin{aligned}
I^{L-1}(-1,y) &\doteq I^{L-1}(0,y),\\
I^{L-1}(x,-1) &\doteq I^{L-1}(x,0),\\
I^{L-1}(n_x^{L-1},y) &\doteq I^{L-1}(n_x^{L-1} - 1,y),\\
I^{L-1}(x,n_y^{L-1}) &\doteq I^{L-1}(x,n_y^{L-1} - 1),\\
I^{L-1}(n_x^{L-1},n_y^{L-1}) &\doteq I^{L-1}(n_x^{L-1} - 1,n_y^{L-1} - 1)
\end{aligned}$$

This allows to compute I^L as follows:

$$\begin{aligned}I^L(x,y) =&\frac{1}{4}I^{L-1}(2x,2y)+\\&\frac{1}{8}(I^{L-1}(2x-1,2y)+I^{L-1}(2x+1,2y)+I^{L-1}(2x,2y-1)+I^{L-1}(2x,2y+1))+\\&\frac{1}{16}(I^{L-1}(2x-1,2y-1)+I^{L-1}(2x+1,2y-1)+\\&I^{L-1}(2x-1,2y+1)+I^{L-1}(2x+1,2y+1))\end{aligned} \tag{8.16}$$

Equation 8.16 is only defined for values of x and y such that $0 \leq 2x \leq n_x^{L-1}-1$ and $0 \leq 2y \leq n_y^{L-1}-1$.

The size of the image level I^L is therefore given by the largest integers that satisfy the following equations:

$$n_x^L \leq \frac{n_x^{L-1}+1}{2} \tag{8.17}$$

$$n_y^L \leq \frac{n_y^{L-1}+1}{2} \tag{8.18}$$

Using equations 8.17 and 8.18, a pyramidal representation of the images can be constructed. For the two successive images in the image sequence I and J, this results in $\{I^L_{L=0,\ldots,L_m}\}$ and $\{J^L_{L=0,\ldots,L_m}\}$. The number of pyramids or pyramidal height L_m is usually picked heuristically, although typical values are 2, 3 or 4. Higher levels are usually not adopted because the resolution then becomes too small for typical image sizes. Detailed information can in that case not longer be discerned, which makes higher pyramid levels not advisable. Because the pyramidal representation is used to handle large pixel motions, L_m is usually chosen according to the maximum expected optical flow in the image.

The pyramidal tracking algorithm is based on computing the optical flow ($\boldsymbol{d}^L$) for every point for the deepest pyramid level L_m. This offers the advantage that the optical flow vector $\boldsymbol{d}^L$ can be kept very small while computing a large overall pixel displacement vector $\mathbf{d}$. The result of this computation can then be used as an initial guess for the pixel displacements at level L_m-1. Using this initial guess, the optical flow field for the feature points is subsequently refined at level L_m-1 and transferred to level L_m-2. This process is repeated until level 0 (the original image) is reached.

Step 2: Pixel matching over the different pyramid levels
To track a given point $\boldsymbol{q_1}$ in image I to its corresponding position $\boldsymbol{q_2} = \boldsymbol{q_1} + \boldsymbol{d}$ in image J, the pixel coordinates of $\boldsymbol{q_1}$ on the pyramidal images I^L, denoted as $\boldsymbol{q_1^L} = [q^L_{1x}, q^L_{1y}]$, need to be determined. Using equations 8.17 and 8.18, these can be computed as follows:

$$\boldsymbol{q_1^L} = \frac{\boldsymbol{q_1}}{2^L} \tag{8.19}$$

Step 3: Iterative optical flow computation
The core of the optical flow computation consists of finding at every level L in the pyramid the vector $\boldsymbol{d}^L$ which minimizes the matching error function ϵ^L. At every level, it is assumed that an initial guess of the optical flow at level L, denoted $\boldsymbol{g}^L = [g_x^L, g_y^L]^T$, is available from the computations done on level L+1. $\boldsymbol{g}^L$ is initialized at zero for the highest pyramid level L_m. To find the residual pixel displacement vector that minimizes ϵ^L, the standard iterative Lucas-Kanade operator is applied to obtain the optimal value of $\boldsymbol{d}^L$ by considering the expression of ϵ^L:

$$\epsilon^L(\boldsymbol{d}^L) = \epsilon^L(d_x^L,d_y^L) = \sum_{x=q_{1x}^L-\omega_x}^{q_{1x}^L+\omega_x} \sum_{y=q_{1y}^L-\omega_y}^{q_{1y}^L+\omega_y} (I^L(x,y) - J^L(x+g_x^L+d_x,y+g_y^L+d_y))^2 \tag{8.20}$$

The vector that minimizes this function can be found by setting the first derivative of ϵ w.r.t. $\boldsymbol{d}^L$ to zero:

$$\frac{\partial \epsilon}{\partial \boldsymbol{d}}|_{\boldsymbol{d}_{opt}^L} = [0,0]^T \tag{8.21}$$

In order to solve this equation, the spatial derivatives in the x- and y-direction in the $(2\,\omega_x + 1 \times 2\,\omega_y + 1)$ neighborhood of the point of interest are computed by using a central difference operator for the derivatives:

$$I_x(x,y) = \frac{I^L(x+1,y) - I^L(x-1,y)}{2} \tag{8.22}$$

$$I_y(x,y) = \frac{I^L(x,y+1) - I^L(x,y-1)}{2} \tag{8.23}$$

The gradient matrix G, centered around the analyzed point, is then computed at level L using these derivatives in the integration window:

$$G = \sum_{x=q_{1x}^L-\omega_x}^{q_{1x}^L+\omega_x} \sum_{y=q_{1y}^L-\omega_y}^{q_{1y}^L+\omega_y} \begin{bmatrix} I_x(x_i,y_i)^2 & I_x(x_i,y_i)I_y(x_i,y_i) \\ I_y(x_i,y_i)I_x(q_i) & I_y(x_i,y_i)^2 \end{bmatrix} \tag{8.24}$$

The computation of the temporal derivatives is performed using the image J (second image) with following expression:

$$I_t(x,y) = I^L(x,y) - J^L(x+g_x^L+d_x^L,y+g_y^L+d_y^L) \tag{8.25}$$

This allows to define the shift vector $\bar{b}$. This vector $\bar{b}$, also known as the image mismatch vector, captures the residual difference between the image patches $I^L(x,y)$ and $J^L(x+g_x^L+d_x^L,y+g_x^L+d_y^L)$ and is computed as:

$$\bar{b} = \sum_{x=q_{1x}^L-\omega_x}^{q_{1x}^L+\omega_x} \sum_{y=q_{1y}^L-\omega_y}^{q_{1y}^L+\omega_y} \begin{bmatrix} I_t I_x \\ I_t I_y \end{bmatrix} \tag{8.26}$$

It can be proven (Bouguet (2001)) that by substituting eq. (8.20) in eq. (8.21) and rearranging the terms, following expression is obtained:

$$\frac{1}{2}\frac{\partial \epsilon}{\partial \boldsymbol{d}} \approx \sum_{x=q_{1x}^L-\omega_x}^{q_{1x}^L+\omega_x} \sum_{y=q_{1y}^L-\omega_y}^{q_{1y}^L+\omega_y} \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix} \boldsymbol{d}^L - \begin{bmatrix} I_t I_x \\ I_t I_y \end{bmatrix} \tag{8.27}$$

which is similar to the classical optical flow equation. This expression can be written more shortly as:

$$\frac{1}{2}\frac{\partial \epsilon}{\partial \boldsymbol{d}} \approx G\boldsymbol{d}^L - \bar{b} \tag{8.28}$$

The optimal optical flow vector becomes therefore:

$$\boldsymbol{d}_{opt}^L = G^{-1}\bar{b} \tag{8.29}$$

Because in the derivation of this expression only a first order Taylor expansion is used, it is only valid for small pixel displacements. For this reason, it is necessary to iterate multiple times with the same scheme (in an Newton-Raphson fashion) to obtain a more accurate result. At the end of every iteration step, the current estimate of the optical flow vector $\boldsymbol{g}^L$ is updated with the new vector $\boldsymbol{d}^L$ as follows:

$$g_x^L = g_x^L + d_x^L \qquad g_y^L = g_y^L + d_y^L \tag{8.30}$$

The iterative loop is terminated in case the last iteration (maximum number of iterations) is reached or in case the measured correction is smaller than a predefined threshold.

Step 4: Propagation of result to next level and end of pyramid loop
In this step the final result of the iterative loop at level L is propagated to a lower level by adopting following expressions:

$$g_x^{L-1} = 2(g_x^L + d_x^L) \qquad g_y^{L-1} = 2(g_y^L + d_y^L) \tag{8.31}$$

This initial estimate of the optical flow at every point of interest is then further refined using the iterative approach explained in Step 3. The pyramid loop is terminated when the lowest level (original image) is reached. The final optical flow solution for every point of interest **d** is then computed based on the finest optical flow computation:

$$\boldsymbol{d} = \boldsymbol{g}^0 + \boldsymbol{d}^0 \tag{8.32}$$

which can also be written in following extended form:

$$\boldsymbol{d} = \sum_{L=0}^{L_m} 2^L \boldsymbol{d}^L \tag{8.33}$$

This finally result in a set of n vectors for the n tracked feature points (corners):

$$\Omega = \{\omega_1, \ldots, \omega_n | \omega_i = (x_i, y_i, v_i, \alpha_i)\} \tag{8.34}$$

where:

x_i, y_i	The x- and y-coordinates of feature i.
v_i	The velocity of feature i.
α_i	The motion direction of feature i.

8.2 Practical implementation of optical flow techniques

8.2.1 Detection of image points in the first processed frame

The goal of this thesis is to obtain a spatio-temporal description of a dynamically changing water surface. In case of self-induced sloshing, the water surface is initially disturbed by the incoming jet when the valves in the lock gates are opened. As a result, a (relatively) large translatory wave propagates in the tank. After a while, the water surfaces becomes more smooth and an oscillatory change in water level can then be observed. When the tank (i.e. lock chamber) is completely filled, the openings in the lock gates are closed and the oscillatory motion damps out. The final images of a measurement sequence therefore normally depict an almost flat and horizontal water surface.

As explained in the previous chapter, a quiet and flat water surface yields in much less refractive distortion of the regular feature pattern on plane F. This makes the detection of the individual features in the images much easier. We therefore start the temporal reconstruction of the water surface at the final image of the image sequence to start the temporal reconstruction in optimal conditions, i.e. without loss of feature points (see Section 8.2.3.).

The blobs that represent the feature points in the image are detected with OpenCV's *SimpleBlobDetector*, similarly as during the initialization step. The centers of gravity of the detected blobs are then sorted in the same way as explained in Chapter 6. This allows to relate the detected image points to the corresponding feature points on plane F. In case of large distortion, it is possible that the refractive displacement is so large that two neighboring feature points would appear in a reversed order in the image plane (denoted as 'separation of feature points' in Chapter 5). Because the final image is taken with a flat water surface, the problem of large distortion of the regular pattern is therefore avoided. The sorted pixel coordinates are finally stored in a list of pixel coordinates, which serve as basis for the rest of the feature detection process.

8.2.2 Detection of image points in an image sequence

The localization of the feature points in the following frames is done by applying the pyramidal LK optical flow method. OpenCV offers a direct implementation of this algorithm, named *CalcOpticalFlowPyrLK*. It uses the image positions in the previous frame as initial guess for the location of these points in the current frame. This makes the detection less computationally demanding compared to the first frame and allows to directly match the observed image points that correspond with the same feature point.

The function gives the predicted pixel coordinates of the shifted image points as result, together with their corresponding status (found or not found) and error. Two error metrics can be chosen to compute the quality of the new image position based on the calculated optical flow. If this error is larger than a predefined threshold, it is assumed that no match is possible (status: not found). The first error metric directly compares the pixel values of patches around the previous and current location of the image point. To this end, the L_1-distance between these patches is calculated:

$$L_1(\mathbf{x_1},\mathbf{x_2}) = \|\mathbf{x_1} - \mathbf{x_2}\|_1 = \sum_{d}^{n} |x_1(d) - x_2(d)| \tag{8.35}$$

By dividing this sum by the number of pixels in the window around the considered point, an error measure is obtained. However, Shakhnarovich (2005) states that this measure is sensitive for small deviations between two considered images. Geometric transformations (shifts and rotations) and changes in the imaging conditions (lighting or noise) make this error measure less advisable. We therefore chose to use the second possible error metric. This error measure computes the minimal eigenvalues of the spatial gradient matrix G corresponding to each individual tracked image point.

We finally obtain a new set of shifted image points, in which the order in which the feature points are listed is preserved. Every detected surface point **p** can in that case immediately be related to the corresponding feature point **f**. The geometric refractive-based problem is then completely defined an can be solved to find the optimal surface coefficients.

In case the optical flow calculation for one image point fails, two approaches are possible. Firstly, we can replace the coordinates of the feature point by NaN's to indicate that no match has been found. A second possibility is to use the best-possible match, for which the error is however larger than the threshold value. Because this estimated new position is unreliable and could negatively influence the optimization result, we finally decided not to use these inaccurate matches.

An additional advantage of using an optical flow technique to track the feature points is its general applicability. We chose to replace the feature points by perfect dots in the image, using their centers of gravity as new location. In further improvements of the algorithm, other feature detection methods might seem more appropriate. Template-matching (with e.g. a chessboard pattern) seems very promising because it uses the entire support region around the feature point to locate its position in the next image. A combination of the pyramidal LK tracker with such a pattern is in that case easy to implement.

8.2.3 Handling loss of feature points

Abrupt changes in the water surface shape (surface slope or water depth) can cause large refractive disparities of some feature points. In case a feature is not matched, it becomes impossible to further track it in the frames that are processed later. Knowledge about both the previous location and appearance of this feature point is in that case missing.

For this reason, we started with the end frame because in a practical application to self-induced sloshing, the most abrupt and largest water motions are expected in the beginning of the image sequence. Loss of feature points then only affects the limited amount of frames which are taken earlier than the frame considered, reducing the impact on the entire temporal reconstruction. Additionally, other possibilities can be used to solve or further mitigate loss of features:

- Using ambiguous matches for the rest of the tracking procedure.

  CalcOpticalFlowPyrLK automatically labels feature points for which the error measure is larger than the threshold as 'not found'. Although these are unreliable to use in the optimization of the current frame, these matches could be used to further track the feature points in the frames that follow.

- Interpolation and extrapolation of the optical flow field.

  The discrete optical flow field obtained with the matched feature points can be used to estimate the position of the points that were not located. The disadvantage is however that the chance of finding the feature points back in later frames remains small. The pyramid LK tracker uses their appearance in the previous frame to locate them in the following frame, which is in that case not known. To solve this issue, we could use their appearance in the last frame where they were found. This proved however difficult to implement using the current algorithms available in the OpenCV library.

- Using an image in which all feature points are found.

  A second possibility is to apply the LK algorithm but using a different frame as 'previous frame' for the remaining images. The image processed after loss of feature points is then compared with the last processed image for which all feature points were found.

  This approach is of course only possible in case the frame rate is sufficiently high. If not, the feature pattern will have a significant different appearance in case of rapid changes of the surface shape. Additionally, the risk that neighboring feature points are incorrectly labeled as 'match' increases. This approach is therefore more advised in case a different feature pattern would be adopted, in which the tracked feature points are more distinctive from each other.

- Use of multiple cameras.

  As was explained in Section 7, using two or more cameras makes the surface reconstruction more accurate and robust w.r.t. localization errors. It also offers the possibility to reconstruct larger areas of the water surface because the views of different cameras can be combined. With respect to feature tracking, multiple cameras offer an additional advantage.

Regions of the water surface where the surface points of one camera are lost can still be incorporated in the surface reconstruction by using the surface points of the other cameras. This yields in an accurate description of the local surface shape. The resulting surface function $\eta(x,y)$ can then be used to calculate where the lost surface points corresponding to the optimized surface coefficients should be located. The predicted positions of these surface points in the image plane can then be used to locate them in following frames. The previous appearance of these points is in that case however not known. Template-matching is therefore advised, as it offers the possibility to obtain a good prediction of the intensity-pattern that has to be found.

8.2.4 Integration in a global reconstruction algorithm

As mentioned previously, the time-dependent reconstruction of a dynamically changing water surface is discretized by considering each time frame separately. For every image of the image sequence, the image points **q'** are located which allow to obtain a set of surface coefficients that describe the instantaneous surface shape. A detailed elaboration of this optimization procedure was given in Chapter 6.

Optical flow is therefore only used to locate the image points in every individual image of the image sequence, in which knowledge about the previous location is adopted to facilitate feature detection. It allows to track the individual feature points across the different frames, making sorting of the detected features not longer required. A schematic overview of the global reconstruction procedure is given by Algorithm 3. As can be seen, this is in principle a combination of successive optimizations that were described in Chapter 6.

Algorithm 3: Spatio-temporal surface reconstruction

```
Input  : Image sequence of feature pattern seen through the water; calibrated camera
         system: distortion coefficients, camera center c, matrices K, R and T; initial
         feature locations f on plane F
Output: List of time-dependent set of surface coefficients

i → endFrame
for i → beginFrame do
    foreach camera do
        if (i=endFrame) then
            Detect feature points with SimpleBlobDetector
            Sort the feature points in systemetical order using Algorithm 1
            Undistort the obtained list of pixel coordinates
        else
            Replace blobs in the image by their center of gravity using SimpleBlobDetector
            Use the pyramidal LK tracker CalcOpticalFlowPyrLK to relate these points
            with feature points in previous frame
            if (feature f=found) then
                Undistort pixel coordinates
            else
                Replace feature point with NaN
            end
        end
        foreach f detected by the camera do
            Compute the error $E_f$ using Algorithm 2
        end
    end
    Determine best coefficients which minimize the total error $E_{tot}$
end
```

8.3 Testing the algorithm in a practical application to the filling of a navigation lock

The developed method is finally tested in a first experiment of a dynamically changing water surface. As was explained earlier in this chapter, a step-based approach is followed to obtain a temporal surface reconstruction. For every frame of an image sequence depicting the varying water surface, the determination of the instantaneous surface shape comprises an optimization procedure of the surface coefficients. The temporal resolution of the reconstruction is therefore dependent on the frame rate of the camera. The reconstructed surfaces are finally compared with what can theoretically be expected in order to validate the results.

8.3.1 Description of the test case

This temporal reconstruction is based on color images (green channel) of a single camera in which the entire feature pattern (50 mm×160 mm) could be captured. This required binning (explained in chapter 6) due to the limited bandwidth of the data cable. Using binning, we obtained a frame rate of approximately 6.8 fps (± 0.15 s between two successive frames).

In the experiment, the average surface height above the feature pattern on plane F was initially ± 38 mm. A certain part of the test tank was chosen as 'lock chamber', for which one side was bounded by a vertical wall. At the opposite side of the tank, a vertical valve was installed which could be opened in order to fill the created lock chamber that had a length of 1 m during this experiment. A water level difference between the lock chamber and the rest of the tank was then established and by opening the vertical valve separating both parts of the tank, a lock-filling process was simulated. This caused a translatory wave to enter the lock chamber, which finally damped out. The image sequence was continued until the steady oscillation of the water surface within the tank also finally disappeared after ± 66 seconds.

The surface reconstruction was then started at the final image of the image sequence. This proved to be highly efficient because the highly dynamic surface in the beginning of the filling process caused some points to be lost during tracking. In case we would have started the reconstruction at the first frame of the image sequence, this would have negatively influenced the entire temporal reconstruction of the filling process. It can be expected that the reconstructed surfaces in the beginning of the image sequence will therefore be less accurate and reliable.

8.3.2 Theoretical expectations

As will be shown in Section 8.3.3, this first attempt to reconstruct a time-varying water surfaces did not result in an accurate reconstruction over the entire water surface. As such, it proved difficult to compare our results with the self-induced sloshing theory explained in Chapter 2.

More basic hydrodynamic laws can however be applied to study the largest surface fluctuations within the tank, i.e. the propagation of a translatory wave. It is known that in case of a (small-amplitude) translatory wave travelling along a (frictionless) channel (with a rectangular cross-section), the wave celerity c_{wave} is given by:

$$c_{wave} = \sqrt{gh} \tag{8.36}$$

where g is the acceleration of gravity ($= 9.81 m/s^2$) and h is the local water depth.

In our experiment, the average water depth changed from ± 38 mm to ± 44 mm. In case we therefore take an approximate water level of 41 mm, this results in c_{wave}=0.63 m/s.

The period of the wave, i.e. the time needed to cross the entire length of the tank and return, equals in that case:

$$T = \frac{2L}{c_{wave}} = 3.2\,s \tag{8.37}$$

where the length of the tank L equals 1 m. The time the wave needs to cross the entire feature pattern with a length of 160 mm is then calculated similarly as:

$$t_{pattern} = \frac{0.160\,m}{c_{wave}} = 0.25\,s \tag{8.38}$$

8.3.3 Discussion of the results

Because it can be assumed that in this experiment the water remains mainly two-dimensional, the local water heights were averaged in the x-direction (width of the tank) to compare them with theoretical predictions. Although three-dimensional aspects of the filling process are in such a way neglected, this allows to mitigate some inaccuracies in the reconstruction.

As was already mentioned in Chapter 7, plane F on the bottom of the tank is not perfectly horizontal and slightly inclined in the x-direction. This is neglected in case we take an average surface height along the width of the tank, which would result in an inaccurate representation of reality. We therefore assume that the final reconstructed surfaces in the image sequence are almost perfectly flat and inclined in the opposite direction as plane F on the bottom of the tank. Based on this assumption, we used the final 30 images to compute ‘a reference surface’. In case the local surface height is expressed relatively to this flat and inclined plane, averaging along the width of the tank only neglects small 3D-aspects of the local surface shape.

Figure 8.4 shows the smoothed variation of the local water height at three particular locations in the tank. We adopted both metric M1 and metric M2, as defined in Chapter 7, as error measure in the optimization of the time-dependent surface coefficients. For every time frame, a surface model with 8 coefficients was optimized to describe the instantaneous surface shape. Every line corresponds with the surface level h_{ref} compared to our reference level, averaged over a strip along the width of the tank.

The incident translatory wave can easily be discerned at approximately 7 seconds after the start of the image sequence. Also the second passage of the wave is observed, approximately 3 seconds after the first passage as was derived in Section 8.3.2. The surface becomes subsequently more irregular and complex due to the multiple reflections of the wave, making a prediction of the three-dimensional shape of the water more difficult. The surface oscillations then damp out and the water surface finally stabilizes. The reconstructed surface in the final 15 seconds of the image sequence match closely with our reference level, which confirms that the water came to rest.

The large and unrealistic oscillations in the beginning of the reconstruction sequence (first 6-7 s) are most likely the result of loss of feature points during tracking across the image sequence. Because the features points were tracked starting from the end, the large surface fluctuations corresponding with the first passages of the translatory wave caused that multiple features were not further incorporated in the reconstruction. The amount of surface points **p** (which serve as input for our reconstruction) decreased significantly, making it more difficult to obtain a reliable description of the surface shape.

A somewhat difference result is also obtained for metric M1 and metric M2. In case the normal collinearity metric (M1) is used, the surface undulations for the three different sections have approximately the same magnitude at one specific time frame. The small lag between the different sections is however obvious because of the time required for the surface wave to travel the distance between these different longitudinal positions in the tank. Metric 1 seems therefore to give an accurate description of the temporal variability of the reconstructed surface area.

The reconstructed surfaces with M2 are however less reliable, as the magnitude for the different sections, considered in a small time window, are significantly different. This is especially noticed in the beginning of the filling process, when multiple features become lost. Because the changes in water level during passage of the wave should be approximately the same for every section in the lock chamber, we conclude that with metric M2 the inclination/ curvature of the surface is sometimes over-predicted.

In the previous section, we derived that the translatory wave needs approximately 0.26 s to cross the entire feature pattern on the bottom of the tank. Because the time between two consecutive frames in the image sequence is ± 0.15 s, the entire wave is in theory passed after two captured frames. This makes a smooth description of the passage of the wave impossible. Figure 8.5 shows however snapshots of the reconstructed surface corresponding with the second passage of the wave, averaged along the width of the tank. The length over which the surface is plotted is taken smaller than the length of the entire feature pattern. As was already mentioned in Chapter 7, extrapolation of the surface shape coefficients in areas for which they are not optimized is not advised. The difference between the results obtained with the collinearity metric (M1) and the disparity difference metric (M2) is small, although it becomes more pronounced in case of more inclined water surfaces. This is most likely the result of the camera position w.r.t. the instantaneous surface shape. For surfaces that are inclined in the negative y-direction (e.g. frame i), the viewing rays are more perpendicular to the surface which makes an accurate reconstruction difficult as was explained in Chapter 7.

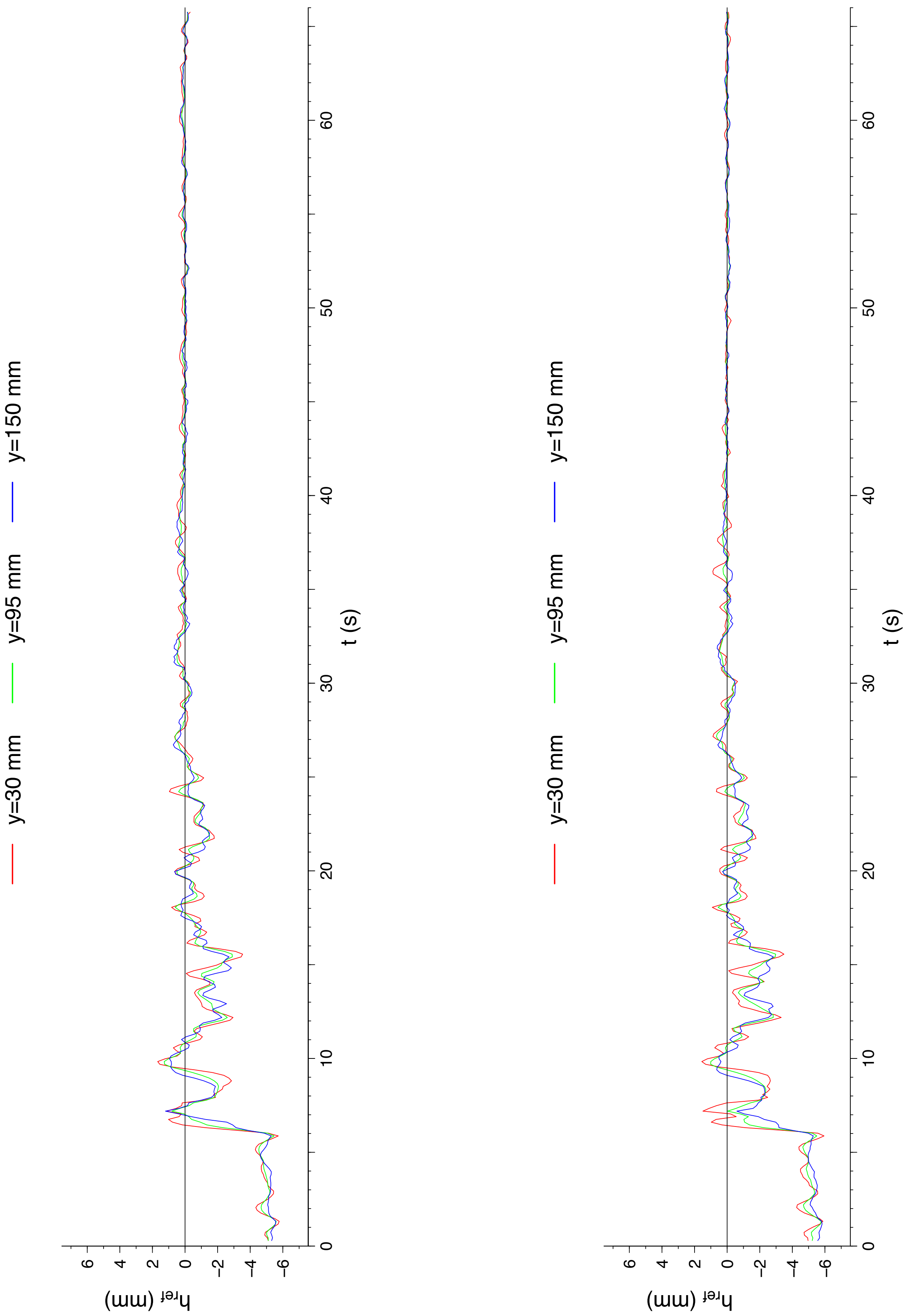
Before the wave crest enters the reconstructed area, the surface is inclined due to the spatial extent of the wave. The passage of the wave, propagating in the negative y-direction, can then be discerned by a smooth increase of the water level at the right side of the reconstructed section. Due to the fast propagation speed of the wave compared to the frame rate of the camera, the crest of the wave can only be observed in frame i+2. Next, the surface becomes strongly inclined in the positive y-direction, with more water stacked in the zone with small or negative y-coordinates. This corresponds with the observation that the translatory wave is reflected at the wall while the instantaneous surface shape at that moment does not change. Because the feature pattern was located close to the outer wall of the tested section, the constant inclination between frame i+3 and frame i+4 seems therefore correct. The wave then propagates in the opposite direction and the inclination of the surface becomes reversed. The wave crest is observed for a second time in frame i+5 and the water surface finally comes to rest as the wave propagates further in the lock chamber.

8.3.4 Critical remarks

In the previous section, we showed that the results obtained with our reconstruction algorithm allow a qualitative description of the filling process of a navigation lock.

On the other hand, loss of feature points and the low frame rate of the image sequence made it difficult to obtain high-accurate quantitative results. In Chapter 7, we showed that flat surfaces can be reconstructed with an accuracy in the range of 0.1 mm. The images that were used in this validation were on the contrary of much better quality than the images used for the temporal reconstruction in this chapter. The latter proved more difficult to post-process and we assume that feature detection was in this test case not sufficiently accurate. This explains the small fluctuations in the last part of the reconstructed surfaces which in theory should be approximately flat.

In Chapter 7, we concluded that the error made by using monochrome images remains small (valid for the range of water depths that was tested). We therefore suggest to do additional tests with dynamic water surfaces using monochrome images. The smaller image size (in bytes) allows in that case to obtain higher frame rates, which was the limiting factor in the test discussed above. No time was however available to do this during this thesis, so that the validation of dynamic surfaces stays limited to the aforementioned results.



(a) Metric M1: The normal collinearity metric.

(b) Metric M2: The disparity difference metric.

Figure 8.4: Temporal surface reconstruction of filling process.

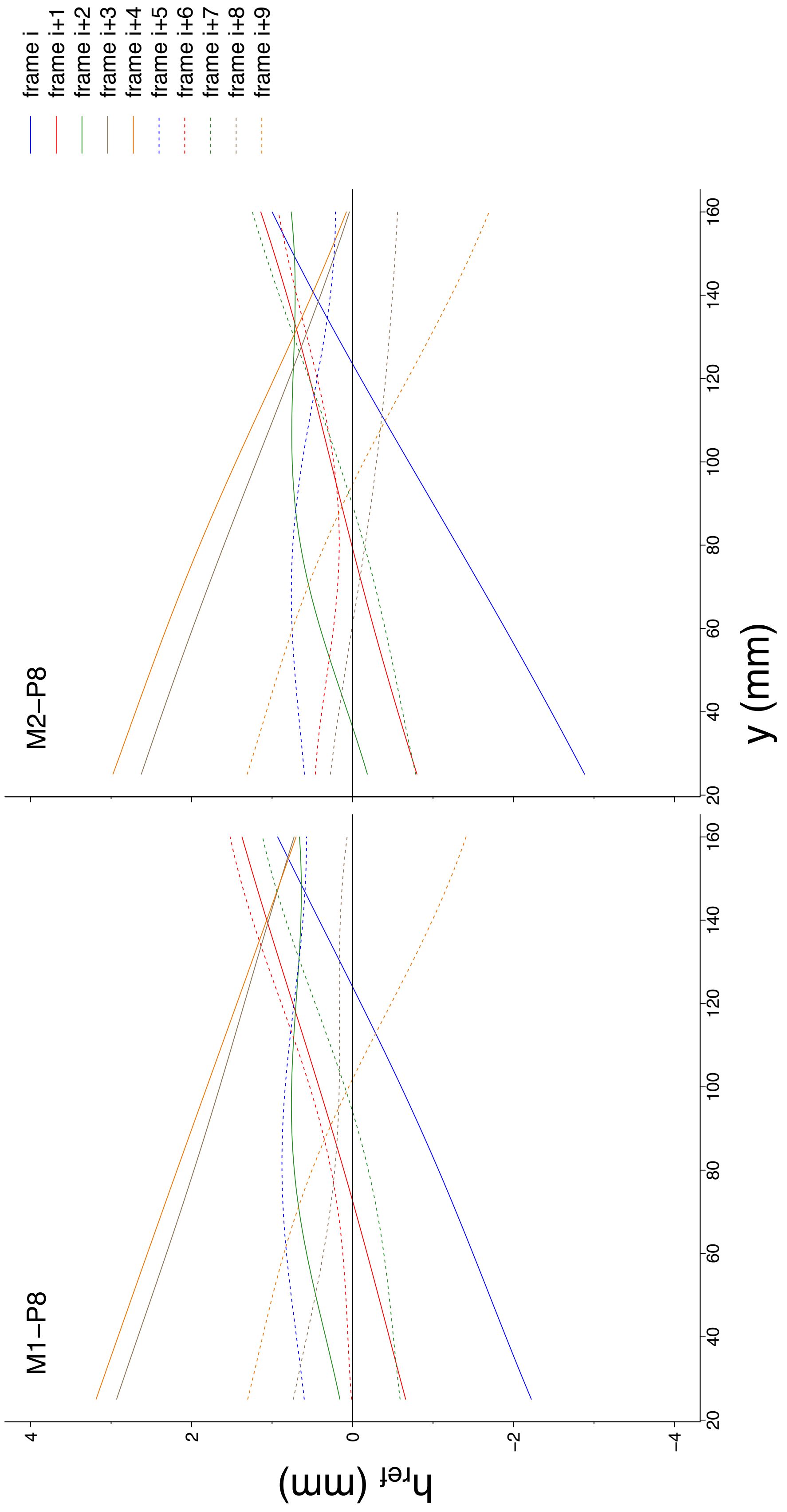


Figure 8.5: Snapshots of reconstructed surfaces during first passage of translatory wave.

Chapter 9

Conclusions and recommendations

The goal of this master thesis was to develop a methodology to obtain a spatio-temporal reconstruction of a dynamically changing water surface. High accuracy of the developed method was needed to study hydrodynamic phenomena in small-scale models. The study of self-induced sloshing was with this respect considered as a first application area. An overview of our accomplishments is given hereafter. Due to the limited amount of time available during this thesis, some aspects that could improve the performance of the algorithm could not be implemented. Hence, an overview of possible improvements is given in Section 9.4 that can be made in further development of the algorithm.

9.1 Development of the methodology

During this thesis, a novel methodology has been developed to obtain a spatio-temporal reconstruction of a dynamically changing water surface. The method comprises a combination of several approaches that have been used to reconstruct the shape of refractive, transparent and specular reflective objects. This work was mainly based on the refractive-stereo approach that was suggested by Morris (2004). The technique presented in this thesis is based on the same principle in which images of a refracted feature pattern seen through the water are used to derive a geometric problem in function of two unknowns: the local water depth and the surface normal.

However, the presented methodology combines a shape from refractive distortion approach with a low parameter surface model. As such, the reconstruction of the entire water surface could be reduced to a multivariate optimization of a limited amount of unknown surface shape coefficients. To that end, two error metrics were proposed, both expressing the dissimilarity between the two alternative surface descriptions. The resulting set of optimized coefficients could finally be used to obtain a three-dimensional description of the local surface height.

This novel approach makes the algorithm more robust to incorrectly detected feature points, reduces the computation time and allows a much larger reconstructed area for the same amount of adopted cameras. The developed methodology was implemented in a single C++ program, which makes it easy to apply in other hydraulic research. The entire program is also modular based, which allows to adapt some aspects of the reconstruction to the phenomenon that is reconstructed in order to obtain more robust and accurate results.

9.2 Validation of the methodology

The method described in this thesis was validated using both experimental tests and numerical simulations. These allowed to verify its accuracy and study the influence of the most crucial parameters in the reconstruction w.r.t. robustness and accuracy of the algorithm.

The experimental tests showed that flat water surfaces can be reconstructed with an accuracy that is assumed to be higher than what can be obtained with a level gauge with vernier scale, i.e. 0.2 mm on water level differences. In case sufficient care is taken to obtain strong and uniform illumination of the feature plane, the image format (mono or color) and the type of lighting seemed to have a small effect on the reconstruction accuracy.

The numerical simulations were conducted to quantify the influence of different parameters within the methodology on the severity of incorrect feature localization. The sensitivity w.r.t. feature localization seemed larger for surfaces located further and more perpendicular to the viewing rays. The latter was also confirmed by experimental tests in which a comparison between the reconstruction results with two different camera positions was made.

Both the experimental tests and numerical simulations additionally showed that the collinearity metric outperformed the disparity difference metric in case accurate feature detection is obtained. This difference seemed however only significant for more extensive models, for which in that case more irregular surfaces are fitted through the incorrectly detected feature points. The simulations showed on the other hand that in case the localization errors are increased, the disparity difference metric proved more robust. Finally, it was shown that a combination of multiple cameras significantly increased the performance of the validated method. Errors related to a single camera could in this way be mitigated by adopting the information from other cameras, increasing the accuracy and robustness of the reconstruction.

Based on these results, it was possible to discern the best configuration settings for the reconstruction algorithm, i.e. allowing the most accurate and robust results. These conclusions and remarks can be used by researchers that adopt the developed methodology, so that the performance of the algorithm can be maximized.

9.3 Application of the methodology

In order to obtain a temporal description of the water surface, optical flow was incorporated, i.e. a pyramidal implementation of the Lucas-Kanade feature tracker (Bouguet (2001)), to track the feature points over multiple images. Because the presented reconstruction methodology considers each time frame separately, this only required minor adjustments to the developed algorithm. The extended algorithm was tested in a first experiment comprising the filling process of a rectangular tank. A significant difference between the two metrics was noticed, in which the normal collinearity metric seemed more physically consistent. As the surface elevations at different sections in the test tank showed roughly the same variation in time, i.e. the same magnitude but a small time lag in between, it can be assumed that the metric is able to obtain an approximate description of the water surfaces. The difference in the undulations at various sections in the tank became much larger with the disparity difference metric, indicating a poorer performance of the algorithm.

The available measurement setup seemed however to be a limiting factor and made a high temporal resolution of the reconstruction difficult. Additionally, the chosen feature pattern proved difficult to track which resulted in loss of a substantial amount of features over the entire image sequence. Compared to the validation on still water, this made the accuracy of the optimized surfaces not nearly as good. The obtained results could therefore only be used to qualitatively describe the filling process.

The motivation behind this master thesis was the fundamental experimental research on self-induced sloshing during navigation lock filling. An extensive study of this phenomenon was however not possible in the limited time span of this thesis. However, it can be assumed that the developed algorithm can be used as measurement technique for further studies on this topic. Especially if combined with velocity measurement techniques, such as PIV or PTV (see Section 9.4), the developed methodology seems a reliable and accurate research tool in experimental research on self-induced sloshing.

The technique was initially developed to study the self-induced sloshing phenomenon in a scale model of a navigation lock. The developed algorithm can however easily be adjusted to study other hydrodynamic phenomena in case the adopted surface model is adjusted accordingly. Moreover, the application area of the methodology is not only limited to water. Other refractive and transparent objects, fluids or solid matter, could also be reconstructed in case a theoretical prediction of the surface shape is available. An appropriate value for the refractive index is in that case obviously required.

9.4 Recommendations for further improvement of the methodology

The performance of the current algorithm can still be improved by adjusting several steps in the reconstruction process. In that regard, improving each aspect of the reconstruction separately is possible because the implementation of the methodology in C++ is modular based. Each adaptation to the algorithm can therefore be implemented without affecting the rest of the program. An extensive overview is given hereafter, intended to serve as basis for further development of the reconstruction technique. The order of these suggestions is chosen according to the presumed additional value to the algorithms performance, in which the most crucial improvements are listed first.

- **Choice of a more robust and accurate feature detector.**
  The implemented feature detector presented in Chapter 5 consists of locating small dots (i.e. the tracked feature points) within the image plane. Although this allowed to reconstruct still water with high accuracy, it proved difficult to track these points across an image sequence depicting fluctuating water. Due to their identical appearance, loss of some feature points was hard to avoid. This can be solved by incorporating template-based corner detectors, as discussed in Chapter 5. The fuzzy corner detector presented by Cuevas et al. (2011) seems for this purpose one of the most promising alternatives. Chessboard patterns or similar templates are in that case advised to facilitate robust feature detection because they are locally specific as explained in Chapter 5.

  Another technique, which was not discussed in Chapter 5 due to its complexity, is presented in Kim et al. (2012). They proposed a hybrid feature detector in which intensity-based (template) corner detection is combined with a contour-based method. This proved to even further enhance the stability and accuracy of the feature detector. More specifically for the presented reconstruction algorithm in this thesis, it also allows to use the additional information derived from distortion of lines instead of movements of discrete feature points. Although it therefore might improve the reconstruction algorithm, this method seems more complex to implement in practice. Based on the resources available, a compromise will therefore have to be made between performance and feasibility of implementation.

- **Incorporating PIV measurements.**
  In order to study hydrodynamic phenomena more in detail, flow velocities are usually required. For this purpose, the techniques of PIV (Particle Image Velocimetry) and PTV (Particle Tracking Velocimetry) are often used in which particles are tracked across an image sequence. The motion of the seeding particles is then computed and used as an approximate estimate of the 2D fluid velocity.

  Combining the image-based surface height reconstruction with one of these two techniques would allow to obtain an even further insight in the flow patterns that are studied. Additionally, this combination could be used to obtain 3D velocities of the seeding particles. In case one first reconstructs the surface height in the zone that is studied, the presented refractive approach could be used to determine the 3D position of the seeding particles in the water mass. Three-dimensional PIV or PTV would allow to study even more complex flow patterns and to gain even more insight in the studied phenomena.

  A combination of both methods could make it on the other hand more difficult to track feature points and/or the seeding particles due to the complex scene that is depicted in the image (particles and features randomly distributed between each other). For this reason, using different color channels for the PTV/PIV and the tracked feature pattern seems the best suitable approach. UV light can then be used to illuminate the water in the tested section in combination with a fluorescent feature plane and fluorescent seeding particles. By choosing a different color for the seeding particles and feature plane, the mutual interference is minimized.

- **Improvement of optimization algorithm.**
  In this work, the Levenberg-Marquardt algorithm (LMA) was adopted to obtain a good estimation of the best-fitting surface shape coefficients. With regard to optimization, two possibilities are suggested that could improve the robustness and accuracy of the optimization result:

  - investigation of the influence of a different step-size or stopping condition.
    As mentioned in Chapter 6, this thesis used a step-based stopping condition with a step-size for each coefficient of 0.01 mm. This choice was based on of a brief comparison of the obtained results using several alternatives. More extensive research could be done on the influence of the chosen optimization configuration, which might show that another choice is more advisable.

  - implementation of another optimization algorithm.
    Although LMA is considered as a good choice for multivariate (nonlinear) optimization, other optimizations algorithm could further improve the final obtained result.

- **Improvement of feature tracking.**
  Once detected, tracking of the feature points was done by the pyramidal LK tracker. Because this algorithm is readily available in the OpenCV library, this allowed to incorporate optical flow in the limited time available during this thesis. Practical tests with this algorithm showed however that it sometimes does not find matches for feature points that have significantly moved or changed appearance between successive frames. Hence, two possible solutions to solve this issue are presented.

  A first alternative is to use a higher frame rate and feature points that are more distinguishable from each other. Secondly, another optical flow technique could be implemented. For this purpose, we directly refer to the Computer Vision Group of the University of Freiburg[1]. They have developed several improved optical flow techniques of which the

---

[1] University of Freiburg, Dep. of Computer Science. Source code available at: Pattern Recognition and Image Processing - Binaries/Code,`http://lmb.informatik.uni-freiburg.de/resources/software.php`

source codes are readily available online. With only small modifications to the current algorithm, implementation of one of these methods should definitely be feasible.

- **Incorporation of the separate color channels.**
  The image-based technique in this thesis was combined with two different color formats: monochrome (grayscale) images and color images in the Bayer GB format. The latter could be split in their three different color channels and the green channel was subsequently used in the reconstruction algorithm. For monochrome images on the other hand, only a single representative value for the refractive index r_w could be used. Experiments showed that this simplification does not have a large influence on the obtained accuracy.

  Sirisha and Sandhya (2013) however presented a corner detection algorithm in which they detect corners with the Harris corner detector in both grayscale (using the intensity level) and color (using the RGB values) formats of the same image. The corners were subsequently analyzed using cross correlation and a RANSAC-scheme, which significantly improved the performance of the feature detection method.

  The same principle could be applied in the developed method, in which image points are detected in each color channel separately and an adjusted refractive index (according to the color channel) is used in Algorithm 2. This would triple the amount of input parameters for the optimization and even further improve the robustness w.r.t. feature localization errors. Processing of all three color channels also facilitates the combination of PIV/PTV with the surface height reconstruction as was mentioned earlier.

- **Detection and removal of outliers.**
  In some cases, it is possible that optical flow methods find an incorrect match for some of the feature points (due to e.g. noise in the image, large displacements of the feature points, ...). These incorrect matches should be removed from the list of refractive disparities because they harm the global optimization result. To that end, the error measure computed by the OpenCV function *CalcOpticalFlowPyrLK* (explained in Chapter 6) could be used.

  A second possibility is to evaluate the consistency of the detected image points between the three color channels. Because the noise level often depends on the type of sensor detector and wavelength of the incident light, the influence of noise in one particular channel can be mitigated by adopting the detected feature points in the other two channels.

- **Combining multiple initializations.**
  In the initialization step of the algorithm, the extrinsic camera parameters are derived based on an image of the feature plane without water. As was explained in Section 7.1.3, the estimated matrix R and vector T suffer from the inherent error related to multivariate optimization algorithms. For this reason, combining multiple initializations could make the estimation much more accurate. Taking initialization images at multiple, known, still water levels would allow to check if the estimated camera parameters are correct. If not, these could be adjusted in order to fit the observed refractive disparities corresponding to the known water level changes.

- **Selecting the best camera for every part of the surface area.**
  Chapter 7 showed that there exists an optimal position and viewing angle of the camera w.r.t. the reconstructed surface for which the performance of the algorithm is ideal. As such, the use of multiple cameras allows to reconstruct each zone of the water surface with the camera for which the viewing rays intersect the surface in the most favorable conditions. Because this is however dependent on the instantaneous surface shape of the oscillating water mass, it proved difficult to implement this in a experimental application. Further research on the practical implementation is therefore required.

Although an extensive validation and error assessment was conducted in Chapter 7, several tests are also advised to gain further insight in the influence of several parameters in the reconstruction:

1. **Experimental tests.**
   In the first test case on a dynamically changing water surface described in Chapter 8, a quantitative validation seemed difficult due to the limitations of the measurement setup. It is therefore suggested to use a more advanced imaging setup to test the developed algorithm on additional image sequences. Critical aspects on which should be focused are a uniform and strong illumination of the feature pattern and a higher possible bandwidth. These seemed in this work the two most limiting factors to obtain a high frame rate and high quality of the obtained images. Because the tests showed that the error made by using monochrome images remains small, the reduced size of grayscale images could help to achieve a higher temporal resolution.
2. **Numerical tests.**
   In Chapter 7, several numerical simulations were conducted. Some aspects of the reconstruction could however not be verified due to the limited time available. For those particular parameters in the methodology, suggestions were already given in Chapter 7 on how these tests could be conducted. These can be used to gain even further insight in the critical components w.r.t. accuracy and robustness of the developed method.

Bibliography

G. Abramovich and L. Schindel. *General properties of turbulent jets.* MIT press, 1963.

Y. Adato, Y. Vasilyev, O. Ben-Shahar, and T. Zickler. Toward a theory of shape from specular flow. In *Proc. IEEE 11th Int. Conf. Computer Vision (ICCV), 2007*, pages 1–8, 2007.

S. Agarwal, S. P. Mallick, D. Kriegman, and S. Belongie. On refractive optical flow. In *Proc. European Conf. Computer Vision (ECCV), 2004*, pages 483–494. Springer, 2004.

ALGLIB. Project: ALGLIB: a cross-platform open source numerical analysis and data processing library. Available at: `http://www.alglib.net/`, 2016.

K. Amano and R. Iwano. Experiments and analysis of jet-induced sloshing in a tank. *Trans. the Japan Society of Mechanical Engineers, Series C*, 57:1947–1954, 1991.

Anonymous. Lecture 21: Two-dimensional wave equations, double Fourier Series. University Lecture, given at Department of Mathematics, Southeast University (SEU), Nanjing.

B. Atcheson, I. Ihrke, W. Heidrich, A. Tevs, D. Bradley, M. Magnor, and H.-P. Seidel. Time-resolved 3d capture of non-stationary gas flows. *ACM Trans. Graphics*, 27(5):132:1–132:9, 2008.

M. Baba, K. Ohtani, M. Imai, and T. Konishi. New laser rangefinder for three-dimensional shape measurement of specular objects. *Optical Engineering*, 40(1):53–60, 2001.

M. Baba, D. Narita, and K. Ohtani. An advanced rangefinder equipped with a new image sensor with the ability to detect the incident angle of a light stripe. *Journal of optics A: Pure and applied optics*, 6(1):10–16, 2004.

M. Banerjee and M. K. Kundu. Handling of impreciseness in gray level corner detection using fuzzy set theoretic approach. *Applied Soft Computing*, 8(4):1680–1691, 2008.

S. S. Beauchemin and J. L. Barron. The computation of optical flow. *ACM Computing Surveys*, 27(3):433–466, 1995.

M. Ben-Ezra and S. K. Nayar. What does motion reveal about transparency? In *Proc. IEEE 9th Int. Conf. Computer Vision (ICCV), 2003*, pages 1025–1032, 2003.

D. N. Bhat and S. K. Nayar. Stereo in the presence of specular reflection. In *Proc. IEEE 5th Int. Conf. Computer Vision (ICCV), 1995*, pages 1086–1092, 1995.

S. Birchfield and C. Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, 20(4):401–406, 1998.

M. J. Black and P. Anandan. A framework for the robust estimation of optical flow. In *Proc. IEEE 4th Int. Conf. Computer Vision (ICCV), 1993*, pages 231–236, 1993.

A. Blake and G. Brelstaff. Geometry from specularities. In *Proc. IEEE 2nd Int. Conf. Computer Vision (ICCV)*, pages 394–403, 1988.

W. Blake. Mechanics of flow-induced sound and vibration: Complex flow-structure interactions. *Applied Mathematics and Mechanics Series*, 17:130, 1986.

T. Bonfort and P. Sturm. Voxel carving for specular surfaces. In *Proc. IEEE 9th Int. Conf. Computer Vision (ICCV), 2003*, pages 591–596, 2003.

T. Bonfort, P. Sturm, and P. Gargallo. General specular surface triangulation. In *Proc. Asian Conf. Computer Vision-part 2, 2006*, pages 872–881, Hyderabad,India, 2006. Springer.

J. Bouguet. Camera calibration toolbox for matlab. `http://www.vision.caltech.edu/bouguetj/calib_doc/`, 2014. [Online; Accessed: 2015-11-10; Last Update: 2015-10-14].

J.-Y. Bouguet. Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. pages 1–10, 2001.

G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. URL `http://opencv.org/`.

G. Bradski and A. Kaehler. *Learning OpenCV: Computer vision with the OpenCV library.* O'Reilly Media, Inc., 2008.

D. C. Brown. Decentering distortion of lenses. *Photogrammetric Engineering*, 32(3):444–462, 1966.

D. Chen and G. Zhang. A new sub-pixel detector for x-corners in camera calibration targets. In *Proc. 13th Int. Conf. Computer Graphics, Visualization and Computer Vision, 2005*, 2005.

T. Chen, M. Goesele, and H.-P. Seidel. Mesostructure from specularity. In *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition (CVPR), 2006*, volume 2, pages 1825–1832, 2006.

T. Chen, H. Lensch, C. Fuchs, and H.-P. Seidel. Polarization and phase-shifting for 3d scanning of translucent objects. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2007*, pages 1–8, 2007.

T. Chen, H.-P. Seidel, and H. Lensch. Modulated phase-shifting for 3d scanning. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2008*, pages 1–8, 2008.

Y.-Y. Chuang, D. E. Zongker, J. Hindorff, B. Curless, D. H. Salesin, and R. Szeliski. Environment matting extensions: Towards higher accuracy and real-time capture. In *Proc. 27th Conf. Computer Graphics and Interactive Techniques*, pages 121–130. ACM Press/Addison-Wesley Publishing Co., 2000.

I. J. Cox, S. L. Hingorani, S. B. Rao, and B. M. Maggs. A maximum likelihood stereo algorithm. *Computer Vision and Image Understanding*, 63(3):542–567, 1996.

E. Cuevas, D. Zaldivar, M. Pérez-Cisneros, E. Sánchez, and M. Ramírez-Ortegón. Robust fuzzy corner detector. *Intelligent Automation and Soft Computing*, 17(4):415–429, 2011.

J. M. Daida, D. Lund, C. Wolf, G. A. Meadows, K. Schroeder, J. F. Vesecky, D. R. Lyzenga, B. C. Hannan, and R. R. Bertram. Measuring topography of small-scale water surface waves. In *Proc. Int. Geoscience and Remote sensing Symposium*, volume 3, pages 1881–1881, 1995.

M. W. Davidson and The Florida State University. Specular and diffuse reflection. `http://micro.magnet.fsu.edu/primer/java/reflection/specular/`, 2013. [Online; Accessed: 2016-05-10; Last Update: 2015-11-13].

J. E. Davis, R. Yang, and L. Wang. Brdf invariant stereo using light transport constancy. In *Proc.IEEE 10th Int. Conf. Computer Vision (ICCV), 2005*, volume 1, pages 436–443, 2005.

J. S. De Bonet and P. Viola. Roxels: Responsibility weighted 3d volume reconstruction. In *Proc. IEEE 7th Int. Conf. Computer Vision (ICCV), 1999*, volume 1, pages 418–425, 1999.

D. F. Dementhon and L. S. Davis. Model-based object pose in 25 lines of code. *Int. Journal of Computer Vision (IJCV)*, 15(1):123–141, 1995. ISSN 1573-1405.

G. Eren, O. Aubreton, F. Meriaudeau, L. S. Secades, D. Fofi, A. T. Naskali, F. Truchetet, and A. Ercil. Scanning from heating: 3d shape estimation of transparent objects from local surface heating. *Optics Express*, 17(14):11457–11468, 2009.

M. A. Föstner and E. Gülch. A Fast Operator for Detection and Precise Location of Distinct Points, Corners and Centers of Circular Features. In *ISPRS Intercommission Workshop*, Interlaken, Switzerland, 1987.

Y. Francken, T. Cuypers, and P. Bekaert. Mesostructure from specularity using gradient illumination. In *Proc. 5th ACM/IEEE Int. Workshop on Projector camera systems*, page 11. ACM, 2008a.

Y. Francken, T. Cuypers, T. Mertens, J. Gielis, and P. Bekaert. High quality mesostructure acquisition using specularities. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2008*, pages 1–7, 2008b.

Y. Francken, C. Hermans, T. Cuypers, and P. Bekaert. Fast normal map acquisition using an lcd screen emitting gradient patterns. In *Proc. Canadian Conf. Computer and Robot Vision, 2008*, pages 189–195, 2008c.

M. Fukaya, M. Baba, K. Okamoto, and H. Madarame. Self-induced sloshing caused by a jet from a rectangular tank wall. In *Proc. HYDRA-2000, 26th Congr. IAHR,*, volume 1, pages 468–473, 1995.

X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, 25(8):930–943, 2003.

Gilbert. OpenCV2 marathon nineteenth times – Harris corner detection . `http://www.programering.com/a/MjM4gDNwATg.html`, 2014. [Online; Accessed: 2016-03-16; Last Update: 2014-07-17].

B. Goldlucke, I. Ihrke, C. Linz, and M. Magnor. Weighted minimal hypersurface reconstruction. *IEEE Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, 29(7):1194–1208, 2007.

D. B. Goldman, B. Curless, A. Hertzmann, and S. M. Seitz. Shape and spatially-varying brdfs from photometric stereo. *IEEE Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, 32(6):1060–1071, 2010.

C. Harris and M. Stephens. A combined corner and edge detector. In *Proc. 4th Alvey Vision Conf.*, volume 15, page 50, 1988.

R. Hartley and A. Zisserman. *Multiple view geometry in computer vision.* Cambridge University Press, 2003.

A. H. Harvey, J. S. Gallagher, and J. L. Sengers. Revised formulation for the refractive index of water and steam as a function of wavelength, temperature and density. *Journal of Physical and Chemical Reference Data*, 27(4):761–774, 1998.

S. Hata, Y. Saitoh, S. Kumamura, and K. Kaida. Shape extraction of transparent object using genetic algorithm. In *Proc. IEEE 13th Int. Conf. Pattern Recognition, 1996*, volume 4, pages 684–688, 1996.

G. Healey and T. O. Binford. Local shape from specularity. *Computer Vision, Graphics, and Image Processing*, 42(1):62–86, 1988.

J. A. Hesch and S. I. Roumeliotis. A direct least-squares (dls) method for pnp. In *Proc. IEEE Int. Conf. Computer Vision (ICCV), 2011*, pages 383–390, 2011.

V. Hilsenstein. Surface reconstruction of water waves using thermographic stereo imaging. In *Image and Vision Computing New Zealand*, volume 2. Citeseer, 2005.

B. K. Horn. *Shape from Shading: A Method for Obtaining the Shape of a Smooth Opaque Object From One View.* PhD thesis, Cambridge, MA, USA, 1970.

B. K. Horn and B. G. Schunck. Determining optical flow. In *1981 Technical symposium east*, pages 319–331. Int. Society for Optics and Photonics, 1981.

H. Hu, T. Kobayashi, T. Saga, T. Segawa, N. Taniguchi, M. Nagoshi, and K. Okamoto. A piv study on the self-induced sloshing in a tank with circulating flow. In *Proc. 2nd Pacific Symposium on Flow Visualization and Image Processing*, 1999.

Y. Huang and X. Zhuang. Motion-partitioned adaptive block matching for video compression. In *Proc. IEEE Int. Conf. Image Processing*, volume 1, pages 554–557, 1995.

I. Ihrke, B. Goidluecke, and M. Magnor. Reconstructing the geometry of flowing water. In *Proc. IEEE 10th Int. Conf. Computer Vision (ICCV), 2005*, volume 2, pages 1055–1060, 2005.

I. Ihrke, K. N. Kutulakos, H. Lensch, M. Magnor, and W. Heidrich. Transparent and specular object reconstruction. *Computer Graphics Forum*, 29(8):2400–2426, 2010.

Itseez. *The OpenCV Reference Manual: edition 3.1.0*, 2015. [Online; Accessed: 2016-5-18; Last Update: 2015-12-18].

H. Jin, S. Soatto, and A. J. Yezzi. Multi-view stereo beyond lambert. In *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition (CVPR), 2003*, volume 1, pages 171–178, 2003.

S. M. M. Kahaki, M. J. Nordin, and A. H. Ashtari. Contour-based corner detection and classification by using mean projection transform. *Sensors*, 14(3):4126–4143, 2014.

A. C. Kak and M. Slaney. *Principles of Computerized Tomographic Imaging.* IEEE press, 1988.

W. C. Keller and B. L. Gotwols. Two-dimensional optical measurement of wave slope. *Applied Optics*, 22(22):3476–3478, 1983.

S. Kim, I. S. Kweon, and W. H. Lee. Orientation based multi-scale corner detection for mobile robot application. In *Proc. IEEE 12th Int. Conf. Control, Automation and Systems (ICCAS), 2012*, pages 466–468, 2012.

P. Kolkman. Low-head navigation locks door filling and emptying systems developed by hydraulic investigations. *Land+water International*, 16:7, May 1973. Also published as Publication no. 111, Delft Hydraulics, Sep. 1973.

S. Krig. *Computer Vision Metrics: Survey, Taxonomy, and Analysis.* Apress, 2014.

V. Kumar and A. Nagappan. Study and comparison of various point based feature extraction methods in palmprint authentication system. *International Journal Of Computational Engineering Research*, 2:82–89, 2012.

K. N. Kutulakos and E. Steger. A theory of refractive and specular 3d shape by light-path triangulation. *Int. Journal of Computer Vision (IJCV)*, 76(1):13–29, 2008.

S. Kyle. Dissecting the camera matrix, part 3: The intrinsic matrix. `http://ksimek.github.io/2013/08/13/intrinsic/`, 2013. [Online; Accessed: 2015-10-30; Last Update: 2013-08-13].

H. Lamb. *Hydrodynamics.* Cambridge University Press, 1932.

V. Lepetit, F. Moreno-Noguer, and P. Fua. Epnp: An accurate o (n) solution to the pnp problem. *Int. Journal of omputer vision*, 81(2):155–166, 2009.

C. Li, M. Shaw, D. Pickup, D. Cosker, P. Willis, and P. Hall. Realtime video based water surface approximation. In *Proc. Conf. for Visual Media Production (CVMP), 2011*, pages 109–117, 2011.

D. Li, Q. Li, L. Tang, S. Yang, N. Cheng, and J. Song. Invariant Observer-Based State Estimation for Micro-Aerial Vehicles in GPS-Denied Indoor Environments Using an RGB-D Camera and MEMS Inertial Sensors. *Micromachines*, 6(4):487–522, 2015. ISSN 2072-666X.

Z.-m. Liang, H. Gao, Z. Wang, and L. WU. Sub-pixels corner detection for camera calibration. *Transaction of China Welding Institution*, 27(2):102, 2006.

T. Lindeberg. Feature detection with automatic scale selection. *Int. Journal of Computer Vision (IJCV)*, 30(2):79–116, 1998.

B. D. Lucas, T. Kanade, et al. An iterative image registration technique with an application to stereo vision. In *IJCAI*, volume 81, pages 674–679, 1981.

L. Lucchese and S. K. Mitra. Using saddle points for subpixel feature detection in camera calibration targets. In *Circuits and Systems*, volume 2, pages 191–195, 2002.

W.-C. Ma, T. Hawkins, P. Peers, C.-F. Chabert, M. Weiss, and P. Debevec. Rapid acquisition of specular and diffuse normal maps from polarized spherical gradient illumination. In *Proc. 18th Eurographics Conf. Rendering Techniques*, pages 183–194. Eurographics Association, 2007.

S. Mahmoudi, M. Kierzynka, P. Manneback, and K. Kurowski. Real-time motion tracking using optical flow on multiple gpus. *Bulletin of the Polish Academy of Sciences: Technical Sciences*, 62(1):139–150, 2014.

S. McHugh. Digital camera sensors. `http://www.cambridgeincolour.com/tutorials/camera-sensors.htm`, 2016. [Online; Accessed: 2016-05-23; Last Update: 2016].

I. Misra, S. M. Moorthi, D. Dhar, and R. Ramakrishnan. An automatic satellite image registration technique based on harris corner detection and random sample consensus (ransac) outlier rejection model. In *Proc. 1st Int. Conf. Recent Advances in Information Technology (RAIT), 2012*, pages 68–73, 2012.

D. Miyazaki, M. Kagesawa, and K. Ikeuchi. Transparent surface modeling from a pair of polarization images. *IEEE Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, 26(1): 73–82, 2004.

H. P. Moravec. *Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover.* PhD thesis, Stanford University, Stanford, CA, USA, 1980.

N. J. Morris and K. N. Kutulakos. Reconstructing the surface of inhomogeneous transparent scenes by scatter-trace photography. In *Proc. IEEE 11th Int. Conf. Computer Vision (ICCV), 2007*, pages 1–8, 2007.

N. J. W. Morris. *Image-based water surface reconstruction with refractive stereo.* PhD thesis, University of Toronto, 2004.

N. J. W. Morris and K. N. Kutulakos. Dynamic refraction stereo. In *Proc. 10th IEEE Int. Conf. Computer Vision (ICCV), 2005*, volume 2, pages 1573–1580, 2005.

H. Murase. Surface shape reconstruction of an undulating transparent object. In *Proc. IEEE 3rd Int. Conf. Computer Vision (ICCV), 1990*, pages 313–317, 1990.

H. Murase. Surface shape reconstruction of a nonrigid transport object using refraction and motion. *IEEE Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, 14(10):1045–1052, 1992.

S. K. Nayar, X.-S. Fang, and T. Boult. Separation of reflection components using color and polarization. *Int. Journal of Computer Vision (IJCV)*, 21(3):163–186, 1997.

D. Nehab, T. Weyrich, and S. Rusinkiewicz. Dense 3d reconstruction from specularity consistency. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2008*, pages 1–8, 2008.

T. Niemann. Radial distortion correction. `http://www.uni-koeln.de/~al001/radcor_files/hs100.htm`, 2006. [Online; Accessed: 2016-02-24; Last Update: 2006].

K. Okamoto, H. Madarame, and T. Hagiwara. Self-induced oscillation of free surface in a tank with circulating flow. *Trans. the Japan Society of Mechanical Engineers Series C*, 57:647–653, 1991.

K. Okamoto, H. Madarame, and T. Hagiwara. Self-induced sloshing in a tank with circulating flow. In *Fluid-Structure Vibration and Sloshing*, volume 232, pages 5–11, 1992.

K. Okamoto, H. Madarame, and M. Fukaya. Flow pattern and self-induced oscillation in a thin rectangular tank with free surface. *Journal of Fac. of Eng., Univ. of Tokyo*, 17(2):123–142, 1993.

K. Okamoto, H. Madarame, and M. Fukaya. Growth mechanism of self-induced sloshing caused by jet in rectangular tank : 2nd report, multimode sloshing caused by horizontal rectangular jet. *Trans. the Japan Society of Mechanical Engineers Series B*, 62(599):2604–2611, 1996.

K. Okamoto, S. Saeki, H. Madarame, H. Hu, T. Saga, and T. Kobayashi. Analysis on the self-induced sloshing using particle image velocimetry. In *Proc. 8th Int. Conf. Nuclear Engineering*, volume 6A, pages 513–518, 2000.

M. Okutomi and T. Kanade. A multiple-baseline stereo. *IEEE Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, 15(4):353–363, 1993.

OpenMVG authors. cameras-Pinhole camera model. `http://openmvg.readthedocs.io/en/latest/openMVG/cameras/cameras/`, 2015. [Online; Accessed: 2016-02-24; Last Update: 2015].

M. Oren and S. K. Nayar. A theory of specular surface geometry. *Int. Journal of Computer Vision (IJCV)*, 24(2):105–124, 1997.

J. Park, I. Sunghyuk, and J. Hyung. Piv measurement on externally-induced 2d sloshing in a rectangular tank. In *Proc. 8th China-Japan-Korea Student Symposium*, 2014.

D. Parks and J.-P. Gravel. Corner detection. *International Journal of Computer Vision (IJCV)*, 2004.

T. P. Patel and S. R. Panchal. Corner detection techniques: An introductory survey. *Int. Journal of Engineering Development and Research (IJEDR)*, 2:3680–3686, 2014.

A. Penate-Sanchez, J. Andrade-Cetto, and F. Moreno-Noguer. Exhaustive linearization for robust camera pose and focal length estimation. *IEEE Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, 35(10):2387–2400, 2013.

S. Rahmann and N. Canterakis. Reconstruction of specular surfaces using polarization imaging. In *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition (CVPR), 2001*, volume 1, pages 1–149, 2001.

E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *Proc. 9th European Conf. Computer Vision (ECCV) - part 1, 2006*, pages 430–443. Springer, 2006.

S. Roth and M. J. Black. Specular flow and the recovery of surface structure. In *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition (CVPR), 2006*, volume 2, pages 1869–1876, 2006.

S. Rozenfeld, I. Shimshoni, and M. Lindenbaum. Dense mirroring surface recovery from 1d homographies and sparse correspondences. *IEEE Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, 33(2):325–337, 2011.

S. Saeki, H. Madarame, K. Okamoto, and N. Tanaka. Numerical study on the growth mechanism of self-induced sloshing caused by horizontal plane jet. In *Proc. ASME FluidsEng. Div. Summer Meeting*, 1998.

S. Saeki, H. Madarame, and K. Okamoto. Excitation mechanism of self-induced sloshing caused by horizontal plane jet. *Trans. the Japan Society of Mechanical Engineers Series B*, 1999.

S. Saeki, H. Madarame, and K. Okamoto. Self-induced sloshing excited by a horizontally injected plane jet. *Journal of Fluid Mechanics*, Volume 448:81–114, 2001.

T. Saga, H. Hu, T. Kobayashi, S. Murata, K. Okamoto, and S. Nishio. A comparative study of the piv and ldv measurements on a self-induced sloshing flow. *Journal of visualization*, 3(2): 145–156, 2000a.

T. Saga, H. Hu, T. Kobayashi, S. Segawa, and N. Taniguchi. Research on the self-induced sloshing phenomena in a rectangular tank. In G. Carlomagno and I. Grant, editors, *Proc. 9th Int. Symposium on Flow visualisation*, page 259, 2000b.

M. Saito, Y. Sato, K. Ikeuchi, and H. Kashiwagi. Measurement of surface orientations of transparent objects using polarization in highlight. *Systems and Computers in Japan*, 32(5):64–71, 2001.

A. C. Sanderson, L. E. Weiss, and S. K. Nayar. Structured highlight inspection of specular surfaces. *IEEE Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, 10(1):44–55, 1988.

S. Savarese and P. Perona. Local analysis for 3d reconstruction of specular surfaces. In *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition (CVPR), 2001*, volume 2, pages 738–745, 2001.

S. Savarese, M. Chen, and P. Perona. Local shape from mirror reflections. *Int. Journal of Computer Vision (IJCV)*, 64(1):31–67, 2005.

D. Scharstein and R. Szeliski. Stereo matching with nonlinear diffusion. *Int. Journal of Computer Vision (IJCV)*, 28(2):155–174, 1998.

P. Schiebener, J. Straub, J. L. Sengers, and J. Gallagher. Refractive index of water and steam as function of wavelength, temperature and density. *Journal of physical and chemical reference data*, 19(3):677–717, 1990.

J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, et al. Fiji: an open-source platform for biological-image analysis. *Nature methods*, 9(7):676–682, 2012.

J. Schindelin, C. T. Rueden, M. C. Hiner, and K. W. Eliceiri. The ImageJ ecosystem: An open platform for biomedical image analysis. *Molecular reproduction and development*, 82(7-8): 518–529, 2015.

C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *Int. Journal of Computer Vision (IJCV)*, 37(2):151–172, 2000.

C. A. Schneider, W. S. Rasband, K. W. Eliceiri, et al. NIH Image to ImageJ: 25 years of image analysis. *Nat methods*, 9(7):671–675, 2012.

H. Schultz. Retrieving shape information from multiple images of a specular surface. *IEEE Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, 16(2):195–201, 1994.

G. Shakhnarovich. *Learning task-specific similarity.* PhD thesis, Massachusetts Institute of Technology, 2005.

O. H. Shemdin. Measurement of short surface waves with stereophotography. In *Proc. Conf. Engineering in the Ocean Environment OCEANS, 1990*, pages 568–571, 1990.

J. Shi and C. Tomasi. Good features to track. In *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition (CVPR), 1994*, pages 593–600, 1994.

B. Sirisha and B. Sandhya. Evaluation of distinctive color features from harris corner key points. In *Proc. IEEE 3rd Int. Advance Computing Conference (IACC), 2013*, pages 1287–1292, 2013.

S. M. Smith and J. M. Brady. Susan: a new approach to low level image processing. *Int. Journal of Computer Vision (IJCV)*, 23(1):45–78, 1997.

L. Sroba, R. Ravas, and J. Grman. The influence of subpixel corner detection to determine the camera displacement. *Procedia Engineering*, 100:834–840, 2015.

T. Stich, A. Tevs, and M. A. Magnor. Global depth from epipolar volumes–a general framework for reconstructing non-lambertian surfaces. In *3DPVT*, pages 916–923. Citeseer, 2006.

R. Szeliski. *Computer vision: algorithms and applications.* Springer Science & Business Media, 2010.

A. Takizawa and S. Kondo. *Computer discovery of the mechanism of flow-induced sloshing.* American Society of Mechanical Engineers, 1995.

A. Takizawa, S. Koshizuka, and S. Kondo. Generalization of physical component boundary fitted co-ordinate (pcbfc) method for the analysis of free-surface flow. *Int. Journal for Numerical Methods in Fluids*, 15(10):1213–1237, 1992. ISSN 1097-0363.

M. Tarini, H. P. Lensch, M. Goesele, and H.-P. Seidel. *3D acquisition of mirroring objects*. Max-Planck-Institut für Informatik, 2003.

I. Thormählen, J. Straub, and U. Grigull. Refractive index of water and its dependence on wavelength, temperature, and density. *Journal of physical and chemical reference data*, 14(4): 933–945, 1985.

M. Trajković and M. Hedley. Fast corner detection. *Image and vision computing*, 16(2):75–87, 1998.

A. Treuille, A. Hertzmann, and S. M. Seitz. Example-based stereo with general brdfs. In *Proc. 8th European Conf. Computer Vision (ECCV), 2004*, pages 457–469. Springer, 2004.

Y. Tsin, S. B. Kang, and R. Szeliski. Stereo matching with reflections and translucency. In *Proc. IEEE Computer Society Conf. Computer Society Conf. Computer Vision and Pattern Recognition (CVPR) (CVPR), 2003*, volume 1, pages I–702, 2003.

T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors: a survey. *Foundations and Trends in Computer Graphics and Vision*, 3(3):177–280, 2008.

A. R. Várkonyi-Kóczy. Fuzzy logic supported corner detection. *Journal of Intelligent and Fuzzy Systems*, 19(1):41–50, 2008.

Y. Vasilyev, Y. Adato, T. Zickler, and O. Ben-Shahar. Dense specular shape from multiple specular flows. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2008*, pages 1–8, 2008.

H. Wang and M. Brady. Real-time corner detection algorithm for motion estimation. *Image and Vision Computing*, 13(9):695–703, 1995.

H. Wang, M. Liao, Q. Zhang, R. Yang, and G. Turk. Physically guided liquid surface modeling from videos. In *ACM Trans. Graphics (TOG)*, volume 28. ACM, 2009.

J. Wang and K. J. Dana. Relief texture from specularities. *IEEE Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, 28(3):446–457, 2006.

Z. Weixing, M. Changhua, X. Libing, and L. Xincheng. A fast and accurate algorithm for chessboard corner detection. In *Proc. IEEE 2nd Int. Congress on Image and Signal Processing, 2009*, pages 1–5, 2009.

Wikipedia. Lock (water navigation) — Wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=Lock`, 2016. [Online; Accessed: 2016-04-02; Last Update: 2016-05-12].

Z. Wu and G. A. Meadows. 2-d surface reconstruction of water waves. In *Egineering in the Ocean Environment*, pages 461–4214. Ocean Engineering Laboratory, College of Engineering, University of Michigan, 1990.

R. Yang, M. Pollefeys, and G. Welch. Dealing with textureless regions and specular highlights-a progressive space carving scheme using a novel photo-consistency measure. In *Proc. IEEE 9th Int. Conf. Computer Vision (ICCV), 2003*, volume 1, pages 576–584, 2003.

X. Zhang and C. S. Cox. Measuring the two-dimensional structure of a wavy water surface optically: A surface gradient detector. *Experiments in Fluids*, 17(4):225–237, 1994.

X. Zhang and T. Zhou. Generic scheimpflug camera model and its calibration. In *Proc. IEEE Int. Conf. Robotics and Biomimetics (ROBIO), 2015*, pages 2264–2270, 2015.

Z. Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, 22(11):1330–1334, 2000.

J. Y. Zheng and A. Murata. Acquiring 3d object models from specular motion using circular lights illumination. In *Proc. IEEE 6th Int. Conf. Computer Vision (ICCV), 1998*, pages 1101–1108, 1998.

J. Y. Zheng and A. Murata. Acquiring a complete 3d model from specular motion under the illumination of circular-shaped light sources. *IEEE Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, 22(8):913–920, 2000.

J. Y. Zheng, A. Murata, Y. Fukagawa, and N. Abe. Reconstruction of 3d models from specular motion using circular lights. In *Proc. IEEE 13th Int. Conf. Pattern Recognition, 1996*, volume 1, pages 869–873, 1996.

J. Y. Zheng, Y. Fukagawa, and N. Abe. 3d surface estimation and model construction from specular motion in image sequences. *IEEE Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, 19(5):513–520, 1997.

T. E. Zickler, J. Ho, D. J. Kriegman, J. Ponc, and P. N. Belhumeur. Binocular helmholtz stereopsis. In *Proc. IEEE 9th Int. Conf. Computer Vision (ICCV), 2003*, pages 1411–1417, 2003.

A. Zisserman, P. Giblin, and A. Blake. The information available to a moving observer from specularities. *Image and vision computing*, 7(1):38–42, 1989.

D. E. Zongker, D. M. Werner, B. Curless, and D. H. Salesin. Environment matting and compositing. In *Proc. 26th annual conference on Computer graphics and interactive techniques*, pages 205–214. ACM Press/Addison-Wesley Publishing Co., 1999.