Utrecht University

Bachelor thesis

Using Free Surface Synthetic Schlieren to Observe the Surface Waves of Water

Authors:
Joel Jeene Castro
Studentnumber 3832953
Utrecht University
i.c.w.
NIOZ

Supervisors:
Leo R. Maas
Utrecht University
and
Sander van Oers
NIOZ

15 June 2016

Abstract

Free-Surface Synthetic Schlieren (FS-SS) is a non-intrusive method for measuring the three-dimensional space-time structure of fluids, that functions using the laws of optics. A method that is capable of observing water surface wave topography, like this, is a useful lab-tool to NIOZ, a research organization that specializes in the multidisciplinary study of ocean processes. We sought to provide them with this instrument, by building it. In this paper we describe the theory behind the method, as presented in literature, its strengths, its limits, and its optimal use and operation. We perform some tests with the instrument to calibrate it. Using these tests we conclude our efforts were successful, having built an FS-SS setup that functions within acceptable range of error.

Contents

Chapter 1

Introduction

At NIOZ, or the Royal Netherlands Institute for Sea Research (The acronym is quite lost in translation), they study the ocean. As the ocean is a big place, there are many departments within the organization that concern themselves with marine biology, microbiology, ecology, and the like. These studies can take many forms and quickly branch out into specialized groups researching a very specific narrow field. All of these fields, however benefit from a greater understanding of the most basic element in the ocean, the water itself.

Hydrodynamics has been a greatly studied topic since the conception of modern science, and yet very real problems still exist that complicate the successful execution of its research. One is the chaotic nature of fluids and similar large ensembles, which can take the simple mechanics of movement and turn it into a confusing mess of trends, estimates and probability. This is unavoidable. Another major problem is the practical difficulty of observing these mechanics. Classical mechanics discusses discrete objects and their movements, preferably perfect spheres in a perfect vacuum. Not so in Fluid Dynamics, where the environment and the object of our interest differ only in where they are at a precise instant and at what speed they may have been moving in that same instant. Even observation of macroscopic properties is not an easy thing to do, as for example the flow of water can generally not be seen from the water itself, but must be deduced from foreign objects being carried by it. Surface waves, while at least a clear indicator something is happening, are difficult to accurately observe as well, as while their movement may be two dimensional, their structure is inherently three dimensional, and has no vantage point from which the full three dimensions can be accurately described in a traditional visual manner. That is to say, if you were to take a photo, you could glean only very little information from it by just looking

To fill this gap in NIOZ‘s arsenal, it was decided to use a non-traditional optical way to visualize the topography of waves, by building a set up that uses the Free Surface Synthetic Schlieren method to accurately measure the 3D structure of waves through purely 2D images. There are other methods too, but this is the most elegant solution, as it requires no extra-ordinary laboratory set up or very expensive equipment. In this paper, we will discuss this technique, its strengths and limitations, and our application of it to make a functional instrument of observation. We will also discuss what this apparatus may do for NIOZ, and how it might be used for further research.

Chapter 2

FS-SS Theory

2.1 The Basics

Free Surface Synthetic Schlieren, for the sake of brevity from here on referred to as FS-SS, comes from a long line of techniques using optical methods to glean information from otherwise hard to observe phenomena in transparent media.
One of the earliest texts on the use of light to detect differences in air density was from Robert Hooke[3]. He observed that differing densities in the air caused the sun to cast shadows out of, well, thin air. Schlieren photography operates in this same manner, as it uses these shadows to study the aerodynamics around objects, like airplane designs in a wind tunnel. Synthetic Schlieren, and FS-SS are crucially different to these in that they do not rely on direct observation of shadows or light as its Schlieren Photography ancestors did. Instead of following the general trend of the light being focused into some places and away from others, it uses the very specific change in the light‘s direction to determine what refractive process caused that, and from that it reconstructs the situation that it's trying to visualise.
Synthetic Schlieren is used to observe the flow of solution that has a different density than its solvent. It uses the refraction of light between the solution and solvent as an optical trick. The mixture of differing density is positioned between a camera and a known image. Because the specimen doesnt universally have the same refractive index, some parts of the image may appear warped to a camera that is focused on the image through the sample. This warping of the image is then analyzed by a Digital Image Correlation (DIC) software, and used to calculate the precise location of this disturbance. FS-SS is based on the same physical principles, but instead it takes a homogenous fluid with an equally homogeneous refraction index, rather than a solution, and uses the refraction of light to determine the *shape* of this fluid, rather than its composition. Here we will discuss, prove and review the theory that we later apply to our experiment, as described in [1]. We would like to stress that unless specifically mentioned, this next section of theoretical exposition is paraphrased from that paper. The refraction of light by an

interface of two materials is determined by the material's, in this case, the water's and air's, refractive index, and the angle of incidence of the lightbeam that is refracted. This refractive index is a constant, so in a body of water the only variable is this angle of incidence. As the surface is disturbed and a wave passes over it, the surface at a single point A will change from its equilibrium state of horizontal stillnessl to a sinoidal wave. A ray of light that would have made a straight vertical lign up through the still surface might now make an angle with the disturbed interface and as such, refract. If one were to look at the bottom of a swimming pool the tiles on the bottom would appear as constantly warping due to the refraction occuring at the surface. This visual warping we call displacement, and it is this visual effect that we use to reconstruct an image of the surface. This is one of the two main pillars of the FS-SS method, which depends on the following two main pillars:

1. The relation between the surface gradient of the liquid and the displacement of the underlying image, as explained above
2. Using this surface gradient ∇h to reconstruct the surface height, by least square integrating it to $h(x, y)$

We will explain the former in the following section. As the latter ties directly into the computational programs we used for this, it will be discussed in the appropriate section **??**.

2.2 Theoretical setup

In this scenario, we have a camera suspended directly above the body of water we wish to observe. Underneath the water is the pattern we reference for distortion. The pattern consists of a series of dots "randomly" distributed to make sure that none of the patterns repeats itself. This situation is shown schematically in Figure 2.1, on the next page. The pattern itself we discuss in the section 4.3. The rub of the technique is to determine the optical displacement field $\delta r(x, y)$ induced by the refraction of the light scattered from a pattern located at $z = 0$ through the interface $z = h(x, y)$. While the surface is still, some refraction will still occur for each point M of the pattern that isn't perfectly beneath the camera. This default "false" image we denote with the virtual points M' corresponding to true point M. The refraction of light that occurs on the interface of the water depends on the angle of incidence the lightray encounters the interface with. This angle of incidence changes with the angle between the waters surface and the horizontal, which is analogous with ∇h. When we generate a surface disturbance, like a sinusoidal wave, the surface elevation topography changes with the wave, and new virtual objects will be generated. The precise detail of what happens to the resulting image we will explain in 2.4. They will be referred to as M''. The displacement δr then simply comes down to relating $\delta r = \mathbf{M'M''}$, with the surface gradient ∇h at the same point.

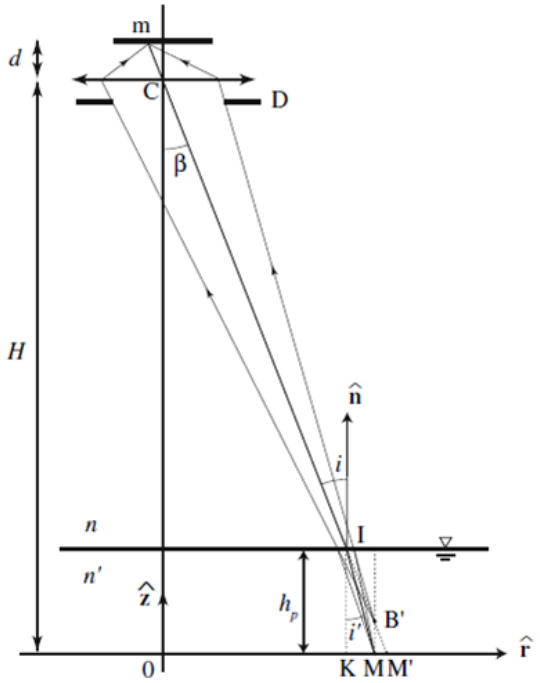


Figure 2.1 – Figure 1: A schematic representation of a typical setup with an undisturbed surface, showing the COM plane with all its accompanying angles, vectors, and values. The lines that streak the edge of the Diaphragm D do not serve a purpose for our calculations, but are there to show how a focused image m is projected onto the CCD through lens C. This image was originally published by Moisy et al. in [1]

We know that water has a refractive index n' that is greater than the refractive index of air n. This means that wave crests and troughs will respectively act as concave and convex lenses. We define our Cartesian frame with its origin directly underneath the camera, so that the vertical axis Z crosses both the origin and the camera.The x and y axes where $z = 0$ correspond to the pattern plane. In this schematic view we consider a surface, where the pattern surface distance h_p is equal to the liquid depth h_0. This need not be the case as there may be layers between the actual pattern and the body of water which have a different refractive index from water, like the transparent plastic bottom of the water vessel used. Corrections are easily made for this, and we will demonstrate them later, in section 2.3. For now, we focus on this case to aid clarity of comprehension.

2.3 Approximations

There are three approximations we must make to get this theory to work. They are based on assuming the following three parameters are small: The paraxial angle β, the surface slope γ and the relative surface deformation. The resulting approximations are:

1. The Paraxial approximation: The pattern-camera distance H is far larger than the field size L. This yields a maximum paraxial angle $\beta_{max} \simeq L/(\sqrt{2}H) << 1$

2. The Weak Slope Approximation: The angle γ between the unit vector normal to the interface $\hat{n}$ and the vertical vector $\hat{z}$ is small. This means that the surface slope θ measured in the incidence plane is also small. γ and θ are the same in the case of the still surface.

3. The Weak amplitude approximation: If we formulate a formula for the surface height $h(x, y) = h_p + \eta(x, y)$, where $\eta(x, y)$ is the amplitude of the surface waves as measured from the equilibrium state of h_p we assume that the amplitude$|\eta|$ is small compared to the mean surface height h_p.

We use these assumptions to linearize the relation between the surface gradient ∇h and the virtual horizontal displacement δr of a feature in the plane z = 0. First we consider the refracted image of the pattern at $z = 0$ as observed through a flat interface $z = h_p$, as seen in Figure 2.1. For each point M, located in $(x_m, y_m, 0)$ one can define a vertical incidence plane with camera C, origin O and imagepoint M as **COM**. This plane crosses the pattern plane $z = 0$ along the radial unit vector $\hat{r} = \mathbf{OM}/|\mathbf{OM}|$. This plane can be seen as it stands in the threedimensional situation in Figure 4.3a. If I could briefly redirect your attention to figure 2.1 again, you can see that the image of M on the camera sensor plate (or CCD) is referred to as m. The corresponding virtual object B' is located above, not on the pattern plane in the position $(x_M, y_M, \alpha h_p)$. This α is a coefficient that depends on the refraction indices of the materials above and below the interface. The relation is deduced from Figure 2.1. For the sake of clarity however we focus on the relevant bit of the installation as shown in Figure 2.2.

One could see two triangles in this image, $\triangle IMK$, and the triangle that would form from IB', and the horizontal projection from B' onto IK, and from that point to I again. We call that point A for now. We wish to know the length of **KA**. To find this we write it as: $\mathbf{KA} = h_p - \mathbf{IA}$. To find it we must first define $\mathbf{IA} = \mathbf{KM}/tan(i)$. KM can be defined through the greater triangle $\triangle IKM$, which gives us $\mathbf{KM} = h_p tan(i')$. Combining these two grants us

$$\mathbf{IA} = h_p \frac{tan(i')}{tan(i)} = h_p \frac{sin(i')cos(i)}{cos(i')sin(i)} \tag{2.3.1}$$

Snell's law dictates that $nsin(i) = n'sin(i')$, so $sin(i') = \frac{n}{n'}sin(i)$ Combining this and 2.3.1, we get $\mathbf{IA} = h_p \frac{nsin(i)cos(i)}{n'cos(i')sin(i)}$

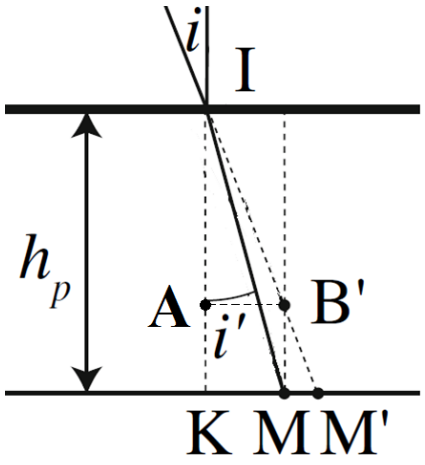


Figure 2.2 – A close up of the light's behaviour beneath and relating to the surface at rest

The two sines cancel one another out, and since we are talking about very small angles i, and i', we can approximate that $\frac{cos(i)}{cos(i')} \simeq 1$. This leaves us with $\mathbf{IA} = h_p n/n'$. We set out to calculate α, which we do here:

$$\mathbf{KA} = \mathbf{MB'} = \mathbf{IK} - \mathbf{IA} = h_p - h_p \frac{n}{n'} = (1 - \frac{n}{n'})h_p = \alpha h_p \tag{2.3.2}$$

This vertically shifted plane of $z = \alpha h_p$, where all B' are situated is where we will focus the camera. This is a small concession to practicality, as our calculations still use the horizontal displacement of the pattern. To this end we extrapolate the line that runs from the camera to B' and where it would cross the pattern plane we call M'. For a still, undisturbed surface, this displacement is always in the radial direction, and can be read from figure 2.2 as:

$$\mathbf{MM'} = h_p(tan(i) - tan(i'))\hat{r} \tag{2.3.3}$$

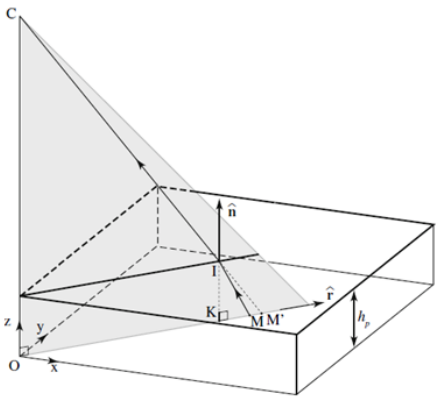
i is equal to the paraxial angle β in this situation, so using equation 2.3.2 and the paraxial approximation simplifies 2.3.3, using $cos(i) \simeq cos(i') \simeq 1$, and Snells law still reigns, so 2.3.3 becomes

$$h_p(sin(i)/1 - \frac{n}{n'}sin(i))\hat{r} = h_p sin(i)\alpha\hat{r} \simeq h_p i\alpha\hat{r} \tag{2.3.4}$$

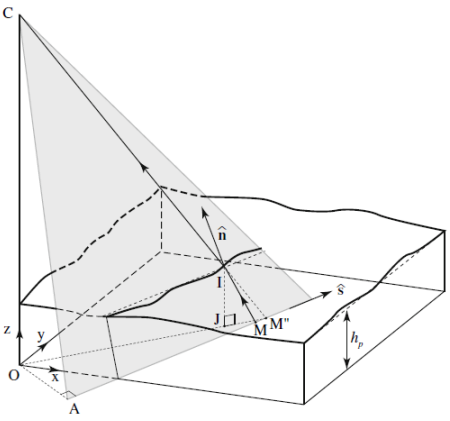
This is the equation with which we can determine the position of any point M from the image M^prime, given we know h_p, H and the Origin.

2.4 The Refracted image from a deformed surface

In the previous section we discussed some of the basic theory, and established a backdrop against which we can paint the bigger and more important problem: the refraction through a water surface that is not still, but disturbed, and in motion. In this scenario, we want to determine a new virtual object $\mathbf{M}''$, which is analogous to $\mathbf{M}'$ in the previous problem. One immediate difference is that the incidence plane is not (necessarily) vertical. Save for some lucky instances, the plane will not include optical axis z. Consider the contrast in figure 4.3a and 4.3b, where the two are shown schematically. The topography



(a) the setup for a still surface, shown in three dimensions.



(b) The setup for a disturbed surface, shown in three dimensions.

of the water's surface has no regard for the vantage point of our camera. Because of this, a wave might go in the $\hat{x}$ direction and in doing so traverse the length of the field of view. This causes a surface gradient that is not always oriented radially from the Origin, and as such will bend a light beam to have a non radial component, which complicates things a bit. The incident plane that results from this, CAM, is defined by the camera C, M, and the unit vector $\hat{n}$, which is the normal of the surface where it is hit by the light ray that goes from M to C. The virtual point M'' where the ray seems to originate from and actual point M make for the line MM''. A is the projection of origin O onto the extrapolation line MM''.When one considers the two dimensional view of plane CAM, as shown in Figure 2.4, it becomes similar to figure 2.1.

$\hat{n}$ can be defined as follows:

$$\hat{n} = \frac{\hat{z} - \nabla h}{\sqrt{1 + |(\nabla h)|^2}} \tag{2.4.1}$$

Earlier, we already assumed that the gradient of the slopes would be quite small. So since $|(\nabla h)| << 1$, we can say $\sqrt{1 + |(\nabla h)|^2} \simeq 1$. Ergo:

$$\nabla h = \hat{z} - \hat{n} \tag{2.4.2}$$

The displacement of M to M'' occurs along a new vector $\hat{s}$, which is defined as the intersection of the incident CAM plane and the pattern plane, where z=0. This yields:
$\mathbf{s} = \mathbf{OM}/H - \nabla h$
As $\hat{n}$, CM'' and $\hat{s}$ are all in the incident plane CAM, we could define

$$\hat{n} = a\hat{s} + b\frac{\mathbf{CM''}}{|\mathbf{CM''}|} \tag{2.4.3}$$

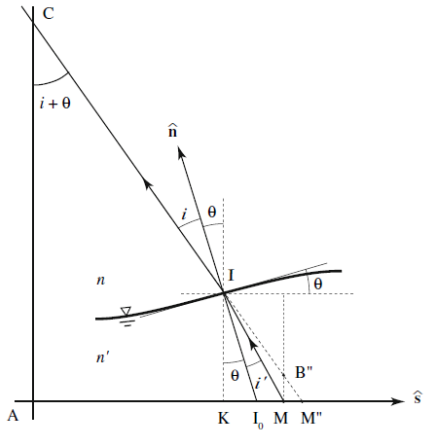


Figure 2.4 – The components of the CAM plane

It will become clear as to why this is useful later. For now, we can determine coefficients a and b through the geometrical relations as gleaned from figure 2.4:

$$\hat{n} \cdot \hat{s} = -sin(\theta) \tag{2.4.4}$$

$$\hat{n} \cdot \frac{\mathbf{CM''}}{|\mathbf{CM''}|} = -cos(i) \tag{2.4.5}$$

$$\hat{s} \cdot \frac{\mathbf{CM''}}{|\mathbf{CM''}|} = sin(i+\theta) \tag{2.4.6}$$

Where θ is the angle between the horizontal and the surface at point I in the inclined plane. This is important, as this θ is not the same as the absolute angle β between $\hat{n}$ and $\hat{z}$. the paraxial angle between C and I then logically becomes the sum of the angle between $\hat{n}$ and line CI and the angle between $\hat{n}$ and the vertical in CAM, θ. These relations yield the following equation:

$$\hat{n} = \frac{sin(i+\theta)cos(i) - sin(\theta)}{cos^2(i+\theta)}\hat{s} + \frac{sin(i+\theta)sin(\theta) - cos(i)}{cos^2(i+\theta)}\frac{\mathbf{CM''}}{|\mathbf{CM''}|} \tag{2.4.7}$$

Our earlier assumptions and approximations still hold true. This means that using the weak slope approximation (θ is very small) and the paraxial approximation (And so is β, or in this formula, $i+\theta$, which also implies that i, too, is small enough to approximate) we can reduce all the sin(x) in equation 2.4.7 to x and all the cos(x) to 1. This yields

$$\hat{n} = i\hat{s} - \frac{\mathbf{CM''}}{|\mathbf{CM''}|} \tag{2.4.8}$$

Now that we have redefined $\hat{n}$, we can also redefine ∇h. First, because of the quite large pattern-camera distance, and very small maximum paraxial angle it is approximately true that $|\mathbf{CM''}| \simeq H$, so we can rewrite 2.4.8 as

$$\hat{n} = i\hat{s} - \frac{\mathbf{CM''}}{H} \tag{2.4.9}$$

Considering $\mathbf{CM''} = \mathbf{CO} + \mathbf{OM''}$ by definition, we can further rewrite 2.4.8 to

$$\hat{n} = i\hat{s} - \frac{\mathbf{CO} + \mathbf{OM''}}{H} \tag{2.4.10}$$

Remembering 2.4.2 we get

$$\nabla h = \hat{z} - \hat{n} = \hat{z} - i\hat{s} + \frac{\mathbf{CO}}{H} + \frac{\mathbf{OM''}}{H} \tag{2.4.11}$$

CO is simply a vertical line of length H, pointing down. This makes $\mathbf{CO}/H$ the equal opposite of $\hat{z}$, so they cancel one another out. $\hat{s}$ is the unit vector in the direction of the line MM'', so it could be written as $\hat{s} = \mathbf{MM''}/|\mathbf{MM''}|$.This culminates in the final rewrite:

$$\nabla h = \frac{\mathbf{OM''}}{H} - i\frac{\mathbf{MM''}}{|\mathbf{MM''}|} \tag{2.4.12}$$

Analogous to 2.3.3 with Figure 2.1, Figure 2.4 shows us that the displacement distance $\mathbf{MM''}$ is given as

$$\mathbf{MM''} = \mathbf{KM''} - \mathbf{KM} = \mathbf{IK}(tan(\theta+i) - tan(\theta+i')) \tag{2.4.13}$$

K is the projection of I onto the line $\mathbf{MM''}$. Note that this is not necessarily a vertical line. The vertical projection of I on the pattern plane we call J. J and K can be, but arent necessarily, the same. As you can see from figure 2.4, we have $\mathbf{IK} = \mathbf{II_0}cos(\theta)$, with $\mathbf{II_0}$ as the line between I and $\mathbf{MM''}$ in the $\hat{n}$ direction, and $\mathbf{II_0} = h(I)/cos(\gamma)$ with

γ the angle between $\hat{n}$ and the vertical $\hat{z}$. Note again, that this $\gamma \neq \theta$. Finally, **IJ** is simply the height of point I, or h(I). This lets us rewrite 2.4.13 to:

$$\mathbf{MM''} = h(I)\frac{cos(\theta)}{cos(\gamma)}(tan(\theta + i) - tan(\theta + i\prime))\hat{s} \tag{2.4.14}$$

Again with our approximations, but they hold water yet. Since we are dealing with small angles and weak deformations, we can say that $h(I) \simeq hp$, 2.4.14 analogous with 2.3.4 becomes

$$\mathbf{MM''} = \alpha h_p i\hat{s} \tag{2.4.15}$$

A notable, indeed the notable difference between Eq. 2.3.3 and Eq. 2.4.15 is that in the former, the image is displaced along $\hat{r}$, while the latter displaces the image along $\hat{s}$. In Figure 2.4 one can see that here too, a virtual object $\mathbf{B''}$ is generated much like in the still surface situation. A difference however, is that this height $\mathbf{B''}$ is not universal, and is dependent on the slope of the surface. This means that $\mathbf{B''}$ cannot be exactly focused. For weak deformations however, and that approximation still stands, the focal surface still remains at the old $z = \alpha h_p$ we determined for $\mathbf{B'}$. The dots in the pattern however are larger than one single pixel so that this small out of focus effect does not affect our measurements, and can be neglected. From equation 2.4.15 one can write $i\hat{s} = \mathbf{MM''}/(\alpha h_p)$. $\hat{s}$ is still $\frac{\mathbf{MM''}}{|\mathbf{MM''}|}$, so we can take Equation 2.4.12 and rewrite it as

$$\nabla h = \frac{\mathbf{OM''}}{H} - \frac{\mathbf{MM''}}{\alpha h_p} \tag{2.4.16}$$

Here we introduce the relation $\mathbf{OM''} = \mathbf{OM'} + \mathbf{M'M''}$ and $\mathbf{MM''} = \mathbf{MM'} + \delta r$. From this follows $\delta r = \mathbf{M'MM''}$, and this is what we were going for from the start. This displacement of the image from the already warped image presented by a still surface to the image we see through a distorted surface is what will allow us to reconstruct a three dimensional model of the surface. Feeding these new definitions into 2.4.16 yields

$$\nabla h = -\delta r\left(\frac{1}{\alpha h_p} - \frac{1}{H}\right) + \frac{\mathbf{OM'}}{H} - \frac{\mathbf{MM'}}{\alpha h_p} \tag{2.4.17}$$

These last two terms cancel one another out, and leave us with a linear relation between ∇h and δr, namely

$$\nabla h = \frac{\delta r}{h*}, where \frac{1}{h*} = \frac{1}{\alpha h_p} - \frac{1}{H} > 0 \tag{2.4.18}$$

This shows that for our three first order approximations the displacement δr is proportional to the surface gradient ∇h. As in practice the camera is placed far above the waters surface ($H << \alpha h_p$), the $1/\alpha h_p$ is dominant in $1/h*$, and one can simplify it to $h* \simeq \alpha h_p$. At the beginning of this section, we remarked on the added difficulty that is

the bottom of the container adding another layer of refraction to our image. Consider the very realistic case of several layers, with water at the top, a glass plate beneath it, and between the glass plate and the pattern, a layer of air. Until now we have used the surface-pattern distance HP and the water depth h0 interchangeably, but as this new scenario implies, we cannot assume this any longer. If we place the camera far above the surface, we can consider only vertical light rays, so that the displacement of an image element in the direction δx is linearly dependent on the slope of the surface in that same direction $\delta h/\delta x$, and on the refractive indices of the respective materials. In the scenario we described above we would have the refractive indices of the air, water, and glass components, called n, n‘, and n_g respectively. The corresponding layers would, from pattern to surface be called h_a, h_g, and h_0. This would yield the total displacement:

$$\delta x = -h_0 + \frac{n}{n\text{‘}}h_0 + \frac{n}{n_g}h_g + \frac{n}{n}h_a \tag{2.4.19}$$

The resulting displacement components can be written in the form of 2.4.18, where the effective pattern-surface distance would be the sum of the layer thicknesses, weighted by the refraction index ratios, as shown here:

$$h_p = h_0 + \frac{n`}{n_g} h_g + \frac{n`}{n_a} h_a \tag{2.4.20}$$

These elements are not variables, but we must take note of this effect to ensure that we dont accidentally convince Digiflow the camera is further away from the pattern than it actually is.

2.4.1 FS-SS Limits

An important limiting factor of this technique is the so called invertibility condition. We have so far assumed that δr could simply be determined by the refracted image of our pattern. In practice, however, crossings of light could occur.
The waters surface acts like a lens. This is of course what our method is built upon, but if this lens is strong enough, it might focus two disparate rays from the same image point into the camera,giving two images to one point. This would severly confuse the image recognition program. Or two points might converge onto the same pixel on the CCD. This is what causes caustics, the bright spots on the surface of for example, the water of a swimming pool as shown in 2.5. Caustics would severely impact the usefulness of any of our measurements.
Their prevention necessitates the aforementioned invertibility condition. To ensure the proper function of our method, we need to know for certain that any given image m on the camera CCD originates from a single point M on the pattern. This condition is satisfied if the focal length of the lens that is the surface curvature is larger than the surface distance pattern of the field. If this is the case, for $H >> h_p$, which has always been part of our assumptions, no two rays from the same point can ever be sufficiently focused to both reach the camera. As found in [4] the relation between the curvature of a simple thin lens and and its focusing power P_lens, and therefore focus distance f is

Figure 2.5 – A visual example of caustics. In this case, the surface of a swimming pool, disturbed by the swimming of many holidaymakers. Image from [2]

$$P_{lens} = \frac{1}{f} = \frac{n_{lens} - n_0}{n_0} \left(\frac{1}{R_1} - \frac{1}{R_2} \right) \tag{2.4.21}$$

Here R_1 and R_2 are the curvatures of the two sides of a lens. In our case though, $R_2 = \infty$, as this lens only has one curve. That simplifies 2.4.21 to

$$\frac{1}{f} = \frac{n_{lens} - n_0}{n0R1}, or f = \frac{n_0 R_1}{n_{lens} - n_0} = \frac{R_1}{\alpha} \tag{2.4.22}$$

According to [5], the curvature of a curved surface κ relates to the radius of its osculating circle R_1 through the simple formula $R = 1/\kappa$. This allows us to rewrite 2.4.22 once again, this time to $f = \frac{1}{\alpha\kappa}$. the curvature of the surface is approximately $\kappa \simeq \frac{\partial^2 h}{\partial x^2}$
The invertibiliy condition mandates that the pattern surface distance must be smaller than the focal length of the surface, meaning that the surface obeys $h_p < h_c$, with $h_c = f = \frac{1}{\alpha \frac{\partial^2 h}{\partial x^2}}$

For a typical measurement it will be easy for hp to remain below the critical h_c. Since we are not able to know the nature of the surface deformation before we actually measure it we demonstrate a typical case to establish a guideline. When we do our tests we can always calculate the $\frac{\partial^2 h}{\partial x^2}$ and determine if it needs to be redone at a lower surface

height. In this example, if we were to measure a sinuisoidal plane wave of amplitude η and wavelength λ we would have $h(x, y) = h_p + \eta cos(2\pi x/\lambda)$, and

$$h_c = \frac{1}{\alpha \frac{\partial^2 h}{\partial x^2}} = \frac{\lambda^2}{4\pi^2 \eta \alpha} \tag{2.4.23}$$

Lets say that the amplitude of this wave is 1mm, and its wavelength 3cm. The critical pattern surface distance would be 9 cm. If we steer well clear of this and stick to a h_c of, say, 3 cm, any test we do can be checked for its h_c afterwards, as we don't know for sure the characteristics of a wave before measuring it, although we are confident that if anything, this estimate errs on the safe side of caution and any actual measurements we perform will probably have greater critical pattern-surface distances.

Chapter 3

The Reconstruction of the Surface

3.1 Digiflow

Digiflow is a Digital Image Correlation program, built specifically to facilitate flow imaging, being designed to rapidly analyze a stream of images with the same ease as a single pair [8]. This program is able to track picture elements from image to image. It takes this difference, and as it knows the time that has passed in between frames (this depends on the camera used), it reports the movement in the image as a velocity vector field v(x,y).

An important part of the function of this program, one we have as of yet not discussed, is its use of PIV analysis, and the differences and advantages this has over PTV. With PTV, or Particle Tracking Velocimetry, the aim is to identify individual points, or particles, and track those from frame to frame. Particle Image Velocimetry, or PIV, divides the incoming image into smaller so called Interrogation Windows of several pixels each, usually 16x16 or a multiple thereof. For each interrogation window, it searches for the pattern it contained in the original image in the new image. Of course there may be minute differences in movement between different elements in the window, be they because of actual difference of movement or because noise entered the measurement. The program takes all the data of these individual image elements, and using statistical methods it determines the most appropriate general velocity for that group of picture elements. It can also use this to deduce more complicated aspects of the fluid, like convergence, divergence, and vorticity, but they do not serve our purposes. It is a good thing to remember, as it may become relevant for other experiments. One could say that PTV matches and follows particles in a Lagrangian sense, while PIV is a Eulerian approach to this problem.

The strengths of PIV are its resistance to noise, and its higher velocity resolution than PTV. These benefits come from its generalizing of groups of particles, as the accuracy of the displacement depends on the cell size and the distribution of the features within it, rather than the pixel resolution of the camera. Where a single velocity vector assigned to a particle in PTV would have a systematic error of ϵ, the error of the velocity vector in PIV would have a systematic error more akin to $\epsilon/\sqrt{N}$, where N is the amount of disparate particles in the image. Note that when speaking of elements, we do not mean pixels, but rather the individual recognizable and trackable points that make up the image. From hereon we will simply refer to them as particles, but remember that those particles are but coloured dots on a piece of paper for our purposes, not actual particles.

The spatial resolution of PIV is naturally inversely proportional to window size, as it is directly proportional to the amount of locations it can describe, and the bigger the interrogation window, the fewer you can populate the image with. The is to find

a sufficiently accurate reading of velocity, without giving up too much of the spatial resolution. For reasons explained in the section on noise and measurement insecurity, we choose interrogation windows of 32x32 pixels.

The downsides of PIV are the time and computation power needed to determine the optimal correlation per window. Particle tracking, as it only tracks single particles, takes far less time for the computer to process. However, time is not of the essence, so we may disregard that downside. As for the vertical movement, we remind the reader that we are not actually visually tracking a three dimensional flow, so the particles only move in the two dimensional sense.

One final argument in favor of PIV is the lack of structure in the PTV vector placement. The positions of the vectors depend entirely on the placements of the particles, which, to make them recognizable from one another, are not spread evenly over the image. This means that the field of displacement is less well known than we would like, and that any faulty match ups between frames are harder to recognize and correct.

Considering all these factors, we decided on using PIV to analyse the images.

One final remark on Digiflow: as its output is velocity vectors, and we are looking for displacement δr, we need to multiply the speed it comes out with (pixels/second) with the time this velocity has to displace the image element. This time is simply the time that passes between frames. In our case, but this could vary from experiment to experiment and camera to camera, it's 1/24 seconds.

3.2 Interrogation Windows, Particles, and Measurement Error

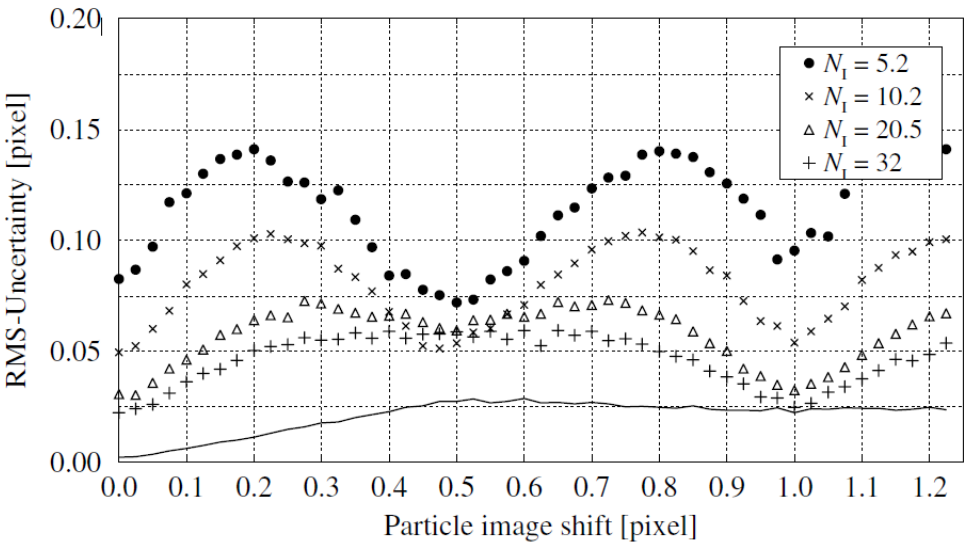


Figure 3.1 – Diagram from [11], showing the relation between the displacement of the image and the size of the interrogation window

We said in the previous subsection that we would defend our decision to use interrogation windows of 32^2 pixels. In general the greater the interrogation window, the greater the accuracy of its velocity vector. However, we still need to describe the space with enough spatial resolution. The camera we used has a resolution of $1380 * 1030$ pixels. Further details of the camera are described in **??**. If we were to use 64^2 windows, we would have a velocity field of 21*16 vectors, as opposed to the 42*32 that 32^2 windows allow us. As we said, we need to balance the spatial resolution and the velocity accuracy

to get the optimal results from our height reconstruction. We think that this proverbial sweet spot occurs in the 32^2 windows.

In "Particle Image Velocimetry: A Practical Guide" [9], the authors discuss the relation between Interrogation window size, particle size, and measurement uncertainty. They describe the total error associated with an observation as a sum of the error caused by an unknown bias in the system, and the random measurement uncertainty or noise. The bias error, as explained in the aforementioned book [10], can be ignored. This leaves only the random noise, which in following Raffel et al, we will call ϵ_{rms}. This is the error in the displacement vector that Digiflow turns out, and because Digiflow only has the information of the pixels in its CCD, the vector and its error are expressed in pixels.

Raffel has worked out the lowest possible random noise for different approaches to PIV, but before we name them we must remark that these uncertainties do not mandate our equipment to be equally accurate. Indeed, as they used Monte Carlo simulations to test the optimal accuracy of this method for each parameter, we can expect ours to be less accurate. On top of that, as we are observing a process in motion with a camera that takes only finite frames per second and has a non-zero shutter time, the program will have to interpret blurred images. We do not have the luxury of assuming the error of our measurements will be as low as Raffel's. Despite this we follow their standards in an effort of reaching the best measurements we can.

For a 32^2 window, the lowest ϵ_{rms} goes with a particle diameter of 2.5 pixels. This refers to the size of the dots on the camera CCD, not the actual composition of the dots in reality. The manner of the dots production will be handled in the section Set-up. Raffel expresses this error in pixel/pixel, meann Raffel expresses this error in pixels/pixels, meaning that for a displacement of 1 pixel, there's an uncertainty of $\epsilon_r ms$ pixels. In the lowest case scenario from the above sentence, that $\epsilon_r ms = 0.02 pixel/pixel$

The image density is another problem that needs discussion. Moisy recommends a density of particles N of at least 5 per interrogation window [6]. "PIV: A Practical Guide" Further explains this as they say that the chance of probability of a valid velocity vector detection exceeds 95% for our method of single exposure double frame PIV if the density of particles is greater than 5. However, they point out that in practice 3 or 4 recognized particles is sufficient to accurately perform an analysis.

The random noise of the measurements is also influenced by the particle density too, and as mentioned before, the two are roughly inversely related. While N=5 may be sufficient to be certain the velocity vectors actually happen, those vectors will have a far larger ϵ_{rms} than I would like. The error for each density Raffel tested also changes per displacement as they demonstrate in their diagram in Figure 3.1:

Over the entire diagram, the lowest ϵ_{rms} is associated with the highest density. However, as they used slightly smaller particles than us and we still need to fit these particles into the paper, and the difference in error is minimal, we shall strive to have 20 particles in each interrogation window.

This puts our ϵ_{rms} for particle density at 0.04 pixels. Although in this diagram it can be seen that this ϵ_{rms} varies with displacement, as long as we consider the worst case scenario, this is not a problem.

We previously found that when completely dependent of particle size, the ϵ_{rms} is 0.02 pixels. We can assume our measurement has the greater random noise of 0.04, to err on the side of caution.

3.3 Surface Reconstruction Software

We had mentioned earlier that the surface would be reconstructed using the least squares method of integration. So far we have established our methods of gaining the $\frac{\partial h}{\partial x}$ for each element of the surface. The chance that an error occurs during measurement is non zero, so at a single point the gradient could be incorrect. If we were to integrate linearly, along

a set path this error would prpagate along and throw off the entirety of the results. For this reason we use this least squares method, that is too detailed to go into here, but is expanded upon in [1], [6], and[7]. The files Digiflow produces after its displacement analysation have to be interpreted for use in Matlab. For this we use a suite of Matlab programs developed by Moisy [**?**]. They were originally made for use with the Digital Image Correlation programme DaVis, but we tailored them to work with Digiflow. These programs mainly busy themselves with the proper translation of Digiflow $.dfi$ files to vector fields we can perform mathematics with, but they are indispensable.

3.4 Surface Reconstruction Error

We have discussed the error inherent to measuring the displacement δr. This error is propagated through the reconstruction of the water surface linearly, due to the linear nature of integration used to solve equation 2.4.18. For greater surface deformations the strain in the generated image grows, and with it the possible error in our measurements. The error is not, however, variable locally with each increasing or decreasing surface gradient. Rather, it is an expression of the difficulty of accurately ascertaining the $\frac{\partial h}{\partial x}$ as the waves grow higher and the accompanying images grow blurrier. Therefore, we can more easily express this component of the error in the shape of the wave. Say that the average amplitude of the sinusoidal wave $\eta_{av} = \eta_0/\sqrt{2}$, where the maximum amplitude η_0 is constant. The surface gradient would grow greater with a smaller wavelength λ, which tells us they are inversely correlated. Likewise, if the wave‘s wavelength remains constant, the surface gradient correlates linearly with the waves maximum amplitude. Using both they offset each other, so we can say that the error correlates linearly with η_{av}/λ.

Furthermore, the error scales linearly with the size of the field L observed. We are still dealing with the computer though, and thus this error is also inversely correlated to the amount of collocation points N the computer discretizes this field L to. This N is not the same as resolution as we have used the word so far, but rather is the amount of interrogation windows applied to the surface. All these factors come together in the equation

$$\Delta\eta = c\epsilon\frac{L}{N}\frac{\eta_{av}}{\lambda},$$

Where c is a unitless constant that Moisy et al. empirically found to be $5 \pm 0,2$. In typical cases this will yield a total measurement uncertainty ϵ_{surf} of roughly 1/1000 of the surface distortion η_{av}.

Chapter 4

Setup

4.1 General

We have built a basic setup, the practical version of what we discussed in the theory. A digital camera is suspended directly above an open water container. The vessel itself is transparent, and below it is the dot pattern, backlit evenly with a matte lamp. We will discuss the fabrication of the pattern at a later point in this section. The camera itself is a Jai CV-M4+CL, which is capable of taking 24 frames per second. For further testing and measurements it would be wise to invest in a camera with a higher FPS, but it allows sufficient precision for our purposes. The CCD of the camera has a resolution of 1380 by 1030 pixels, with each pixel sized 6.45 by 6.45 micrometers. As we have no need for fancy maneuvers with camera zoom or similar things, we simply mounted a fixed focal point prime lens on the camera.

4.2 Optic Considerations

At this point we have only expressed the measurements in the pixels the CCD uses to see and Digiflow uses to interpret the displacements. What we are actually measuring, of course, is an actual surface that is measured vertically and horizontally in millimeters, or other similar distance units. To know the appropriate size of real surface a pixel represents we cannot just measure the size of the basin with a ruler or something similar. It is too impractical to position and or zoom the camera in such a way that the basin 100% overlaps with the image on the CCD, and placing a ruler or something similar in the image to judge by eye is just as impractical as it is inaccurate. So, we use optics to devise a measuring method for which we do not actually need to do that. I do need to preface this by saying that this method only functions if we keep the camera perfectly horizontal, and directly above the surface we are observing. We were already enforcing these two practices, so this does not actually complicate our measurements any further.

Any camera, simply put, is a combination of 1: a lens through which incoming light rays are focused onto 2: An image plane where those images are recorded through for example photosensitive chemicals or a CCD. A schematic image is shown on the next page in figure 4.1. Here you see how the Field Of View or Angle of View of a camera can be deduced from its physical properties. As long as the camera is pointed perfectly perpendicular at the surface, we can use this field of view and its distance to the pattern to determine the size of the field its measuring. These calculations do not require any additional measurements from the ones we already made.

In figure 4.1 one can see that the size of the CCD plate is called d. This size can of course vary from camera to camera. Since the resolution i.e. the amount of pixels of the camera and the pixel's physical size are known, we know how large d is. With it, we can determine the angle β by taking $tan(\beta) = \frac{d/2}{b}$. We do not have b, but we do know the true and tried simple lens equation $\frac{1}{f} = \frac{1}{v} + 1\frac{1}{b}$, where f is the focal length of the lens, v is the distance to the object from whence the light reaches the lens, and b is the distance to the place where a focused image is projected by the lens.The lens equation

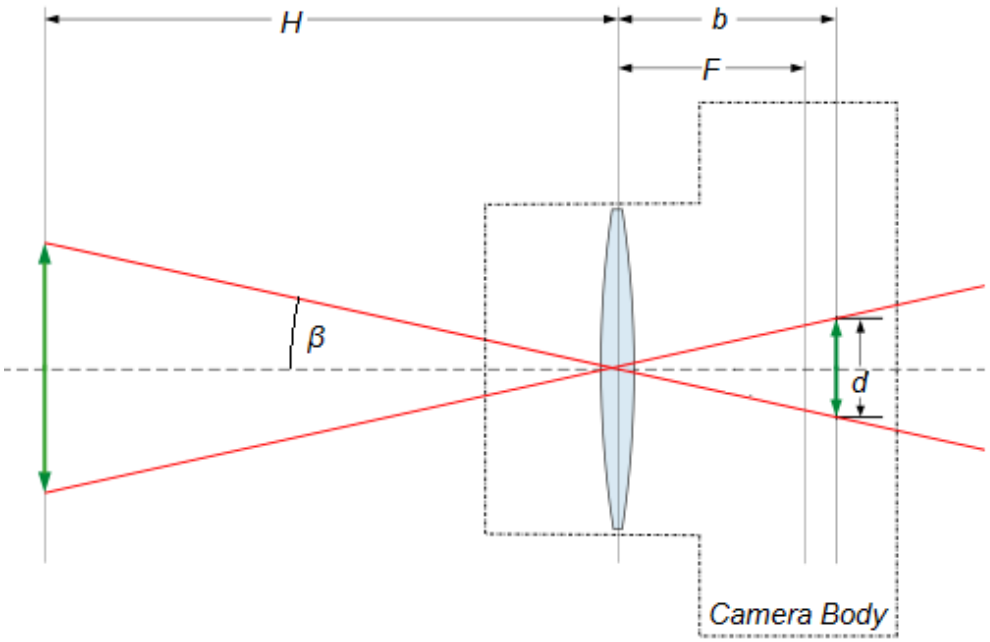


Figure 4.1 – A schematic representation of the camera, its lens, CCD, and the plane it is viewing

can be rewritten to $b = \frac{vf}{v-f}$. Note that this equation determines at what distance from the lens a focused image is formed. An image can also form on another distance but it will be blurry. We will of course make sure this does not happen. We can combine this definition for b with our equation for β to get:

$$\beta = arctan\left(\frac{d/2}{b}\right) = arctan\left(\frac{d}{2vf/(v-f)}\right) = arctan\left(\frac{d(v-f)}{2vf}\right) \tag{4.2.1}$$

This v is defined in our experiment simply as camera-pattern distance H, which results in

$$\beta = arctan\left(\frac{d(H-f)}{2Hf}\right) \tag{4.2.2}$$

The size of the field our camera observes, L, depends on the camera's angle of view and the distance H. Since β is the same on both sides of the lens, we know that the field of view of the camera $L/2$ is simply $L/2 = Htan(\beta)$, so the field the camera can see is exactly

$$L = 2Htan(\beta) = 2Htan(arctan\left(\frac{d}{2Hf/(H-f)}\right) = 2Htan(arctan\left(\frac{d(H-f)}{2Hf}\right)) = d(H/f-1) \tag{4.2.3}$$

The resolution of the camera remains unchanged, so to get the amount of real space that a pixel measures and represents we simply take this L and the resolution of the camera in pixels $L_{x,y}/resolution_{x,y}$. Note that for a CCD that is rectangular but not square, this calculation will start out with different values for the x and y axis but will ultimately result in the same pixelsize.

4.3 The Dotted Pattern

The picture of random dots we use and have so far called a pattern, is of course anything but that. To ensure that no two elements of the image are confused for one another, the

pattern is generated numerically, ensuring no groups of dots are the same. This is done with a program of Moisy's devising [6], "Makebospattern.m", which allows for customization of the size of the pixels and their density. We spend a lot of time staring at these patterns, so we used white dots on a black background, which we find to be easier on the eye than black dots on a white background. This has no effect on performance. This program prints the generated pattern on A4, the composition of the pattern depending on one's preferred amount of dots/A4 and their diameter in mm.
We have previously established several conditions the pattern has to fulfill for our measurements:

1. The density of the pattern must be at least 20 particles per interrogation window (32^2)

2. The dots must have a diameter of roughly 2.5 pixels

The program we are using to generate our pattern has as input the diameter of the dots in millimeters as will appear on the printed page. The size of how they will appear on the CCD of our camera does not concern the program, so we must link the two. The same goes for the image density. The variable the program uses is how many dots one wants to appear on the page, so we must translate the density of particles per interrogation window to the particles per area unit of A4. Reusing the method we used to determine the relation between a pixel and real space, we can determine how large the dots have to be in mm, to become a 2.5 pixel on the CCD. and we can use the same method to determine how many 32^2 interrogation windows fit into an A4 and multiply it by 20 to get the amount of dots we need.

$$x_{pixel} = d(\frac{H}{f} - 1)/N_{pixels} \tag{4.3.1}$$

So for an optimal result, the diameter of the dots must be $2.5 * x_{pixel}$ This also means that the size of the interrogation windows is $(32 * x_{pixel}mm)^2$. A4 paper is 297 by 210 mm^2, which means that to get the necessary density, the printed a4 must have at least $297 * 210/(32 * x_{pixel}mm)^2 * 20$ dots.

4.4 Test

Until this point we have remained in the realm of theory, as we decided on the best course of action for the practice of FS-SS. It is here that we wished to put this plan into practice by showing our Platonic Ideal of a FS-SS analysis, perfectly obeying our own conditions and in general show precisely what this method is capable of.

Thanks to what can only be described as a colossal hardware failure however, this test never came to pass. Instead, we describe an older diagnostic test we had done earlier, in those hopeful days of functional equipment.

In this test, we use the setup we have detailed before, but with a pattern-camera distance of $H = 0.75m$, and water depth of $h_0 = 4.5cm$.The container that held the water was made of 2mm thick transparent PVC plastic that had a refractive index of 1.53. The pattern was affixed to its bottom in such a manner that there was no air between the watervessel and the pattern. This results in an effective pattern-surface distance of $h_p = 45 + \frac{n'}{n_g} h_g = 45 + \frac{1.33}{1.53} 2 = 46.7mm$.

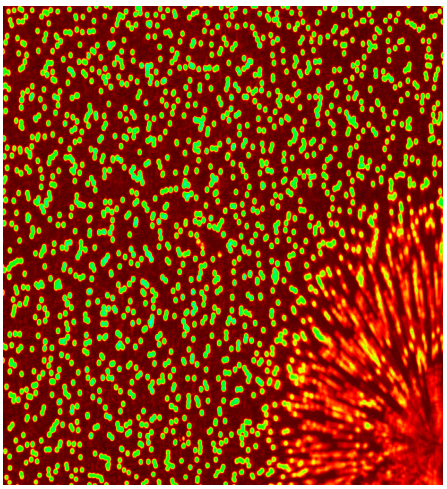
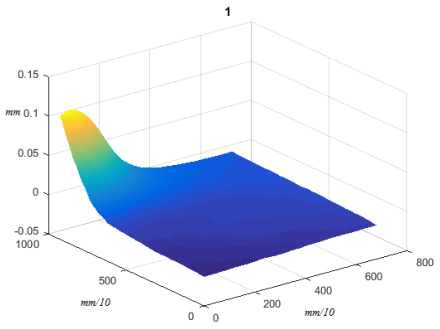


Figure 4.2 – the second frame after the drop hit the surface, before any analysis was done. The streaking of the dots is an effect the software is equipped to handle, and as such presents no problem.

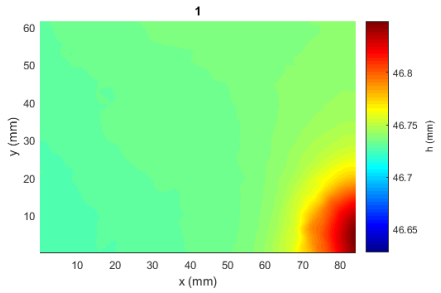
We use this to observe the surface distortion from letting a single droplet of water fall into the water, and measure the surface waves it generates. In a free surface such a droplet would result in perfectly circular waves radiating from the original point where the drop met the surface. In our case, we dropped the droplet near the corner of the water basin, which simply means only a quarter circle is visible. A frame of the camera's recording is shown in Figure 4.2. It is this frame that Digiflow will compare with the reference image and calculate the displacement. Here one can also see the necessity for the equation we detailed in subsection 4.2, as the original image the camera recorded included the edges of the container. We cut that part of the image to not render uninteresting image elements, and conserve memory, but if we had measured the size of the image with a tape-measure or similar instrument we would not have had that freedom nor this level of accuracy.
The Dotpattern we used for this was the kind that Moisy recommends for general use, which is to say, it was an A4 with 50000 dots with a 1mm diameter.

The data we collected was used to reconstruct the surface. The reconstruction results in a scalarfield with the numerical values of the surface height in each point. There are of course several ways of visualising those results, and here we show the most useful. From top to bottom we show the chronological order of recording frames of the surface disruption. The left side shows a threedimensional reconstruction, while the right shows the corresponding image in 2D, showing the surface elevation as colours from a top view rather than a 3d image from an arbitrary vantagepoint. Both have their pros and cons. While the 3d visualization shows very evocatively what shape the surface has, the fact that it shows the surface as an opaque layer means information may be lost from certain perspectives. For example, it is difficult to see the form of a wave trough if the wave crest is quite literally in front of it. The 2D colour coded version gives a universal view of the surface field at the cost of immediate clarity. Of course if we want very precise information from the surface height of a specific location, we would quite simply take the scalarfield and check what value it has on whatever coordinate we wish to know about. The left half shows only the amplitude of the disruption $\eta(x, y)$, while the ones on the

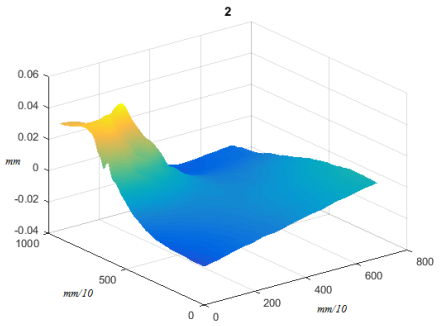
right show the elevation as total surface height $h(x, y)$. This is done purely to show the options that one has.



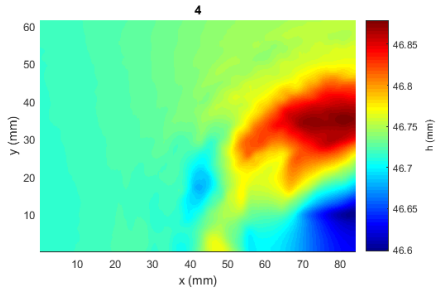
(a) The threedimensional representation of the first frame: The moment the drop hits the water's surface. Note that the horizontal axes are due to some quality in Matlab denoted as 10^{-1}mm, and the verticals as simply mm.



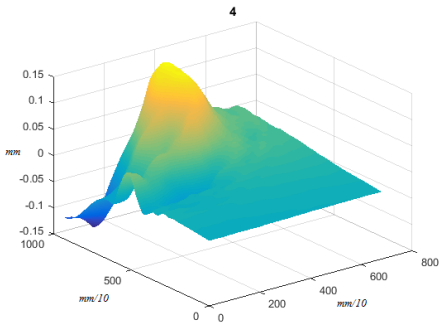
(b) The two dimensional visualisation of the first frame



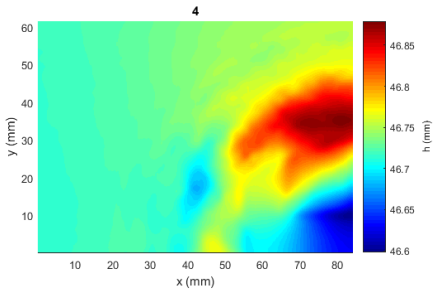
(a) The second frame after impact in three dimensions.Note that the horizontal axes are due to some quality in Matlab denoted as 10^{-1}mm, and the verticals as simply mm.



(b) Second frame after impact



(a) the fourth frame after impact, chosen to show how difficult it is to make out the structure of the trough in three dimensions Note that the horizontal axes are due to some quality in Matlab denoted as 10^{-1}mm, and the verticals as simply mm.



(b) the fourth frame after impact in two dimensions. In this visualisation, no information is lost.

4.5 Test Error and Critical Pattern-Surface Distance

4.5.1 Test Error

In secton 3.4 we established the equation that determines the uncertainty of our reconstruction, eq. 3.4. c is still $\simeq 5$, and the PIV error ϵ is still as small as it always was, 4/100. But since we used a small drop instead of generating a sinusoidal wave, the rest of the values are far smaller than the typical case that we imagined in 3.4. $\eta_{av} \simeq 1mm$, as seen in the measurements, $L = 80mm$, $N \simeq 900^2/32^2 \simeq 600$. λ is a bit trickier to read, as we're dealing with a single pulse. We can guess that at its shortest, that is to say, at the point it will generate the greatest uncertainty, $\lambda \simeq 3cm$. This gives us $\Delta\eta = c\epsilon\frac{L}{N}\frac{\eta_{av}}{\lambda} = 5 * 0.04\frac{80}{600}\frac{1}{30} \simeq 0.001$. This is still roughly equivalent with the typical error, which means that there's a lot of give in that estimate, and the method is a stable one.

4.5.2 Critical Pattern-Surface Distance h_c

We can assume that this wave we generated with the droplet is sinusoidal enough to have a similar critical distance equation as the fictional example we used before. So if we have the equation$h_c = \frac{\lambda^2}{4\pi^2\eta\alpha}$, did we actually steer clear of the h_c? In this case where α is simply (1-1/1.33), λ is 30mm, and η is 1mm, we get
$h_c = \frac{900mm^2}{4\pi^2 1mm(1-1/1.33)} \simeq 9.2cm$. It seems that our measurement obeys our conditions, albeit perfectly by accident.

Chapter 5

Conclusions and Futures

5.1 Conclusion

We have considered the theoretical foundations of this method, noting at each step the necessary assumptions and approximations necessary to have it function. In doing so we decided upon the optimal way of using this technique, and the various decisions necessary to minimize measurement error. We discussed the practice of building a FS-SS set up, and closed off with a test set up of our own, showing that even when we cannot obey our own conditions for maximal accuracy, this method still allows for a near maximal performance. It is very stable, and will hopefully prove useful to NIOZ in the future.

5.2 Discussion and examples of future plans

For the immediate future, we would like to perform the test that eluded us, to ensure the calibration of this instrument meets every condition we have set out to fulfill. Suppose however that we will succeed with this test. Having constructed a functional instrument of measurement, what could NIOZ, and its affiliates use this technique for? We have prepared some examples

Leo R. Maas, in his paper "Wave Attractors: Linear yet Non-Linear explores linear wave attractors which are a phenomenon that occurs in water that is linearly stratified in density. An example of such fluids could be salt water, like in our oceans. Gravity, over time, ensures that its density is greater at the bottom than at the surface, with gradually less dense layers of water making up the distance. The Linear wave attractors occur because of these layers. In a homogenous fluid, surface waves remain just that: surface waves. In these stratified fluids however, the interfaces between the different layers are linked, which means that a surface wave gains a vertical component. If this occurs in a container that has broken symmetry, a three dimensional interference pattern will occur, and can be observed visually. As described in the paper these effects have only been visualized from the side however, making a three-dimensional image of this effect impossible. We suggest that the FS-SS method could be used to show this three-dimensional image that had so far eluded science. Of course, in its current state it assumes the fluid has a constant refractive index, but this is a solvable problem. In

the same vein, Sander van Oers is researching the subglacial oceans of Neptunes moon Enceladus. Various visual cues on the ice suggest that the "volcanic activity beneath its surface is of a Nonlinear Wave Attracting nature. Before sending probes into the void to find out, we might try to simulate those conditions back on earth, and in doing so gain more insight both into lunar volcanic activity and, into the behavior of the Linear Wave Attractors. FS-SS as a non-intrusive imaging method might prove a very valuable asset in this research. An attractive thought is the possibility that FS-SS may be used outside

of the laboratory, or "in the field, as it were. This may once prove possible but we fear that the limiting factors may simply prove too stringent to allow it. If someone was to

try to map a body of water, and successfully established pattern recognition, they still would have the problem of caustics. As most bodies of water are either deep or agitated enough to cause ray crossings and caustics, any information gained in even this best case scenario would be rife with error. An unforeseen development might make it realistic to apply this technique outside of a laboratory environment but for now we have a better idea.

In [12], Sokoray-Varga and Jósza detail their use of PTV to analyze a physical miniature model of the Maros river in Hungary. They built a specific bend in that river, and observed the flow of water, the surface deformations, and the various eddies and vortices that formed. These kinds of irregular chaotic currents could prove very useful to study for the biologists and ecologists at NIOZ, to better understand the factors that play a role in say, the reproductive habits of fish, where they lay their eggs, and why. Or it might provide insight into the kind of current that more efficiently pollutes key areas. The end result of such research would be a better understanding of how the most complex and vulnerable parts of the ocean could better be protected, or indeed recreate such conditions where earlier conservationist efforts have already failed.

Bibliography

[1] F. Moisy, M Rabaud, K Salsac *A synthetic Schlieren method for the measurement of the topography of a liquid interface*, Experiments in Fluids, 13 January 2009.

[2] Image found at, and borrowed from http://fibretechinc.com/tips-maintenance/choosing-the-right-surface-for-your-pool

[3] R. Hooke, *Micrographia*, 1665. Found at https://ebooks.adelaide.edu.au/h/hooke/robert/micrographia/observ58.html

[4] C. R. Nave,*Lens Maker's Formula*, Hyperphysics, 2012. Last visited on 14-6-2016 at http://hyperphysics.phy-astr.gsu.edu/hbase/geoopt/lenmak.html

[5] Weisstein, Eric W. *Radius of Curvature*, MathWorld–A Wolfram Web Resource http://mathworld.wolfram.com/RadiusofCurvature.html

[6] F. Moisy, M. Rabaud, *Free-Surface Synthetic Schlieren: A Tutorial*, 10-9-2010, last visited on 14-6-16 at http://www.fast.u-psud.fr/~moisy/sgbos/tutorial.php

[7] F. Moisy, the Matlab program Surfheight.m, to be found at http://www.fast.u-psud.fr/~moisy/sgbos/tutorial.php

[8] S. B. Dalziel & Research Partners, *Digiflow User Guide, version 3.4*, June 2012, research.com/digiflow/

[9] M. Raffel, C. Willert, S. Wereley, J. Kompenhans, *Particle Image Velocimetry: A Practical Guide*, 2007

[10] Above book, pg.170

[11] Diagram from the above book, pg. 173.

[12] B. Sokoray-Varga, J. Jósza, *Particle tracking velocimetry (PTV) and its application to analyse free surface flows in laboratory scale models*, Civil Engineering 52/2 (2008) 6371, web:http://www.pp.bme.hu/ci

c Periodica Polytechnica 2008