University of Groningen
Faculty of Science and Engineering
Industrial Engineering and Management

Bachelor Integration Project semester 1 (TBIP15E.2018-2019.1)

Validating the Bladder Storage System proof-of-concept of Ocean Grazer 3.0 using principles of photogrammetry

By C.B. van den Hoek
S2734036
Date: January 2019

Supervisors:
First supervisor: prof. dr. A.I. Vakis
Second supervisor: dr. ir. A.A. Geertsema

Abstract

Worldwide the demand for sustainable energy is increasing. To balance the production capacity and the demand, electrical energy needs to be stored for a period of time. One of the potential solutions is the storage system of Ocean Grazer 3.0. This machine will be positioned at the bottom of the ocean and transforms wave and wind energy into potential energy which can be stored and finally be transformed into electrical energy.

The Ocean Grazer 3.0 is a novel concept which pumps water from a rigid reservoir into a flexible 'bladder' reservoir, storing potential energy which can be harvested using hydrostatic pressure of ocean water to send a water flow through a generator, from which electricity will be generated.

A proof-of-concept of the Ocean Grazer 3.0 has been built, in this integration project it is investigated using principles of photogrammetry, to get a 2D model of the movement of the bladder. This model is compared to an existing 2D COMSOL model, with this data the 2D COMSOL model can be improved in the future. This is needed to find more efficient designs and to upscale to project.

Keywords: Photogrammetry, Geometry, Bladder Reservoir, Validation, Experimenting, Simulation

Table of Contents

Nomenclature

Abbreviations and Acronyms

AAE	Adobe After Effects
AOV	Angle of view
BSS	Bladder Storage System
DTPA	Discrete Technology and Production Automation
EPDM	Ethylene propylene diene monomer rubber
FHS	Free-hand stereo
FOV	Field of view
fps	Frames per second
IEM	Industrial Engineering and Management
OG	Ocean Grazer
OGG	Ocean Grazer Group
POC	Proof-of-Concept
PVC	Polyvinyl chloride
SSS	Simple stereo system
UG	University of Groningen
UW	Under water

List of Variables

a	Angle of view
alpha	Angle in FHS of left camera
B	Width between cameras
beta	Angle in FHS of right camera
dB	Width from angle beta to Z in FHS
F	Focal length/point (millimetres)
gamma	AOV
nw	Refractive index
t	Time
V	Volume (litres)
X	Horizontal distance
Y	Vertical distance or height
Z	Depth

1 Introduction

According to the BP Statistical Review of World Energy, every year the global energy consumption is growing (Figure 1) [1] and non-renewable energy sources are declining. Significant progress has been made in the field of offshore electricity generation. In 2017, the Danish, German and Dutch power-grid operators signed a trilateral agreement for the development of a large renewable European electricity system in the North Sea called 'North Sea Wind Power Hub'. It has the potential to supply 70 to 100 million Europeans with renewable energy by 2050 [2]. Now, new energy storage solutions are demanded. One potential solution is the Ocean Grazer.

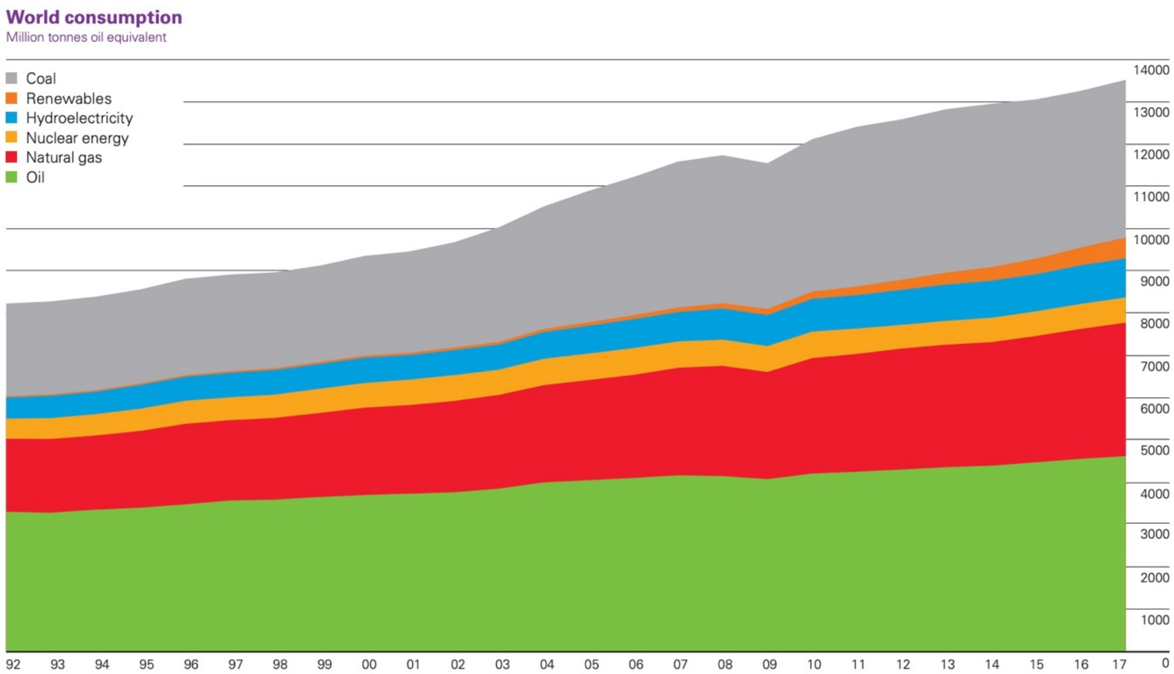

Figure 1: World energy consumption in million tonnes of oil, from 1992 to 2017 [1]

The University of Groningen has been developing a device to capture wave energy since 2012 under the name 'Ocean Grazer'. According to Prins et al. [3], OG is a new offshore renewable energy harvesting concept. It combines wave energy converter technologies with on-site energy storage and a modular design which can also include wind turbines, all to generate and store renewable energy offshore. The latest concept (version 3.0) of OG is illustrated in Figure 2. One can observe a floater blanket system, a pumping system, and a storage reservoir system. These three systems in combination with turbines make the harvesting of wave energy and storage possible. A more detailed explanation of OG will be given in the system description section (2.4).

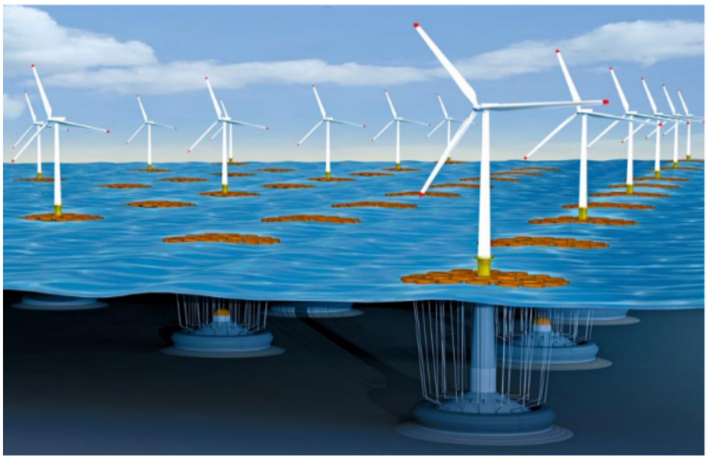

Figure 2: Overview of Ocean Grazer 3.0 [4]

1.1. Research and design topic

A key system of OG is the storage reservoir system, it enables OG to store energy on-site. The reservoir consists of a flexible bladder and a rigid reservoir both placed on the seabed. If the flexible bladder is charged, it stores energy by utilizing the hydrostatic pressure of the surrounding ocean. Since the bladder is charged with an internal fluid and not with ocean water, it must be able to separate ocean water from the internal working fluid. Hence, the bladder must be highly impermeable to water as well as ocean water.

In 2018 a proof-of-concept was created following a master thesis by J.A. Koning [5]. The difference between a proof-of-concept and a prototype, according to Skelia.com [6], a software development company: *“A proof-of-concept is a small project created to test whether a certain idea or theory about the product can be implemented. For example, when you don’t know if a feature can be built, you test the idea’s feasibility by creating a POC.*

Similarly, to a POC, a prototype’s main purpose is to help you make decisions about the product development and reduce the number of mistakes. But it does differ. While a POC offers you a model of just one product’s aspect, a prototype is a working model of several aspects of the product. The development team usually uses prototyping to discover errors in the system. By building a prototype, they test the product’s design, usability and often functionality. With a proof-of-concept, you don’t get to do all of that because it’s smaller and can verify only a single issue.”

This POC is created at such scale that it will fit in the water tank located at the Nijenborgh building at the University of Groningen.

This report aims to contribute to the validation of a 2D COMSOL model created by bachelor student Sietse van den Elzen [7]. COMSOL is a computer program containing possibilities to do multiphysics simulation. Experiments using multiple cameras will be done to the POC, in particular, the displacement of the bladder-prototype will be investigated. These values can help to optimize the 2D COMSOL model, which is needed to test other designs to increase efficiency and for upscaling of the concept.

2 Problem Analysis

2.1 Problem context

To store the wave energy harvested by OG's floater blanket system, the Ocean Grazer Group initiated the idea to create overpressure in a flexible underwater reservoir by pumping an internal working fluid from the rigid reservoir into the flexible bladder [8]. Hence, energy is stored because the hydrostatic pressure of the surrounding ocean makes sure the flexible underwater reservoir can be discharged anytime. When the bladder is discharged, the internal working fluid runs through turbines back into the rigid reservoir. Subsequently, energy is generated.

2.2 Problem owner analysis

There is one problem owner, namely the Ocean Grazer Group. It is a research group consisting of a Chief Executive Officer (Frits Bliek), Chief Technical Officer (Marijn van Rooij), an advisory board of 2 professors (A.I. Vakis & B. Jayawardhana), Postdoc, PhD, Master- and Bachelor students, who are developing and researching the preliminary design of OG. The problem that the OGG faces is that there is currently no thoroughly investigated design for the flexible underwater reservoir. In order to design an applicable flexible reservoir, knowledge about the bladder's movement during all stages of the charging and discharging cycle is required. Moreover, the OGG wants to gain more insights on how the POC performs in terms of efficiency of the total water in the system compared to the water that actually flows through the turbines.

2.3 Stakeholder analysis

A stakeholder can be defined as a person who or entity that has a stake in the result of a project. This project has one stakeholder; the Ocean Grazer Group.

The OGG is a stakeholder since the members of this group are researching distinctive parts of the preliminary design of OG. Since all distinctive parts of OG's preliminary design are interrelated, the entire OGG is a stakeholder of this bachelor thesis. The results of this bachelor thesis might be used to show investors the capabilities of OG, which will result in more funding, which is needed to further develop OG.

2.4 System description

The latest iteration of the OG system consists of three subsystems: a floater blanket, a pumping system, and a storage reservoir system. The latter is of concern to this project. The storage reservoir system is depicted in Figure 3. It consists of a rigid reservoir and a flexible underwater reservoir; the bladder.

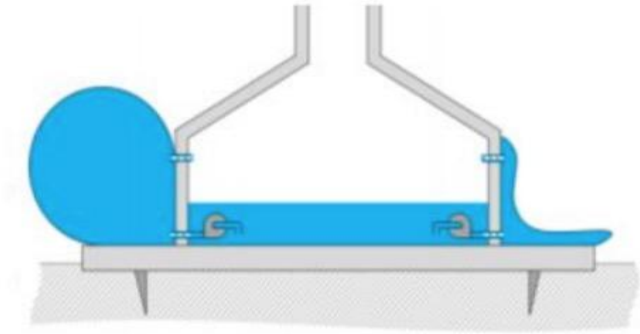

Figure 3: Storage reservoir system in Ocean Grazer 3.0 [3]

A brief explanation is given in order to grasp the concept of the OG as a whole. The inputs of the system are the movements of waves. Vertical movements are used by the pumping system to transport water from the rigid reservoir to the bladder reservoir. The water flow input to the bladder should be measured to make sure that the maximum volume capacity is not exceeded, and a regulated water stream should be the output from the bladder to the electrical generator. The internal working fluid is assumed to be water and the bladder is assumed to be made of EPDM rubber. Discharging happens as a result of the pressure difference between the flexible and the rigid reservoir. As a matter of fact, the pressure inside the flexible bladder is equal to the hydrostatic pressure. Since pressure inside the rigid reservoir is equal to the atmospheric pressure, overpressure which is relative to the rigid reservoir is created inside the bladder when the bladder is charged. This overpressure is used to store potential energy in the following way. The hydrostatic pressure of the surrounding ocean makes sure the flexible underwater reservoir can be discharged anytime. When the bladder is discharged, the internal working fluid runs through turbines back into the rigid reservoir. Consequently, energy is generated.

2.5 Research goal and scope

This project aims to do experiments that require skills that have been obtained in the IEM bachelor curriculum. Interviews were held with my supervisor Mr. Vakis and the CTO of the company; Marijn van Rooij. Much information has been obtained in these interviews and in feedback sessions, it was clear that there are many problems to resolve. Then, a clear overview of the stakes was made. These goals were devised to be in line with the goals of the stakeholder. These goals are put in a list and priorities are given in the following table:

Goal	**Priority**
Wanting to create a total model of the working of the Ocean Grazer	Highest
Wanting to upscale the proof-of-concept to get a better view of how the bladder behaves in the final size	High
Wanting more funding, so more investors	High
Wanting to increase the Technology Readiness Level to get closer to the actual Ocean Grazer being used at sea	High

Wanting to know the difference between the COMSOL model and reality	High
Not wanting any folds in the design, which will decrease the lifetime of the bladder, which decreases its performance.	Medium
Wanting to know what effect the sea-life has on the durability of the bladder, in terms of the effect of salt water but also on dirt and algae forming on the outside of the bladder	Medium

Table 1: Goals of the company

The last wish is outside my scope; however, my thesis does contribute to the rest of the goals in some way.

My goal is to create a graph which represents the actual data of the displacement of the bladder in the POC. This graph and data which can in a later stage be used to validate the 2D COMSOL model in terms of movements of the bladder. The POC will be studied using 2 cameras, from which the displacement can be calculated. It is not possible to calculate this using 1 camera, because it is not possible to place a camera perpendicular to the bladder. This is because the POC is placed inside the water tank, and it lies just below the see-through glass part; see the limitations section. Two cameras will be needed to find the 2-dimensional movement of the bladder. Finally, a 2-dimensional graph of the cross-section of the bladder will be created. How this is done will be explained in the methods section (3.2).

This goal will open doors to further improve the COMSOL 2D model. One could say that it can help with the validation of the COMSOL model. This needs to be done, because with an accurate COMSOL model, new designs can be made which could reach higher performances, for example an important requirement is that the design does not fold, which it does do in this design. Folds damage the material, which shorten its lifetime, which will be costly to replace often. One could say that the performance is decreased when the design contains folds. These designs are needed to reach higher Technology Readiness Levels. These levels are explained in figure 4.

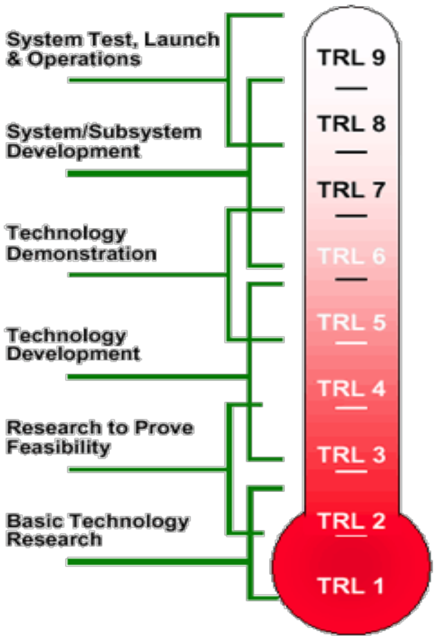

Figure 4: Technology Readiness Levels [9]

A complete model must be created in the future, because the company wants to have a total model of the whole Ocean Grazer. It is not yet clear in which program this will be done, Marijn does not think it will be COMSOL.

2.6 Research problem

The research problem this bachelor thesis focuses on is previously described in the problem analysis. The problem owner is the OGG and its problem is that there is currently no thoroughly investigated flexible underwater reservoir for OG. Therefore, the research problem can be defined as follows: 'There is not enough data collected to validate the flexible underwater reservoir of the Ocean Grazer.'

2.7 Research question

The research problem stated in the previous section leads to the following research question:

"How does the proof-of-concept of the Bladder Storage System behave?"

The goal is to create a 2D graph of 3D camera video of the movements of the bladder. This can be used to validate the 2D COMSOL model in the future. Sub-questions which are in line with this goal are:

"How does one calculate 2D displacement using 3D data, taking into account the changing optics of having a proof-of-concept which is located under water?"
"How does one create a graph containing points which change over time?"

3 Research Design

3.1 Design steps

The research can be related to two cycles of Hevner; see figure 5[10]. My project will be focused primarily on the Rigor Cycle, as it is about testing and making a research set-up. The output can be seen as knowledge. This will be fed back into the Design Cycle, which will be done by the OGG, who will use the information to further develop the design of the BSS.

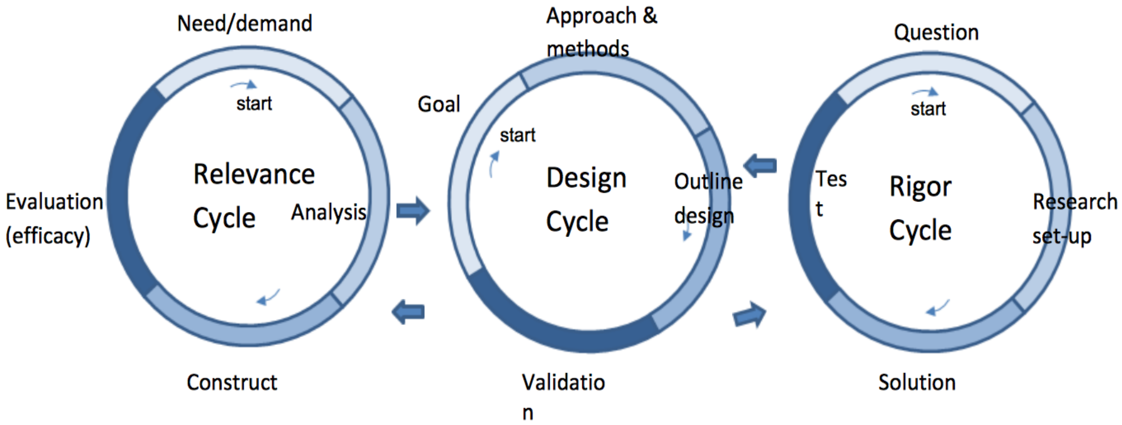

Figure 5: Cycles of Hevner [10]

3.2 Methods

To answer the previously formulated research questions the following research methods are used. Firstly, a literature review is done in order to find useful information about previous research that has been done by the OGG. This information will give some expectations of what the results of a bachelor thesis should look like, it helped to decide on the amount of research that needs to be done and what can be done.

Then, the research problem had to be obtained. Marijn clearly showed the capabilities of the POC. The moments of opening and closing of valves as well as the starting and stopping of the pumping from the rigid to the flexible reservoir can be clearly tracked.

Information about possible cameras was obtained. For normal motion capture activities, usually depth sensing cameras are used. These cameras consist of an infrared transmitter and receiver to calculate the distance to a certain object, which can give a value to each pixel, generating a 3D image. Resentfully, using infrared does not work in an underwater setting. A paper has been found that experimented with using commercial depth sensing cameras to do the same experiments [11], however, the goal of this thesis is to use normal cameras (see limitations section).

UW depth measurements are usually done by pressure sensors for robots and other UW devices. For this POC, just 2 normal cameras will be used.

Dots will be placed on the bladder, which can then be tracked using Adobe After Effects (Matlab is also an option). An example is shown in figures 6a and b from the Ocean Grazer Introduction video [12].

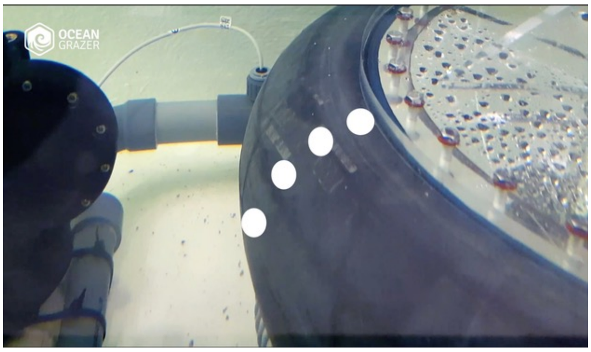

Figure 6a: BSS inflated

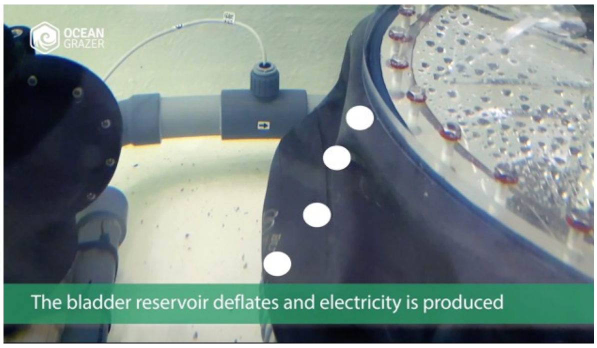

Figure 6b: BSS deflated

The stickers will be attached in a straight line, with a clearly measured space in between the stickers. Using principles from trigonometry and optics, a model will be created to calculate the x and y positions of the dots on the bladder, which will result in a graph of the cross-section of the bladder-reservoir. A rough sketch of the final graph is shown in figure 7a, which should be able to be compared to the current output of the 2D COMSOL model, as shown in figure 7b.

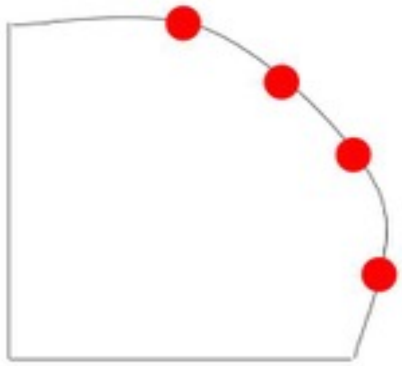

Figure 7a: Rough sketch of final result

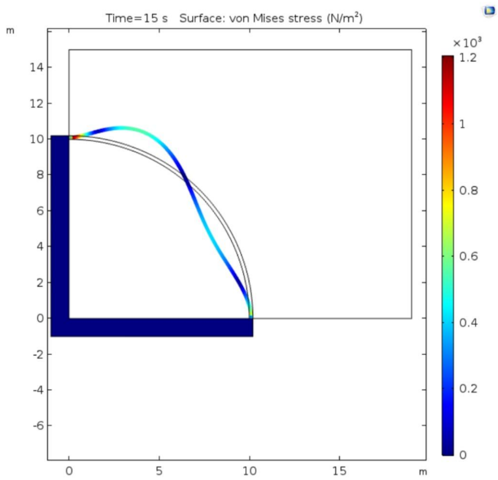

Figure 7b: COMSOL model result

3.3 Resources needed and risks

A working POC is needed. It turned out the POC was leaking, however, this could easily be resolved. Leakage made the POC unable to deflate. Water had to be drained out of the system using a tube filled with water and emptying it in a bucket which was located at a lower point compared to the inside of the water. Water would start flowing using Bernoulli's Law. If too much water got out of the rigid bladder, which would not be a problem, new water would be added using a hose connected to a tap.

Other resources needed to obtain an answer to the research question are 2 working UW cameras. In this project a Nikon Coolpix AW100 and a GoPro Hero5 Session will be used (see section 4.2). After getting into contact with Dr. Eize J. Stamhuis [13] it was decided that a GoPro would be able to do this job.

To process the data; a mix of Matlab, Excel and Adobe After Effect will be used. A fast computer is needed to process AAE and Matlab data.

4 Literature study

For some background information about Ocean Grazer multiple theses were studied; of Sietse den Elzen [7], Ton Koning [5] and an interview and the thesis of Lennard Hut [14] has been studied to discover what the IEM bachelor thesis entails.

4.1 Photogrammetry

The main thing to research is ways to calculate a 3D point out of multiple 2D images. This principle is called photogrammetry: the use of photography in surveying and mapping to ascertain measurements between objects. Two ways of tackling this problem were investigated.

Important to remember, when speaking of X, Y and Z; X is the width of the image, Y is the height of the image, and Z is the distance from the camera to the object.

4.1.1 Free hand stereo

Firstly, a way where 2 cameras are looking at a set of points from 2 different directions was considered. According to literature; this method is called 'free hand stereo'; FHS. 1 experiment using this setup has been made; see section 5.2.1.

Using a free-hand stereo system the following formula can be used:

$$z = \frac{B\ \tan(\alpha)\tan(\beta)}{\tan(\alpha) + \tan(\beta)} \quad (1)$$

Below a schematic view of the setup including the variables is shown:

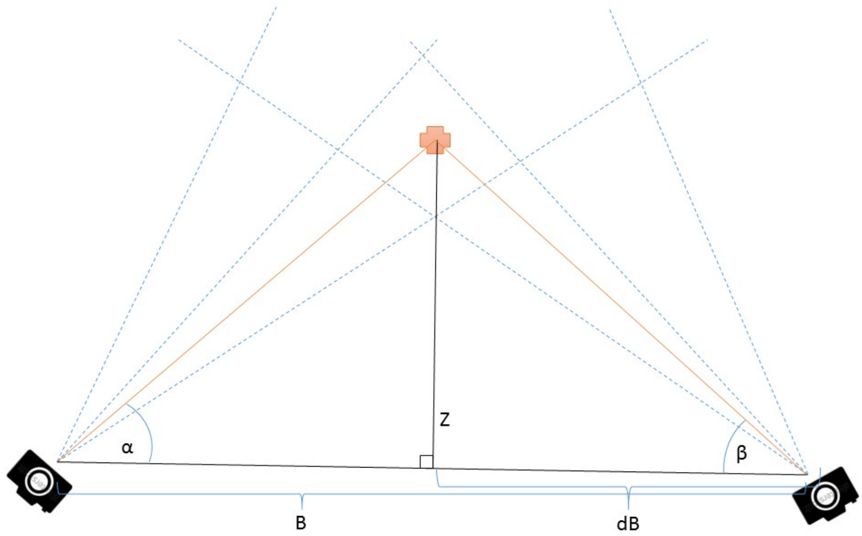

Figure 8: The freehand stereo method

To calculate alpha and beta, the exact angle between line B and the centre focus point of the camera must be known. By knowing the AOV and knowing how many pixels there are in your image, the angle between the centre of the image and the point on the bladder can be calculated. Along these lines, the angle should be subtracted from the angle between A and the centre point to get alpha and beta.

4.1.2 Simple stereo

A common method for extracting depth information from intensity images is to acquire a pair of images using two cameras displaced from each other by a known distance. Disparity refers to the difference in location of an object in corresponding images (left and right) as seen by the left and right camera. In a pair of images derived from stereo cameras, the apparent motion in pixels for every point can be measured and an intensity image is made out of the measurements. A disparity map refers to the apparent pixel difference or motion between a pair of stereo images [15]. This method is also known as a ‘simple stereo system’; SSS.

Lecture slides of a course in Computer Vision from the Paul G. Allen School of Computer Science and Engineering of the University of Washington have been found to grasp the concept of triangulation [16]. Triangulation is the tracing and measurement of a series or network of triangles in order to determine the distances and relative positions of points spread over an area, especially by measuring the length of one side of each triangle and deducing its angles and the length of the other two sides by observation from this baseline. In this formula, the so called ‘focal point’, which is located at the focal length ‘f’, of the camera must be known. The definition of a focal point is the point at which rays or waves meet after reflection or refraction, or the point from which diverging rays or waves appear to proceed. When a camera is focussed at infinity, as in this instance, the picture will be depicted on the focal length.

Using this information, a formula has been derived for calculating distances for this problem (formula 2). Deriving this function is done in Appendix 2. See Figure 9 for an overview of the measurements.

$$z = \frac{Bf}{xl + xr} \tag{2}$$

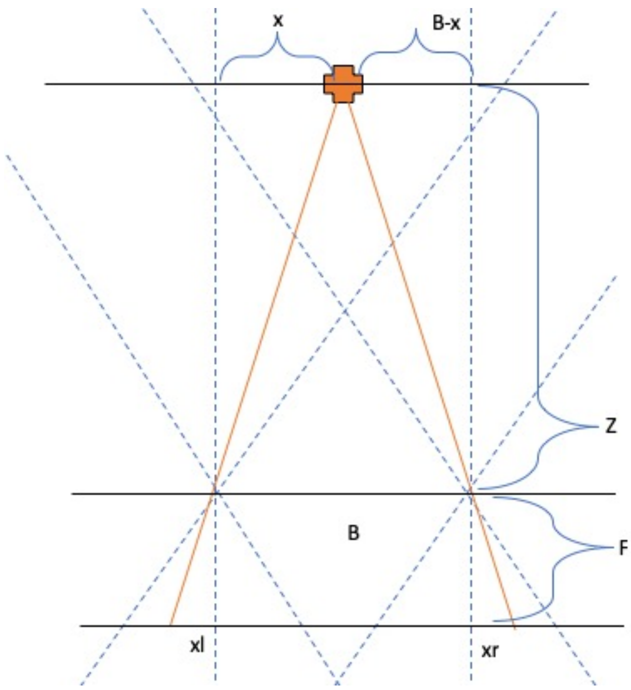

Figure 9: Disparity map of a Simple Stereo Method

Z will be the distance from a line of where both cameras are placed. In a camera image, you are able to find the exact location of pixels of a point which you are trying to track. The transition from pixels to a value of millimetres had to be made. To calculate this, the AOV must be known. A formula to calculate this is given in section 4.1.3.

To get Xl and Xr out of the x pixels of the frame produced by After Effects, formula 3 was used:

$$xl = \tan\left(\frac{\gamma}{2}\right) f \frac{x}{960} \tag{3}$$

Where xl=xr depending on the x of which camera you are using. Gamma is the underwater AOV. This is divided by 2 because we are looking at the distance between the centre of the image and the marker. This is also the reason why 960 is used, it is the total amount of x pixels divided by 2. F is the focal length; 28mm in this instance (see section 4.2).

4.1.3 Angle of view

AOV is calculated as follows [17]:

$$diagonal\ AOV = 2\arcsin\left(\frac{\sin\left(\frac{a}{2}\right)}{nw}\right) \tag{4}$$

Where *a* is the coverage in air (75,38 degrees) and *nw* is the refractive index. In table 2, different angles are given for certain refractive indices.

	Pixels	Angle in degrees, Open air	Angle in degrees, UW (salt)	Angle in degrees, UW
horizontal	1920.00	67.91	48.20	48.56
vertical	1080.00	41.49	28.25	28.48
diagonal	2202.91	75.38	54.34	54.73
Refractive index		1	1.339	1.33

Table 2: Angle of view with different refractive indices

Because the calculations are done at an angle of 45 degrees, some trigonometry has to be applied. The distance in Z direction of the camera only changes by the tilting of the cameras. The height change is measured. The tilt of 45 degrees is also exactly measured, now you can use this to calculate Z while only Z' is known, where Z' is the straight distance from the camera to a marker.

For Z, everything is multiplied by the sine of 45 degrees, for Y the formula is divided by the cosine of 45 degrees (because the sine and cosine functions are in radians in Matlab, 45 degrees must by multiplied by pi and divided by 180 degrees.)

The cameras only change in Y and Z direction; X stays the same for the cameras and the markers. A tangent function is not needed.

4.2 Cameras

The focal point depends on which cameras are used. For this project, the cameras I used were a Nikon and a GoPro, as shown in figure 10 a and b:

Figure 10a: Nikon Coolpix AW100

Figure 10b: GoPro Hero 5 Session

Resolution, focal length and the number of frames per second are equal for both cameras, namely 1080p, 28mm and 30fps [18] [19].

The GoPro has a fish-eye lens, it is an action camera, so it is made for wide shots. There is a setting which convert the picture to a narrow view, namely a view with a focal length of 28mm.

At the Discrete Technology and Production Automation (DTPA) research group at the UG, more research is done using cameras. To find characteristics of cameras, Apriltags are used in ROS [20]. They are handy for checking statistics of the cameras. However, this method only works for USB cameras, it would have taken too much time to rewrite a script to make Apriltags work for the used cameras. This subject will be discussed in the limitations section.

Other terminologies have been found which also have to be taken into account. Epipolar geometry is the geometry of stereo vision. When two cameras view a 3D scene from two distinct positions there are a number of geometric relations between the 3D points and their projections onto the 2D images that lead to constraints between the image points. This is shown in Figure 11.

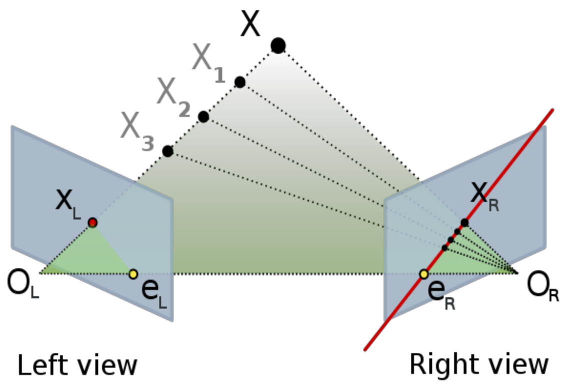

Figure 11: Epipolar geometry [21]

The exact measurements of the proof-of-concept had to be known, these are stated in the master thesis of Ton Konings [5].

The output of the data which COMSOL is showing a plot with x and y axis in centimetres, the output of the Matlab script must look similar.

4.3 Video tracking

There are multiple ways to track certain points in a video. First of all, clear markers must be put in place so software can easily track certain points. This can be done in 2 ways.

In Matlab the images of the video can be binarized, where only a black and white picture will appear. Using the function 'centre of mass', the positions of the dots can be shown. This is treated in the course Mechatronics at UG.

Adobe After Effects can track an object in an easier way, although it must be able to create a way where multiple videos can be analysed as quick as possible. Adobe After Effects is able to give the location of the pixels of a certain moving part, which must then be calculated into a real-life distance. This will then be done using all the distances and angles calculated before the experiments, and useful camera data as the angle of the lens.

5 Experiments and building a setup

5.1 Adhesives on bladder

The markers should not damage the bladder and they should also not hamper the movements of the bladder. This means that the markers should be small, also because the markers will be UW, they should be made of a waterproof material. Also, the colour must be white, which will make it easier to find the markers on the grey bladder in Matlab or AAE.

Before the first experiment inside the wave tank, a way to attach some kind of tape to the bladder must be found. A setup was created where 5 different adhesives where put on a piece of EPDM rubber, which had been used in an old setup. This piece of rubber was placed UW for 70 hours; the result was that electrical tape stays stuck UW. It is also easily deformable and easy to remove afterwards, so it will be used in this experiment. In Figure 12a the setup is shown, in 12b the results are shown.

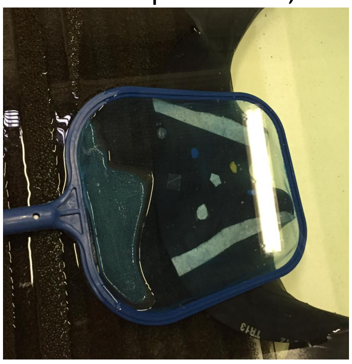

Figure 12a: Adhesives UW

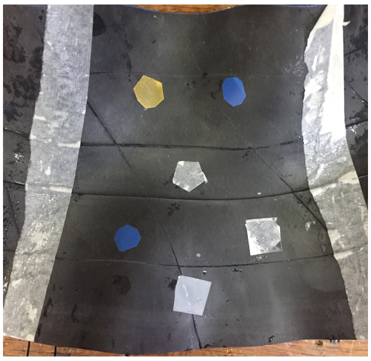

Figure 12b: Results of experiment

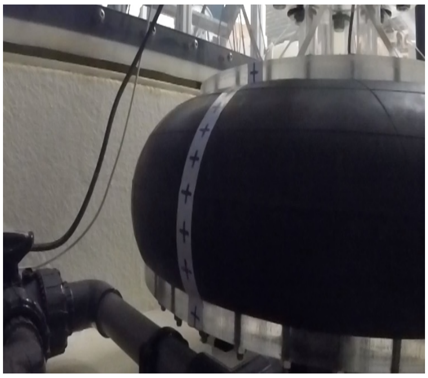

Figure 12c: Final markers on bladder

Instead of placing dots on the bladder, there was decided that a strip can be placed on the bladder with crosses drawn on them. To draw crosses is easier compared to cutting small pieces out of electrical tape and to attach them in an accurate line on the bladder. The final result looks as shown in Figure 12c.

Also, a marker was placed on top and at the bottom of the bladder, as an indication point.

5.2 Camera setups

5.2.1 First camera setup

The first idea was to use a camera outside of the wave tank to shoot video from another angle, however, after the wave tank was filled the camera could not show all the markers clearly. This camera was a Panasonic DMC-FZ200. The picture looked as follows; a shadow can be seen. This can only be solved by turning the lights off, although this would make the bladder itself hard to see as well. Instead of the camera, a colleague from the OGG brought an UW camera in that will be able to create a video from the other side of the bladder. In Figure 13 the image of the bladder recorded from outside the wave tank can be seen.

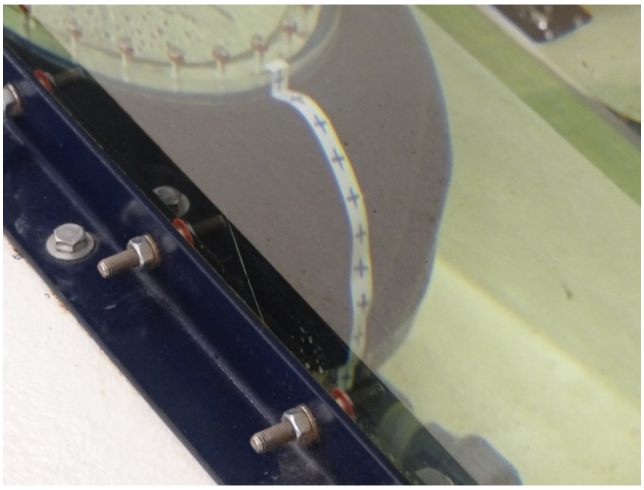

Figure 13: Best view using a camera in open air

Then, a camera setup for inside the wave tank was made. Out of PVC pipes a construction was made to hold the UW camera (a GoPro; at this time the Nikon was not acquired; see limitations section) in place. This side was the right side because there was no place on the bottom of the wave tank on the left side because of the pipes. Wistfully, because of refraction UW, the image of the UW camera was a bit too close so not all the markers are in the field of sight. This must be taken into account in the next setup. The setup is shown in figure 14a and a screenshot of the UW video is shown in figure 14b.

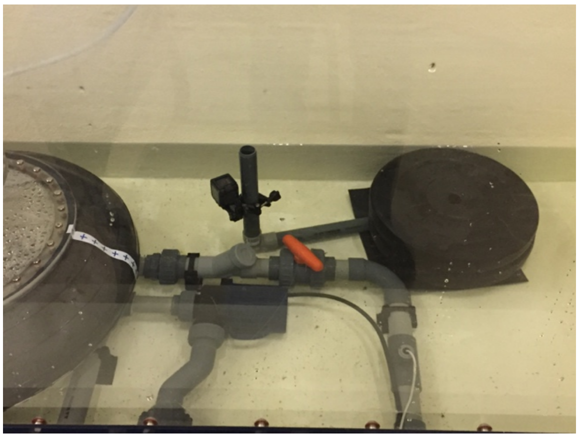

Figure 14a: First setup

Figure 14b: Image from first setup

To get the camera out of the wave tank using this setup, the water tank had to be completely drained. This filling and emptying takes much time, a couple of hours each, and much water is spilled. See limitations section.

Using the one video where 3 points are visible on the bladder, the data was analysed. At first, AAE was used to track the 3 crosses on the bladder. The UG online environment was used, the calculations took a substantially long time, namely about 4,5 hours for 3,5 minutes of following 1 marker, because it takes around 2,5 seconds to calculate 1 frame. It showed that AAE must be run on a fast computer in the future.

With the pixel data, the X axis was calculated (Y axis stays the same) of a camera that would be virtually on the other side of the wave tank. This is just for testing purposes. With this data I could then apply the formula from section 4.1.1.

5.2.2 Second camera setup, hanging

A new idea was considered; to create a hanging structure to film the bladder and its movements. Thusly, the water level inside the wave tank can be kept the same. Using two tight wooden parts, the PVC pipe can be kept in place, and by moving it around, it can be lowered into the water. A simple Solidworks sketch with measurements of where the cameras lie in space is shown in figure 15.

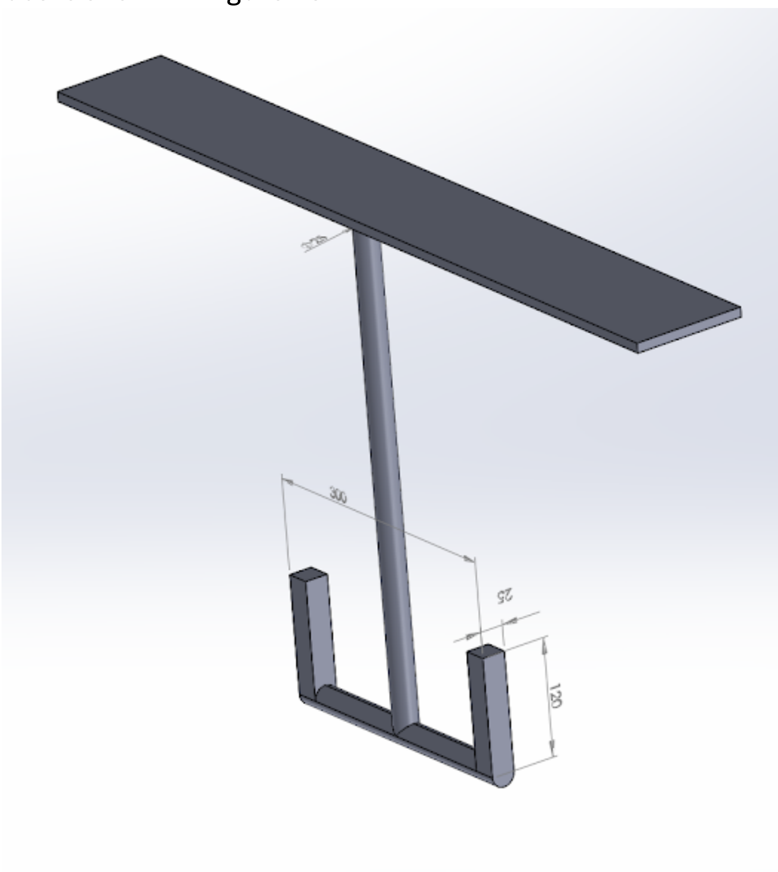

Figure 15: The hanging setup; with a camera width of 300mm, the centres of the camera lenses are located 120mm off the bottom of the device, length under angle is taken into account

Because the GoPro has a different attachment method compared to the Nikon, a bent part had to be used in the PVC construction. If the GoPro was placed vertically, the centre of the camera would not be the same height as the Nikon.

The Nikon was put on a tripod head attachment device, which a ring was added to, to connect to the PVC. The tripod head attachment device allowed the Nikon to be decoupled easily. This is shown in figures 16a and 16b.

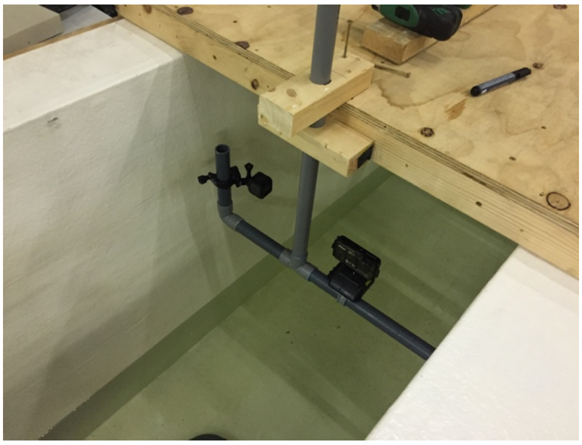

Figure 16a: The second setup, without adjustments

Figure 16b: Close up view of second setup

5.2.3 Later adjustments

Firstly, a horizontal setup was created. Unfortunately, not all the markers would be in the display. The cameras would have to be moved up and be tilted, so the top of the bladder would also come in range. This required a change in the design; the Nikon camera was already able to tilt. However, this is impossible for the GoPro. A piece is placed on the PVC pipe to the GoPro which allows the GoPro to disconnect from the setup, it also allows the GoPro to rotate. The same decoupling unit was used at the T-division of the setup. This allows the whole unit to disconnect, so it is easier to change angles or to test to structure above the water. This unit is shown in figure 17a.

A floodlight is added to the structure to give a clear picture for both cameras, especially for the Nikon because without lighting; the image would look extremely green. Another reason is so it can be used as a clear point of recognition to set the frames equal to each other. The frame number of when you only see a little change in colour (the cameras cannot record this whole change at once) can be noted and subtracted from your first measuring point. For the GoPro this number of frames is extremely large because it uses a date annotation. When subtracting both large numbers a small difference will be the result, it is the difference in frames of when which camera was started. By logical thinking you can then set the points equal to each other so the frames match. This floodlight is shown in figure 17b.

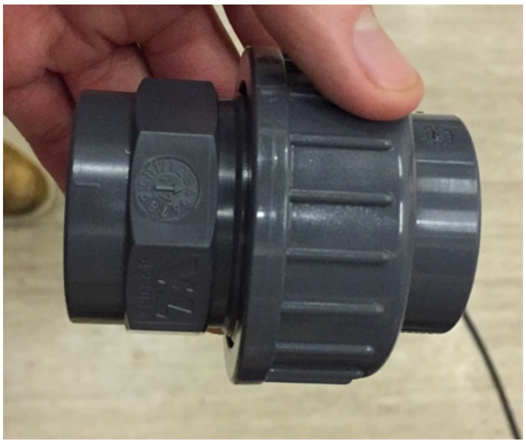

Figure 17a: Decoupling unit

Figure 17b: Floodlight

5.3 Calibration device

5.3.1 Introduction

The setup must be calibrated to know if the values measured in the formulas are correct.

Firstly, experiments were done in open air to validate the formulas. Tests I did were:

- Moving a cross in front of the camera to discover if the points are tracked in the right way.
- Moving a box with markers on it on a table, at exact positions so 2 values can be matched

The first test showed that the concept worked, however the accuracy could not be shown. The second test contributed to the fact that when a marker is recorded close to the border of your FOV; the depth measurement will become less accurate.

5.3.2 First device

Figure 18: First calibration device

In figure 18 the first calibration device is shown. The idea for this calibration device is that is can be recorded both in open air and UW, because it is hanging from 4 strings. The strings are of such a length that the piece of wood would be in a 30-degree angle, if the top piece of wood is held exactly horizontal. This was checked using a water level attached on the top. Markers were placed on the lower piece of wood with exact distances in between each marker, these would be my reference points to do the calibration.

However, this device did not work. The wood would float in the water, even after increasing the weight of the wood such that it would be difficult to carry the structure. Another calibration device had to be created.

5.3.3 Final device

In the previous experiment it was learned that the calibration device had to be heavy. An aluminium plat was found that could be used and a sketch was made to create a device that sits over the bladder (see figure 19a). The aluminium was cut in exact sizes and holes were cut the put screws in to attach the plates to each other and to attach the legs. Finally, small pieces of rubber were added to the legs and to the point where the device sits on the bladder.

The calibration device was placed on top of the bladder, and using a water level, the legs were set to the exact height of where the calibration device would be in a horizontal position (figure 19b). The last step was to add tape with markers at exact points on the calibration device (figure 19c).

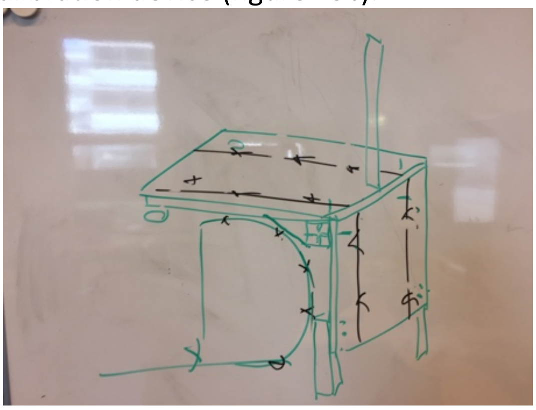

Figure 19a: Sketch

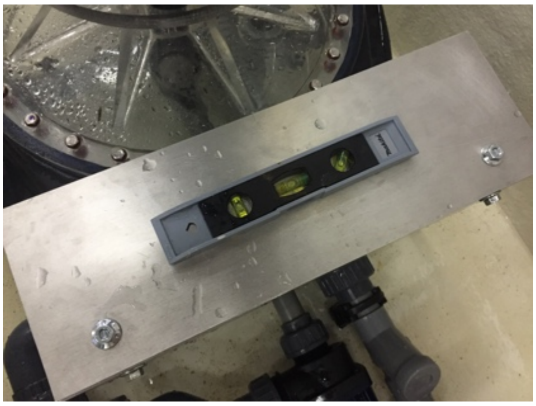

Figure 19b: Legs set at the right height

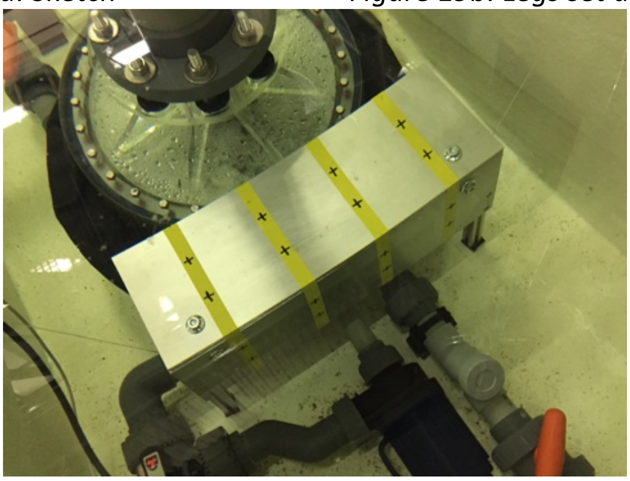

Figure 19c: Final setup with tracking points attached

A Solidworks assembly of these parts were made including distances. These are shown in figure 20.

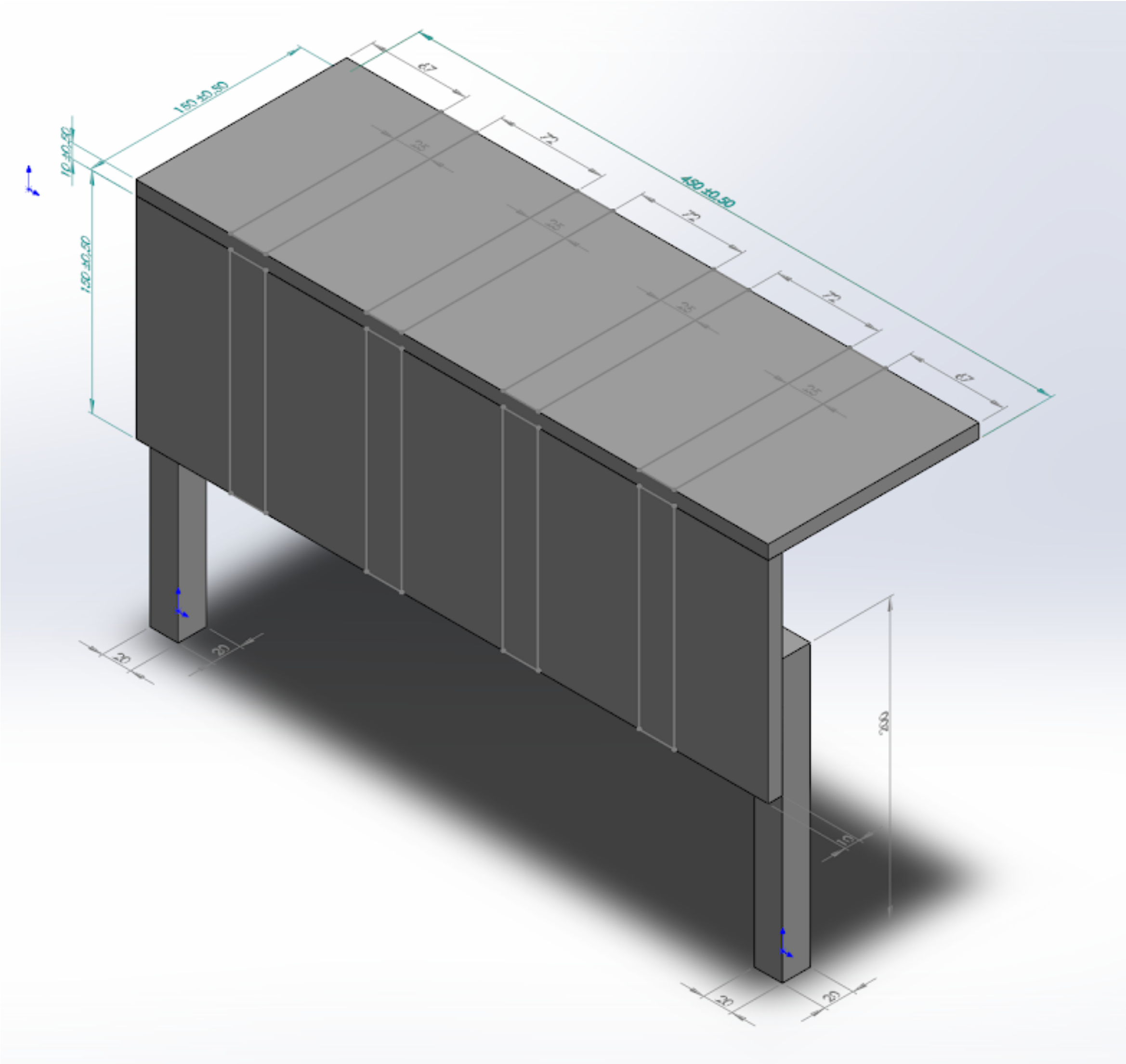

Figure 20: Solidworks assembly of calibration device

A string was connected to this device so it could be lifted out of the wave tank. The bladder was recorded with the calibration device on, then the calibration device was removed, and the moving bladder was recorded. A limitation was that this was only done once; the wave tank had to be emptied before this setup could be made again.

6 Results

In this section the results of a hanging setup at 45 degrees using 2 cameras using SSS principles are discussed, including its calibration. The FHS system was not suitable for the bladder reservoir being inside the wave tank. Every time something about the FHS setup needs to be changed, the wave tank would have had to be drained, which is time consuming and extremely bad for the environment. A hanging setup in FHS could have been made, however more angle measurements would have to be done. This would take time to get accurate values and a SSS is just simpler to build, using PVC pipes.

The X axis is not discussed in this thesis, which is because only 1 cross-section of the bladder reservoir will be investigated. If more cross-sections are investigated, the same calibration device can be used.

6.1 Calibration device

First, the data of the calibration device was obtained in Adobe After Effects; as shown in figure 21. This is done for both the left and the right camera, so 32 markers will be selected, containing data for the amount of x and y pixels.

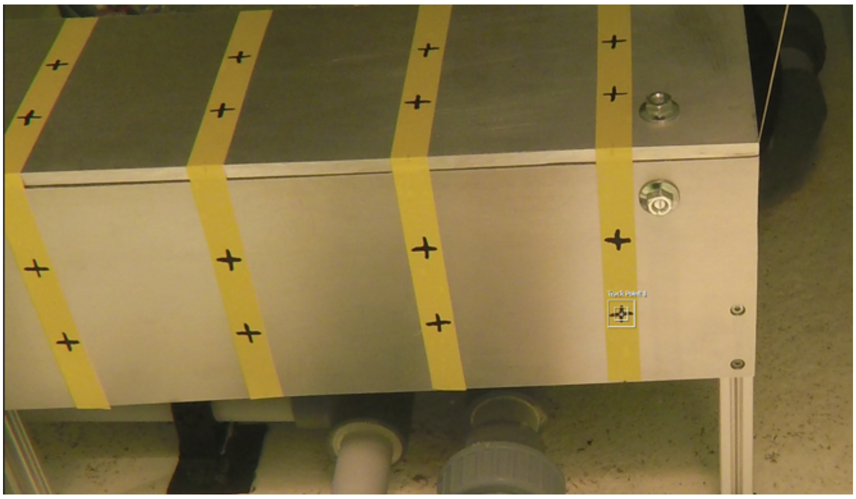

Figure 21: Calibration device as seen with GoPro

These results are put in Excel, where first some calculations were done to see if the results were clear. Some problems arose in this step. Firstly, the GoPro seemed to contain barrel (fisheye) distortion. As stated in section 4.2; the GoPro has a fish-eye lens. It is an action camera, so it is made for wide shots. There is a setting which converts the picture to a narrow view, although at the corners of the picture, the image is still being bent in a certain way. This makes the values around the corners of the image less accurate, wistfully not a certain change can be made to the formula which accurately describe the differences. This will be further discussed in the discussion section.

6.1.1 Z values

Using formulas 2, 3 and 4, the Z could be calculated for all the 16 points. Now, a way had to be found to transfer to data from the camera to an accurate x-y plane, as can be seen in figure 22:

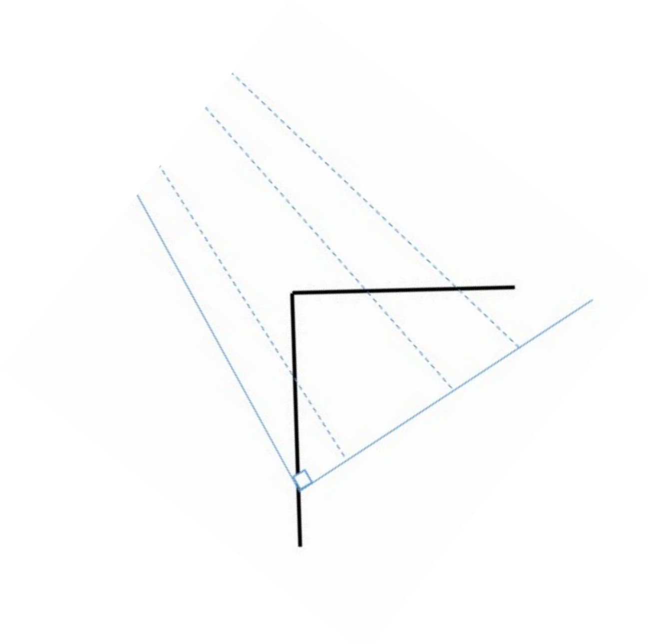

Figure 22: Schematic representation of Z values on a horizontal-vertical plane

Where the dashed lines meet the calibration device (in black) is where the current Z is calculated. The lowest blue line is located at the middle of the FOV of both cameras. The real values of Z for these distances could also be calculated by knowing the exact positions of all the points and both the cameras in 3D. The simple $a^2 = b^2 + c^2$ rule was used here to get straight line values of Z. The differences are shown in the next table, where Z is calculated by the formula, and 'Z real' is the actual distance. The table can be interpreted as follows; the first 4 rows are the top points from left to right, the next 4 rows are for the row of points underneath and this is the same for the bottom 2 columns.

Z	difference	Z real	error
838.86		790.58	
843.29	52.71	790.58	6.67%
845.52	54.94	790.58	6.95%
833.38		790.58	
796.97		755.90	
800.97	45.06	755.90	5.96%
802.98	47.08	755.90	6.23%
792.02		755.90	
800.97		764.45	
807.04	42.60	764.45	5.57%

812.18	47.73	764.45	6.24%
800.97		764.45	
839.96		800.46	
847.77	47.31	800.46	5.91%
854.58	54.11	800.46	6.76%
842.18		800.46	

Table 3: Z values

One can see difference of around 50 for each point. As stated in the introduction of the results section, the outermost dots will not be accurate because of the fish-eye effect, this is why the errors are not calculated for these points.

In the Matlab script a certain number will be subtracted from Z to match the view where the Z axis is the vertical inner of the bladder and the bottom of the bladder would be at Y=0, so nothing is done with these differences. See the discussion section.

6.1.2 Y values

Secondly, Y values were calculated, this is done using the following formula (5). Y should be the same for both images because the cameras are placed on a same height Y. This is why the average of Yl and Yr is used, these are the pixel values of a certain point.

$$y = (\frac{540 - \frac{yl + yr}{2}}{540} * \tan\left(\frac{vertical\ AOV}{2}\right) * 800.46)^{1.04} \qquad (5)$$

This will calculate the height of a point, with the lowest point being 0 at a Z of 800.46 (this value is calculated to be the real distance from the centre of the cameras to the lowest point on the calibration device, which is in the middle of the FOV. These heights will differ from the actual heights because they are reflected on an axis which is perpendicular to Z=800.46mm. The real values of Y were calculated using rules of geometry. These results contained many differences; not by a certain value as is this instance with the Z data. This is the reason why the values were calculated to the power of 1.04. This will give a value which is closest the actual answer; as shown in Table 5.

Yaverage	Yaverage^1.04	Y real	error
141.44	172.43	180.72	4.81%
145.58	177.67	180.72	1.72%
113.98	137.75	138.33	0.42%
116.05	140.36	138.33	-1.44%
30.47	34.93	33.05	-5.40%
36.30	41.91	33.05	-21.15%

-8.84		0.00	
-4.33		0.00	

Table 5: Y values

This value of 1.04 has been found by looking at the differences between the calculated Y values and the real Y values. It is impossible to add a constant or make a linear adjustment, so an exponential was used. 1.04 gives the best result. It is unclear why this error is so large, it will be discussed in the discussion section.

6.2 Recording the bladder

Then, using AAE, 10 points are tracked during 4 cycles of the bladder. On a computer with enough processing power, it will be done fairly fast. This is shown in Figure 23. First the number of pixels to start the tracking must be decided on, as explained in section 5.2.3. A point of discussion is that when a point is tracked in AAE for an extended period of time, there will be more chances of errors occurring.

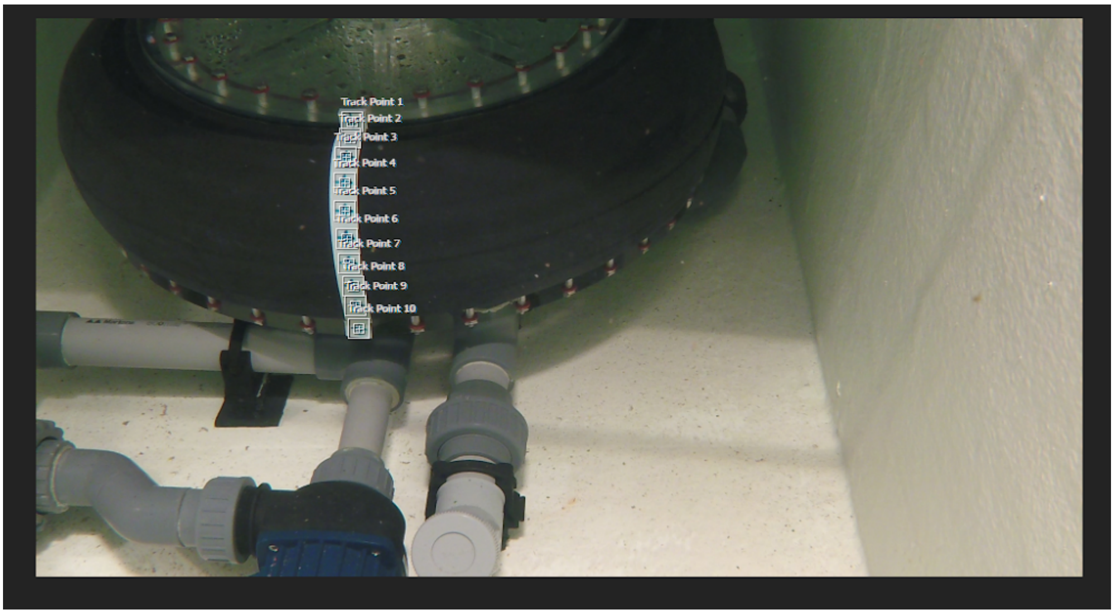

Figure 23: AAE view of bladder containing tracker points

After the tracking is completed, there are 2 possible rows to copy; the Attach Point and the Feature Centre. The attach point is one point put in the centre of every marker. This one is chosen. This data is copied to Excel. 3 columns will be shown; the frames, X pixels and Y pixels. Tracker 1 will be at the top; the other trackers will be below. Paste the left camera data in the sheet first and the right camera data in the next columns.

Now it is important to make sure that the picture is the same for different frames. Look at the light signal and add or subtract the number of frames. Basically, it is the difference of when the cameras are started.

Make sure that decimal numbers are actually seen as decimal numbers; not as thousands. This differs in US and UK versions of Excel and possibly Adobe After Effects.

Make sure that the columns match in size, if there are white rows in between, that does not matter. Save the data with a name that can easily be found.

6.3 Matlab script

In Matlab; import the previously created Excel file. In this thesis it is called data.xlsx. At the top of the screen a section ‘Unimportable Cells’ is shown. Instead of the standard ‘Replace unimportable cells with NaN’ check the options to exclude all these rows and columns with unimportable cells, so only numbers will be left. Then save this workspace variable as data.mat. Now you can run the Matlab script (for the full script; see Appendix 1).

First, the column of frame data from the right camera will be deleted which is column 4; there should be 6 columns originally. Then, all the variables will be set. With this setup they are as follows:

- Focal length is 28mm (the whole script is in mm)
- The AOV of the UW camera is 54,34 degrees (see Section 4.1.3)
- The total amount of x pixels is 1920 (for 1080p camera footage)
- The total amount of x pixels is 1080
- B it the width between both cameras; 300mm in this instance
- R is 510/2mm, which is the radius of the inside of the bladder
- The total amount of data points will be read out of the important data
- The amount of tracking points is 10

Then, an if-loop is applied to see if the total data is correctly imported. The total data divided by the amount of tracker points should be a round number. If this is not true, the program will ask to change your variables or imported data.

Then, in table T, 6 columns are added which will calculate the needed X Y and Z data.

- Column 6 will calculate Xl in millimetres
- Column 7 will calculate Xr in millimetres
- Column 8 will calculate Z from the camera
- Column 9 will calculate Y from the camera
- Column 10 will calculate Z rotated
- Column 11 will calculate Y rotated

The workings of columns 6 to 9 have been described earlier. Rotating the data is done by creating a rotation matrix. The Z and Y data is rotated around a certain point (0,0 in this instance) over an angle of 44 degrees anticlockwise. 44 degrees seemed to be the exact angle looking at the dimensions calculated in the wave tank.

A value of 740mm is subtracted from the Z data and 660 is added to the Y data to match the first and last points on the bladder.

This complete overview will be saved for later use. Now, all other data apart from Z and Y will be deleted from the table and the table will be split in 10, creating a new table for each tracker. It is impossible to create 10 new tables in a for-loop. This is called dynamic variable creation, it is the reason why tables are made one by one.

First, the program will give the possibility to ask for a certain frame. It will show the position of the bladder for that frame. It is achieved as follows:

- For a certain frame which is put in, a table 'final1' is created with Z and Y data of all 10 trackers for that specific frame.
- An array is created of this table
- A plot is made

If false numbers are pressed the program will ask to retry. If 0 is entered the program will go on to the next process, namely showing a moving picture of the bladder per frame. The same idea as before is used, although it is now calculated for every point in a while loop. Using the pause function a moving image is shown.

First, some commands are added which will create a video of the figure which is saved in your MATLAB folder in your computer. The same way of gathering the data is used as before. The values of the first and last marker (or node as it says in the script) is set equally, which is done for a more accurate volume calculation.

This volume calculation is done by creating a pchip or a spline function over the points of Y. Pchip stands for 'Piecewise Cubic Hermite Interpolating Polynomial', it returns a vector of interpolated values corresponding to the query points of yy, which are the y points multiplied by a sampling rate; 10 in this instance. A sampling rate of 10 is used to calculate a graph over 100 points in total. The outcome of the pchip function are determined by shape-preserving piecewise cubic interpolation of x and y.

Using the spline function, which does a cubic spline data interpolation, returns a vector of interpolated values s corresponding to the query points in yy is also returned. The difference is that the values of the spline function are determined by cubic spline interpolation of x and y. This results in a 'more curved' outcome, as you can see in the videos named in the Conclusion section (7.1).

There are however some errors occurring using these 2 functions. Both functions approach the points from a small value of Y to a bigger value of Y. Sometimes, the bladder will contain a small-big-small Y point. The function will return some sharp bends to correct this. See the discussion section for more information.

The volume is calculated by using the trapz function over the line created by spline or pchip (pchip was more accurate so also used more), it is added by 510/2mm, which is the radius of the inside of the bladder. This would result in the total volume of the 3D structure, so the trapz formula is used over the radius of the bladder and is subtracted from the previous trapz formula.

Time is added to the title and a figure is created showing both the diagonal view and the volume of the bladder.

The text in Matlab will look as follows:

First the program will ask which frame you want to see. Then a real time representation will be shown.
Which frame? Choose between and including 1 and 4782. Write 0 to stop. 1
Which frame? Choose between and including 1 and 4782. Write 0 to stop. 0
Running...
Done
Elapsed time is 685.413024 seconds.

This is the time it takes to analyse 4 cycles; around 11 minutes. 1 cycle will take 191.401802 seconds; around 3 minutes.

6.4 Results

The video of the Matlab outcome can now be compared to the COMSOL model and also to camera recordings from a side view of the bladder. Data in Matlab is transformed to match the data in COMSOL. In figures 25a and b one can see the outcome after one cycle of the bladder reservoir in Matlab and COMSOL, respectively. In figures 25c and d, two figures are shown representing the bladder being halfway through a cycle.

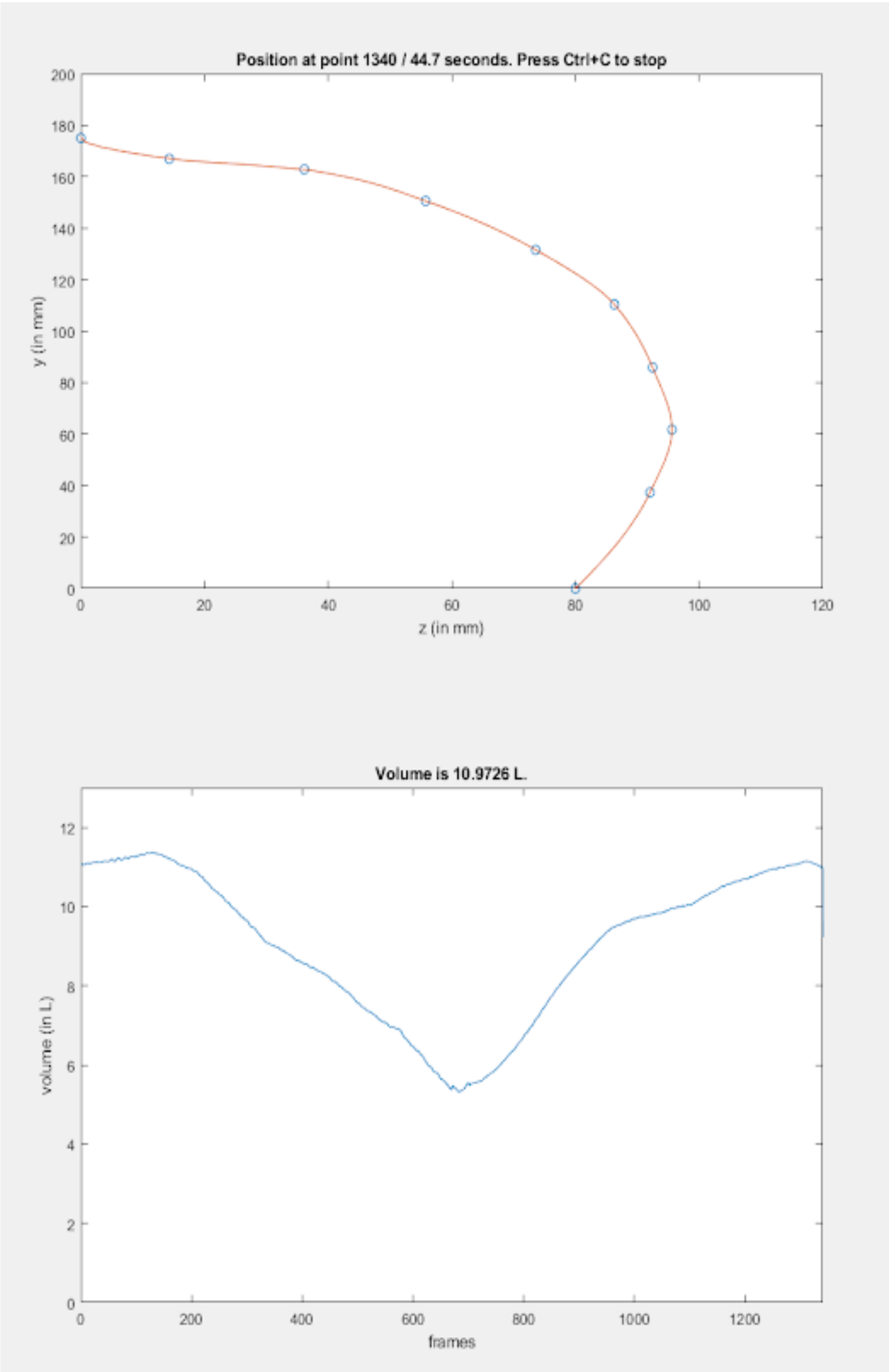

Figure 25a: Outcome of Matlab after 1 cycle

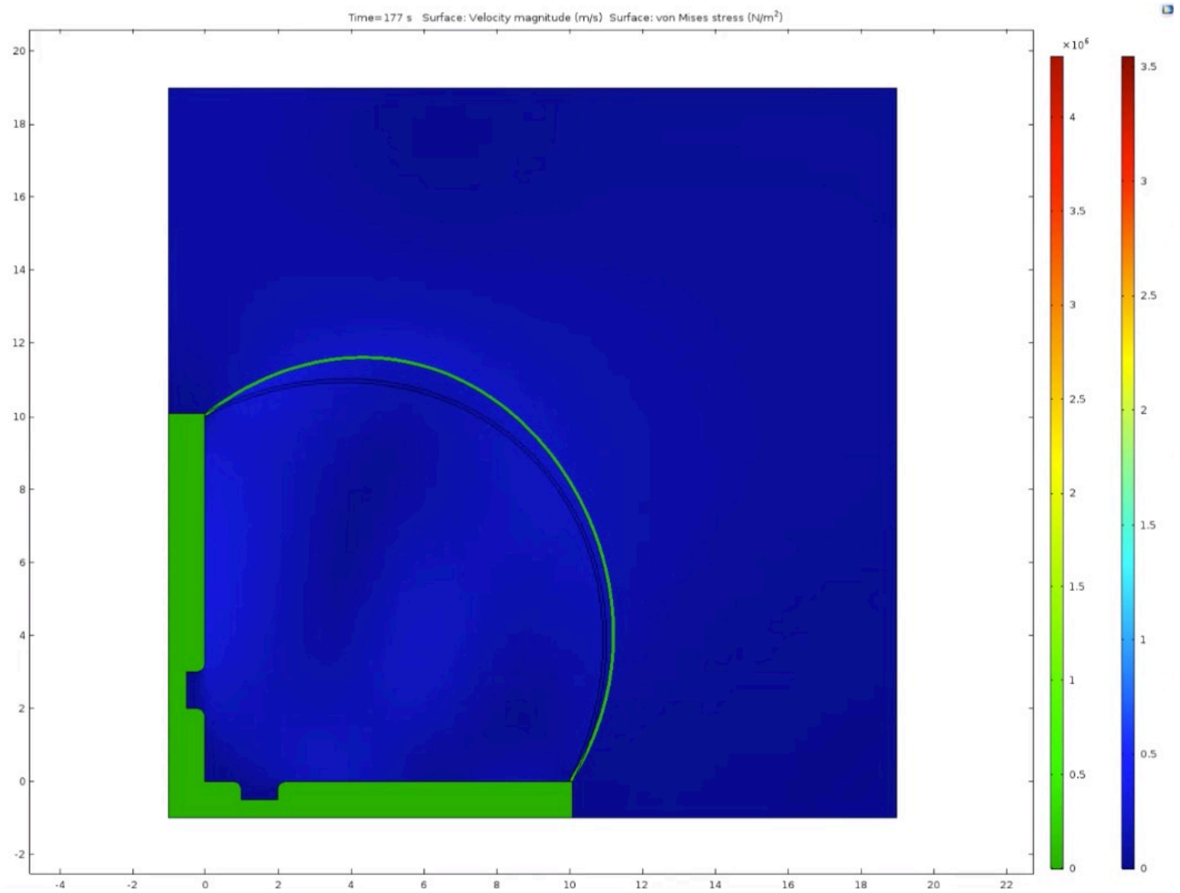

Figure 25b: Outcome of COMSOL after one cycle

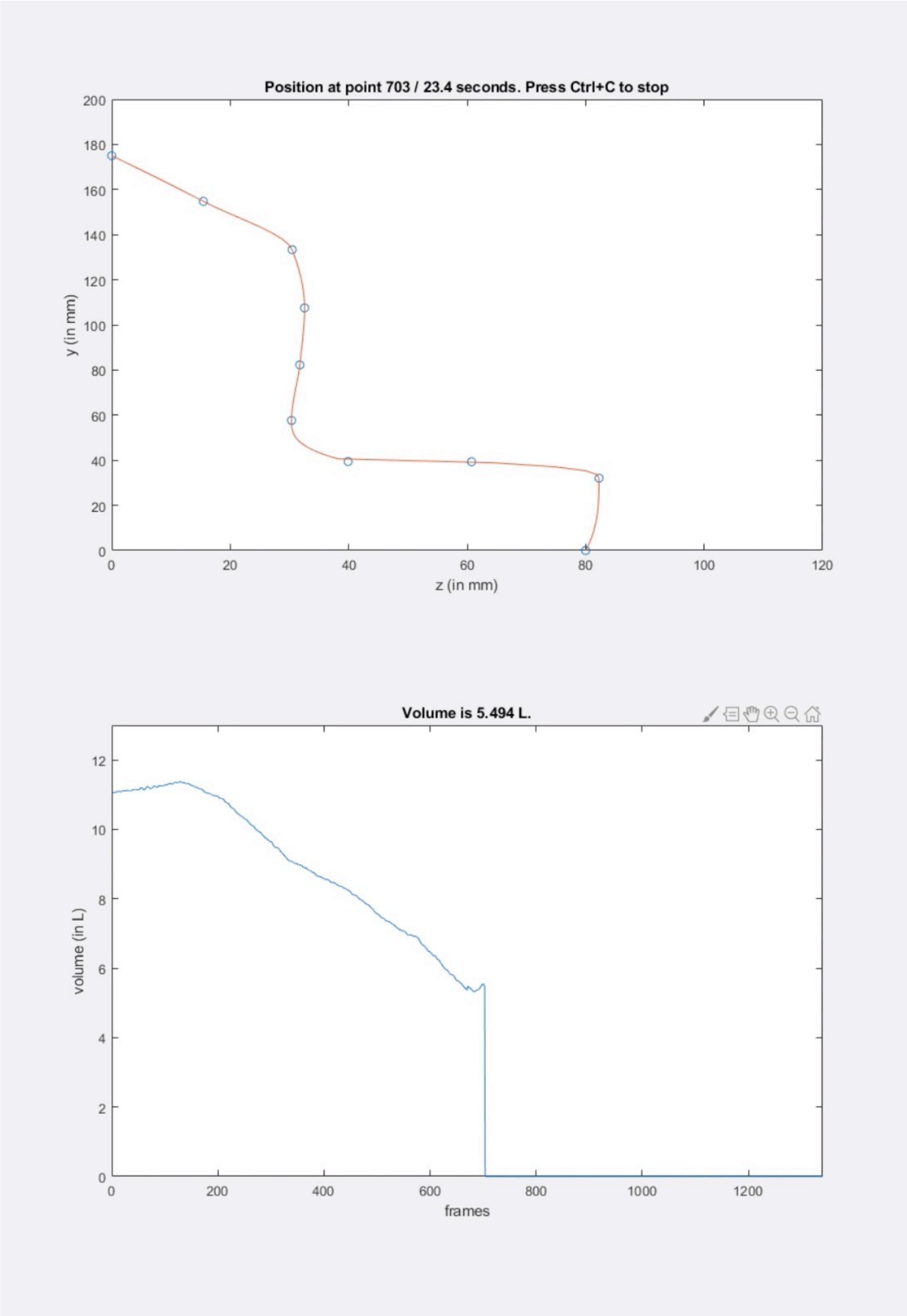

Figure 25c: Outcome of COMSOL half-way a cycle

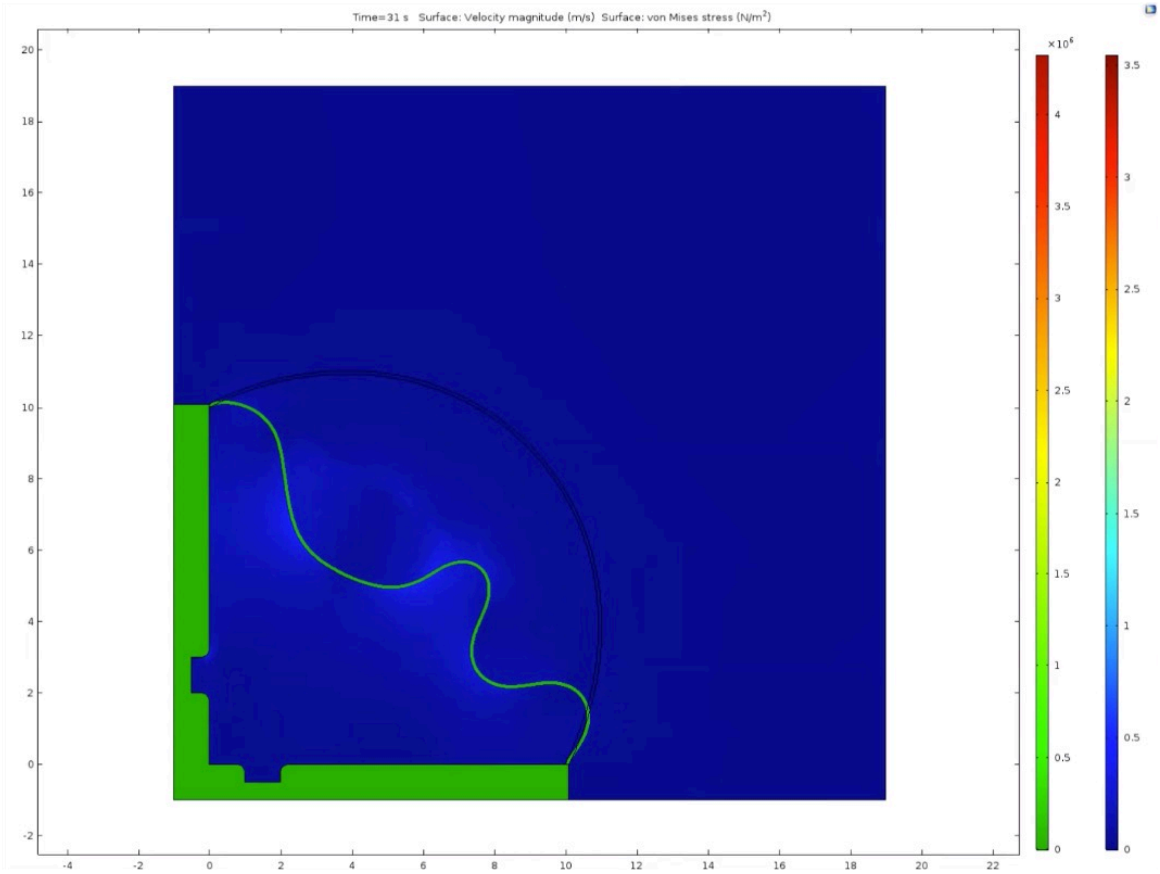

Figure 25d: Outcome of COMSOL half-way a cycle

There are some obvious differences to be seen. First of all, in a running COMSOL model, one will observe that the COMSOL model is not realistic. It does not represent the slow movements of the bladder.

As we can see in the results of Matlab, the bladder is clearly being deflated with the middle moving inwards first, compared to the COMSOL model, where the middle of the bladder keeps sticking out and the parts around it deflate.

The dimensions of the COMSOL model are still inaccurate, nevertheless, the dimensions of the Matlab model are also inaccurate. In the following figure (26) the real dimensions of the bladder are shown. The main difference is in the Y axis.

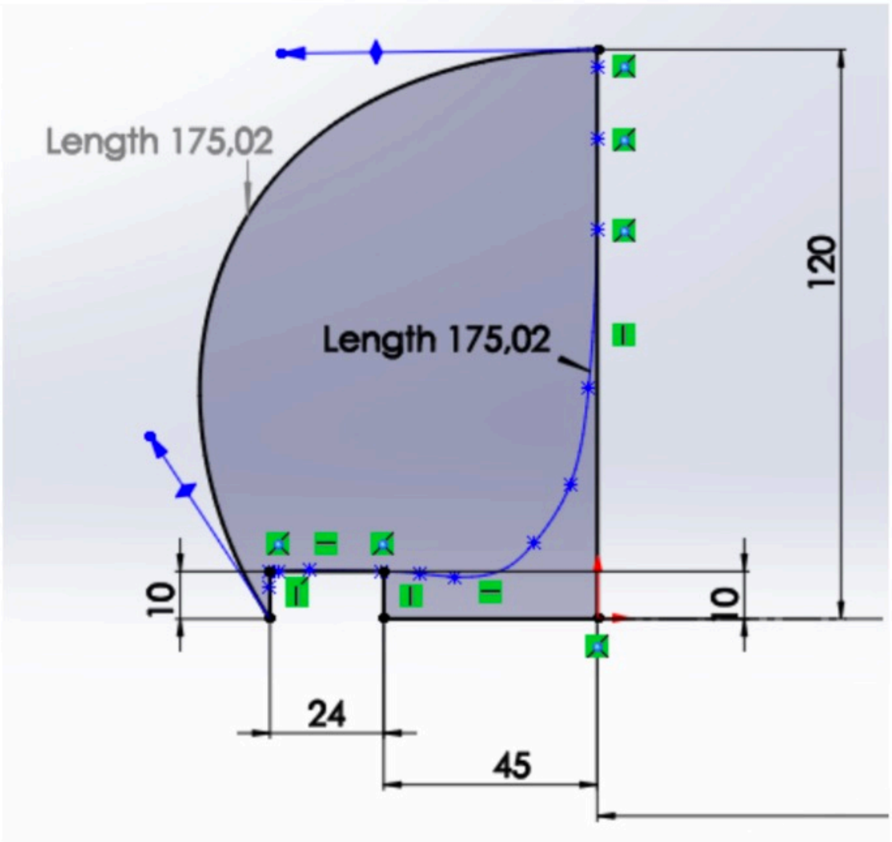

Figure 26: Dimensions of the bladder

7 Conclusions

7.1 Conclusions

To answer the previously formulated research question, *"How does the proof-of-concept of the Bladder Storage System behave?"*, the sub-questions must be answered first.

The first sub-question is *"How does one calculate 2D displacement using 3D data, taking into account the changing optics of having a proof-of-concept which is located under water?"*. This question is answered throughout the report, starting with a literature research.

Formulas for calculating Z and Y where formulated and applied to current cameras, with certain focal lengths and angles of views. The effect of recording movements under water are taken into consideration. A setup to use two cameras using the Simple Stereo method was built and placed under an angle to make sure that all the markers would be in the field of view. The calculated points under an angle were multiplied by a so called 'rotation matrix' so every point would shift a certain amount of degrees; 44 degrees anticlockwise using this setup.

The second sub-question is *“How does one create a graph containing points which change over time?”*. This question is mainly answered in section 6.3 where the Matlab script is discussed. In a while-loop, for each frame a table is created containing Z and Y coordinates. These points can be plotted by using a plot function and a line between the points can be added. However, this line would not be smooth, unlike the actual bladder, so a function (either pchip or trapz) is used to create this smooth line. By having this line, the area under this line can be calculated. Using this area another calculation is done where the volume of the round bladder is calculated. This is plotted in a graph below the movements of the bladder. Here it can be seen in which stage the bladder is and a calculation of the volume in litres.

To answer the main research question, the figures 25 a, b, c and d will help as well as the videos created by Matlab, namely *Bladder_simulation_COMSOL* (showing the COMSOL model), *Bladder_simulation_pchip_matlab* (showing 1 cycle of the bladder using the pchip function), *Bladder_simulation_pchip20points_matlab* (showing 1 cycle of the bladder using the pchip function for only 20 points), *Bladder_simulation_spline_matlab* (showing 1 cycle of the bladder using the spline function), *Bladder_simulation_line_matlab* (showing 1 cycle of the bladder with a line. The volume is calculated using the pchip function), and *Bladder_simulation_line4_matlab* (showing 4 cycles of the bladder with a line. The volume is calculated using the pchip function).

There are some obvious differences to be seen. A running COMSOL model does not represent the slow movements of the bladder. As we can see in the results of Matlab, the bladder is clearly being deflated by the middle moving inwards first, compared to the COMSOL model, where the middle of the bladder keeps pointing outwards and the parts around it deflate.

The efficiency of the storage can also be looked at. The ideal situation would be that the bladder fully empties as also shown in figure 26. This is not possible because of the folds being created by the extra EPDM. It would also have drawbacks, for instance the lifetime of the rubber could decrease because of rubbing which causes breaking. The efficiency should still be as high as possible. From the master thesis of Ton Konings, the theoretical efficiency of the system can be calculated using the following statement.

‘$Total\ amount\ of\ moving\ water\ between\ reservoirs$ = 12,52 $Liter$ − 0,98 $Liter$ = 11,54 $Liter$’

This would mean that an efficiency of 92,17 percent would be possible. In the Matlab model an efficiency of $\frac{10.55-4.50}{10.55} * 100\%$ litres can be calculated, which corresponds with an efficiency of 57,35 percent.

7.2 Discussion

In this section the accuracy of the results will be discussed. Firstly, the so called ‘barrel effect’ of the GoPro fisheye lens has to be resolved. Finally, the Y data was calculated to the power of 1.04 to decrease the errors. This is not a correct approach, the real reason for this change must be found. However, in the time-span of this thesis the reason has not been found. Because of 1 camera behaving differently, the accuracy of the calculations cannot be stated. Normally, error can be calculated by the absolute value of the average minus 1 of the 2 Y values divided by the average times 100 percent.

Furthermore, there are always some errors present in a setup which is mobile. In the system; differences of millimetres are measured. Every unclear value must be taken into account. For example, the distance B, the distance between the centre of the lenses of both cameras could not be measured exactly using a ruler, because of the PVC pipe in the middle of it. Instead, a piece of string is wrapped around 2 pencils which calculate the width, although some error must be taken into account because of inaccuracy.

Errors occurred in the AAE tracking. The most accurate way is the follow each point and correct it when it moves, however, this is highly time consuming for 10 different points and around 4800 frames. This will also result in more inaccuracy.

Another error occurred using pchip and spline. In section 6.3, about the Matlab script, the following was stated: The function approaches the points from a small value of Y to a bigger value of Y. Sometimes, the bladder will contain a small-big-small Y point. The function will return some sharp bends to correct this. Together with Postdoc Yanyi Wei this problem has been investigated, however, no clear solution of how to solve this problem has been found.

7.3 Limitations

The first limitation was the high glass on the side of the wave tank. This resulted in the inability to use the camera supplied by the OGG. Cameras had to be found elsewhere, fortunately I own an UW action camera and Postdoc Yanyi Wei brought his camera. The perfect situation would have been where 2 identical cameras were available when the thesis started. In that instance, experiments could have been done immediately. This has to do with the time-span of only 9 weeks.

The Nikon and the GoPro have different mechanisms to be attached to a PVC pipe. A GoPro can be easily taken out of its housing or be unscrewed; however, the Nikon camera has a screw at the bottom like most regular cameras. To connect it to a piece of PVC pipe, a tripod head attachment device was used. A plastic hole was drilled, and metal parts are set in the water, which will decrease their quality. Dr. Eize Stamhuis did have GoPro's at his disposal, unfortunately he was unable to supply them.

Depth cameras could have been used and more accurate results might have been obtained using these, however, this would have decreased the difficulty of the project.

Due to time constraints, Apriltags were not investigated to be used within Matlab. This would have made it possible to account for the fisheye effect of the GoPro to get more accurate results

The bladder had been placed inside the wave tank, which took a long time to fill up and drain the water again. This is time consuming, bad for the environment, and it makes it more difficult to easily carry out experiments, because everything has to be prepared well. The final calibration setup has only been carried out once because of this reason.

Finally, the bladder was leaking. This was not a big problem because water could be drained by a hose under the Bernoulli equation, although this did cost some time. Also, if multiple experiments would have been done for more accurate results, these would differ because of water coming into the bladder.

7.4 Recommendations

In order to reach higher Technology Readiness Levels, several follow up researches can be done.

This research mainly contributes to a further improvement of a model for the storage system. This can be a COMSOL model, which has already been made, though it needs further improvement. Other programs can be used too. The final goal would be to have an accurate model of a large-scale system before building a new prototype or POC.

The POC was designed with the possibility to connect different kinds of sensors, for example to study the pressure behaviours in the rigid and bladder reservoir using pressure sensors positioned in both reservoirs. Also, water velocity can be studied by using the mini water flow measurement systems positioned in the pipeline system below the prototype. Flow data could also show the amount of water which is inside the reservoir at a specific time, which will say something about the efficiency. This information will be interesting to grasp the concept better.

Furthermore, the POC should be tested in a depth water basin test set-up. It is useful to study the behaviour of the POC in this depth water basin, since more pressure will work on the bladder reservoir. These tests should be done to discover if the theoretical ideas about the movement of the bladder are in line with practical observations.

Finally, only a prototype of the Ocean Grazer 3.0 is made which is concentrated on the working of the bladder reservoir. It is recommended to create a prototype that will demonstrate the working of the complete OG. When creating a prototype of the complete OG; the working of the pumping system which transforms wave energy into potential energy will also be included. It will be interesting to study the behaviour of the pumping system together with the performance of the bladder reservoir in practice. Investigating the role that the floater plays in the total system is important, especially when the total system is scaled to a larger size. When the floater is incorporated in the model, the complete power generation of the Ocean Grazer can be simulated, where new problems might arise.

References

[1] BP, BP Statistical Review of World Energy: June 2018. https://www.bp.com/content/dam/bp/business-sites/en/global/corporate/pdfs/energy-economics/statistical-review/bp-stats-review-2018-full-report.pdf, 2018
(accessed 12 Jan 2019)

[2] Tennet, Windenergie: het net op zee. https://www.tennet.eu/nl/onze-kerntaken/innovaties/north-sea-wind-power-hub/
(accessed 2 Nov 2018)

[3] W.A. Prins, M. van Rooij, A.I. Vakis, B. Jayawardhana, Technical White Paper: Reservoir Storage Structure (2017)

[4] Ocean Grazer BV, Ocean Grazer. http://www.oceangrazer.com/
(acceded 31 Oct 2018)

[5] J.A. Koning, Studying, designing, creating and testing the bladder reservoir of the Ocean Grazer 3.0, Master Thesis Industrial Engineering & Management (2018)

[6] Skelia, MVP VS POC VS PROTOTYPE: WHAT DOES YOUR COMPANY REALLY NEED? https://skelia.com/articles/mvp-vs-poc-vs-prototype-company-really-need/
(accessed 31 Oct 2018)

[7] Sietse van den Elzen, The Ocean Grazer: Designing a flexible underwater reservoir, BACHELOR INTEGRATION PROJECT (2018)

[8] Prins, W.A., van Rooij, M., A.I. Vakis & B. Jayawardhana, Underwater energy storage system, European patent application EP17206416.4 (11 December 2017)

[9] ESA, TECHNOLOGY READINESS LEVELS HANDBOOK FOR SPACE APPLICATIONS. https://artes.esa.int/sites/default/files/TRL_Handbook.pdf
(accessed 22 Jan 2019)

[10] Hevner, Alan R, A Three Cycle View of Design Science Research, Scandinavian Journal of Information Systems: Vol. 19 : Iss. 2 , Article 4. (2007)

[11] Sundara Tejaswi Digumarti, Gaurav Chaurasia, Aparna Taneja, Roland Siegwart, Amber Thomas, Paul A. Beardsley, Underwater 3D capture using a low-cost commercial depth camera, 2016 IEEE Winter Conference on Applications of Computer Vision (WACV). DOI:10.1109/WACV.2016.7477644

[12] Video: Ocean Grazer, Storage_final (2018)

[13] Dr. Eize J. Stamhuis (personal communication, 18 Dec 2018)

[14] Lennard Hut, Investigating the dynamic behaviour of a multi-piston pump, Bachelor Integration Project performed at The Ocean Grazer Project (2018)

[15] Bordeaux Aquitaine, GOPRO CAMERAS MATRIX AND DEPTH MAP IN COMPUTER VISION. www.labri.fr/perso/falleri/dist/ens/projets/s8/rapports/1.pdf (accessed 2 Jan 2019)

[16] Ricardo Martin Brualla, Paul G. Allen School of Computer Science and Engineering of the University of Washington, CSE445, Lecture 16: Computer Vision, Stereo. (2018)

[17] David W. Knight, Cameras Underwater, The angle of coverage (or field of view) of a lens. https://www.camerasunderwater.co.uk/lens-fov (accessed 31 Oct 2018)

[18] Nikon, Coolpix AW100 Reference Manual. https://cdn-10.nikon-cdn.com/pdf/manuals/kie88335f7869dfuejdl=-cww2/AW100EN.pdf (accessed 14 Dec 2018)

[19] GoPro, Hero 5 Session User Manual. https://gopro.com/content/dam/help/hero5-session/manuals/HERO5Session_UM_ENG_REVD_WEB.pdf (accessed 14 Dec 2018)

[20] April Robotics Laboratory, AprilTags https://april.eecs.umich.edu/software/apriltag.html (accessed 14 Dec 2018)

[21] A.D. Jepson and D.J. Fleet, Computer Science University of Toronto, Epipolar Geometry. http://www.cs.toronto.edu/~jepson/csc420/notes/epiPolarGeom.pdf (accessed 14 Dec 2018)

Appendices

Appendix 1: Matlab script: calculationvolume.m

Contents


```
clc;
clear all
close all;
T_raw=load('data.mat');%loading data file with NaN removed
T=T_raw.data;
T(:,4)=[]; %delete frame data

%setting variables
f=28; %focal length
horizangleUW=48.59; %horizontal underwater angle of view
vertangleUW=28.48;
xpixels=1920;
ypixels=1080;
b=300; %width between 2 cameras
r=510/2; %radius of inside of the bladder
totaldatapoints=height(T);
trackerpoints=10;
datapoints=totaldatapoints/trackerpoints;
if round(datapoints)==datapoints %check for integer
```

```
    %adding 6 collumns
    NewCol=rand(totaldatapoints,1); %easy way to create a new collumn
    T=addvars(T,NewCol,NewCol,NewCol,NewCol,NewCol,NewCol); %adding 6
collumns, 2 for changing x to mm values, z, y and z and y rotated
    T.Properties.VariableNames = {'frame' 'x1' 'y1' 'x2' 'y2' 'x1mm' 'x2mm'
'zcamera' 'ycamera' 'z' 'y'};

    %doing calculations for the first 4 collumns
    T{:,6}=((T{:,2}-(xpixels/2))/(xpixels/2)).*tand(horizangleUW./2).*f;
    T{:,7}=(((xpixels/2)-T{:,4})/(xpixels/2)).*tand(horizangleUW./2).*f;
    T{:,9}=((((ypixels/2)-
((T{:,3}+T{:,5})./2))./(ypixels/2)).*tand(vertangleUW./2).*800.46).^(1.04);
     T{:,8}=sqrt(((f.*b)./(T{:,6}+T{:,7})).^2 – (T{:,9}).^2);

    % Create rotation matrix
    theta = -44; % to rotate 44 degrees counterclockwise
    R = [cosd(theta) -sind(theta); sind(theta) cosd(theta)];
    %point where the data must be rotated on, in this case 0,0, will be
    %accounted for later on
    z_center = 0;
    y_center = 0;
    center = repmat([z_center; y_center], 1, totaldatapoints);
    v = real([T{:,8}';T{:,9}']); %somehow imaginairy numbers show up
    s = v - center;      % shift points in the plane so that the center of
rotation is at the origin
    so = R*s;            % apply the rotation about the origin
    vo = so + center;    % shift again so the origin goes back to the
desired center of rotation
```

```
    T{:,10} = 740-vo(1,:)'; % put in table T, and setting values to fit
    T{:,11} = 660+vo(2,:)';

    %only keeping Z and Y
    A=T; %saving T
    A(:,9)=[];
    A(:,8)=[];
    A(:,7)=[];
    A(:,6)=[];
    A(:,5)=[];
    A(:,4)=[];
    A(:,3)=[];
    A(:,2)=[];
    A(:,1)=[];

    %splitting table into amount of tracker points
    T1=A(1:datapoints,:);
    T2=A((datapoints+1):(datapoints*2),:);
    T3=A(((2*datapoints)+1):(datapoints*3),:);
    T4=A(((3*datapoints)+1):(datapoints*4),:);
    T5=A(((4*datapoints)+1):(datapoints*5),:);
    T6=A(((5*datapoints)+1):(datapoints*6),:);
    T7=A(((6*datapoints)+1):(datapoints*7),:);
    T8=A(((7*datapoints)+1):(datapoints*8),:);
    T9=A(((8*datapoints)+1):(datapoints*9),:);
    T10=A(((9*datapoints)+1):(datapoints*10),:);

    clear A;
```

asking for which frame

```
    disp (['First the program will ask which frame you want to see. Then a real 
time representation will be shown.']);
    pause(2);
    %creating table for graph for a certain frame
    while 1
        frame=input(['Which frame? Choose between and including 1 and ' 
num2str(datapoints) '. Write 0 to stop. ']);
        if frame==0
            break
        else
            if frame<=datapoints
                Tgraph1=array2table(rand(10,2));
                Tgraph1(1,:)=T1(frame,:);
                Tgraph1(2,:)=T2(frame,:);
                Tgraph1(3,:)=T3(frame,:);
                Tgraph1(4,:)=T4(frame,:);
                Tgraph1(5,:)=T5(frame,:);
                Tgraph1(6,:)=T6(frame,:);
                Tgraph1(7,:)=T7(frame,:);
                Tgraph1(8,:)=T8(frame,:);
                Tgraph1(9,:)=T9(frame,:);
                Tgraph1(10,:)=T10(frame,:);
                Tgraph1.Properties.VariableNames = {'z' 'y'};
                graph1=table2array(Tgraph1);
                plot(graph1(:,1),graph1(:,2),'o-')
                xlabel('z')
                ylabel('y')
                title(['Position at frame ' num2str(frame)])
                axis([0 120 0 200])
            else disp('Try again')
            end
        end
    end
```

```
    else disp('The amount of frames is not an integer, please change your
    variables.')
    end
```

plotting all datapoints

```
point=1;
volumetable=zeros(datapoints,2);
v=VideoWriter('CHANGENAME.avi'); %to remind someone to change the name before the
file is rewritten
v.Quality=100;
v.FrameRate=30;
open(v);
tic;
figure
disp('Running...');
while point<=1340 %change to see the desired amount of datapoints
    Tgraph2(1,:)=T1(point,:);
    Tgraph2(2,:)=T2(point,:);
    Tgraph2(3,:)=T3(point,:);
    Tgraph2(4,:)=T4(point,:);
    Tgraph2(5,:)=T5(point,:);
    Tgraph2(6,:)=T6(point,:);
    Tgraph2(7,:)=T7(point,:);
    Tgraph2(8,:)=T8(point,:);
    Tgraph2(9,:)=T9(point,:);
    Tgraph2(10,:)=T10(point,:);
    Tgraph2.Properties.VariableNames = {'z' 'y'};
    graph2=table2array(Tgraph2);
    x = graph2(:,1);
    y = graph2(:,2);
    x(1)= 0; %setting values for the first and the last node
    x(10)= 80;
    y(1)= 175;
    y(10)= 0;

    %using a spline function to create a smooth line between points
    samplingRateIncrease = 10;
    % the next 3 lines are a different way to get yy
    %     for i=1:(samplingRateIncrease-1)
    %         yy(10*(i-1)+1:10*i)=linspace(y(i),y(i+1),10);
    %     end
    yy = linspace(min(y),max(y),length(graph2)*samplingRateIncrease);
    xx=pchip(y,x,yy); %can be changed to spline
    %volume calculation
    xbladder=xx+r;
    rx=r.*(ones(1,length(yy)));
    volume=(pi()*r*(trapz(yy,(xbladder))-trapz(yy,rx)))/1000000; %To get volume in
litres
    volumetable(point,:)=[point,volume];
    %calculating time to add to title
    time=round(point/3)/10;

    %actual plot
    subplot(2,1,1);
    plot(x,y,'o',x,y,'-') %can be changed to show spline/pchip or just a line
    xlabel('z (in mm)')
    ylabel('y (in mm)')
    title(['Position at point ' num2str(point) ' / ' num2str(time) ' seconds. Press
Ctrl+C to stop'])
    axis([0 120 0 200])
```

```
    subplot(2,1,2);
    plot(volumetable(:,2));
    title(['Volume is ' num2str(volume) ' L.'])
    xlabel('frames')
    ylabel('volume (in L)')
    axis([0 1340 0 13]) %change to the amount of points desired

    set(gcf,'position',[10,10,900,1300])
    pause(0.0001);
    video=getframe(gcf); %write video
    writeVideo(v,video);
    point=point+1;
end
disp('Done')
toc
close(v);

% publish('calculationvolume.m') to publish this script
```

Published with MATLAB® R2018b

Appendix 2: Derivation of formulas

With 2 non-parallel cameras, the ‘free-hand stereo method’:

$$\tan(\alpha) = \frac{z}{B - dB}$$

$$\tan(\beta) = \frac{z}{dB}$$

$$B - dB = \frac{z}{\tan(\alpha)}$$

$$dB = \frac{z}{\tan(\beta)}$$

$$B - \frac{z}{\tan(\beta)} = \frac{z}{\tan(\alpha)}$$

$$\frac{B\ tan(\beta) - z}{\tan(\beta)} = \frac{z}{\tan(\alpha)}$$

$$B\tan(\alpha)\tan(\beta) - z\tan(\alpha) = z\tan(\beta)$$

$$B\tan(\alpha)\tan(\beta) = z\,(\tan(\alpha) + \tan(\beta))$$

$$z = \frac{B\ \tan(\alpha)\tan(\beta)}{\tan(\alpha) + \tan(\beta)}$$

For parallel cameras, the ‘simple stereo method:

$$\frac{z}{f} = \frac{x}{xl} = \frac{b - x}{xr}$$

$$x = \frac{z\,xl}{f} = \mathrm{b} - \frac{z\,xr}{f}$$

$$z\,xl = bf - z\,xr$$

$$z\,(xl + xr) = bf$$

$$z = \frac{bf}{xl + xr}$$

Appendix 3: List of hours worked

#	Date	Time		Activity	Comments
1	12-Sep	20.00	23.00	Project research	
2	13-Sep	13.00	15.00	Kick off	
3	16-Sep	21.00	22.30	First research on the Ocean Grazer	
4	19-Sep	13.00	15.00	First meeting supervisor	
5	19-Sep	22.00	23.00	Literacy Test	
6	20-Sep	10.30	13.00	Literacy Test and Preliminary Literature Results	
7	20-Sep	13.00	15.00	Library Training	
8	21-Sep	15.00	17.00	Preliminary Literature Results	
9	23-Sep	15.00	18.00	Preliminary Literature Results	
10	24-Sep	9.00	11.00	Preliminary Literature Results	
11	24-Sep	14.30	17.00	PLR and arranging a meeting with supervisor	
12	26-Sep	9.00	10.15	Meeting with supervisor + preparation	
13	27-Sep	14:00	15:30	Working on PAL	
14	30-Sep	13:30	15:00	Working on PAL	
15	1-Oct	10:00	17:00	Working on PAL	
16	4-Oct	13:00	13:45	Ocean Grazer Group meeting	
17	4-Oct	13:45	15:00	Working on RDP	Asking Marijn about usable hardware
18	5-Oct	9:15	13:00	Working on RDP	Working mainly on layout and useful sections
19	8-Oct	9:15	13:00	Working on RDP and presentation	
20	8-Oct	13:00	15:00	Presentation to supervisors	
21	9-Oct	11:00	12:45	Working on RDP	Asking Marijn about my problem statement
22	10-Oct	9:00	11:45	Working on RDP	
23	11-Oct	11:50	13:00	Working on presentation for OGG	
24	11-Oct	13:00	14:30	Ocean Grazer Group meeting and presentation	
25	13-Oct	13:15	17:00	Working on RDP	
26	14-Oct	20:45	22:00	Working on RDP	
27	14-Oct	9:30	17:00	Finishing RDP and handing-in	
28	18-Oct	11:30	13:00	Working on mathematics	
29	18-Oct	13:00	15:00	Meeting OGG	
30	24-Oct	9:00	12:30	Working on presentation	
31	24-Oct	20:30	22:30	Working on presentation	
32	25-Oct	9:15	10:00	Going over presentation	
33	25-Oct	10:00	12:00	Presenting for A. Geertsema	
34	25-Oct	13:00	15:30	Ocean Grazer Group meeting	
35	29-Oct	9:30	15:30	Working on the repair	
36	29-Oct	21:30	22:45	Working on the repair	
37	30-Oct	13:15	14:30	Preparing meeting Antonis and Marijn	
38	30-Oct	14:30	15:00	Meeting Antonis and Marijn	

39	30-Oct	15:00	17:00	Doing literature research for the repair	
40	30-Oct	19:30	21:30	Doing literature research for the repair	
41	31-Oct	12:00	17:00	Working on the repair	
42	31-Oct	19:00	23:00	Working on the repair	
43	1-Nov	9:00	13:00	Working on the repair	
44	1-Nov	13:00	13:15	OGG meeting	
45	1-Nov	13:15	14:00	Working on a plan to do calculations for cameras	
46	6-Nov	9:30	12:30	Working on a plan to do calculations for cameras	Waiting for reaction Geertsema and Vakis
47	12-Nov	10:00	16:15	Working on a plan to do calculations for cameras	
48	13-Nov	10:30	13:30	Working on a plan to do calculations for cameras	
49	13-Nov	13:30	15:00	Start-up meeting and academic writing	
50	15-Nov	10:00	11:30	Academic writing session	
51	15-Nov	11:30	13:00	Making plan	Talking to Marijn about wave tank usage, sensors and adhesives on bladder
52	15-Nov	13:00	13:45	OGG meeting	
53	16-Nov	15:00	16:45	Set up for adhesive spot experiment	
54	19-Nov	10:45	16:45	Finishing first experiment, emptying wave tank, getting ready for tomorrow	
55	20-Nov	10:00	12:45	Working on the first setup, calculating and processing data in Excel	
56	21-Nov	9:30	15:00	Filling wave tank, building setup for experiment	
57	21-Nov	16:00	17:30	Using Adobe After Effects	
58	21-Nov	21:45	23:00	Literature research	
59	22-Nov	12:00	13:00	Finishing presentation	
60	22-Nov	13:00	14:15	OGG meeting	
61	22-Nov	16:00	16:45	Working on calculations in Excel	
62	23-Nov	15:30	18:00	Working on calculations in Excel	
63	26-Nov	10:15	16:45	Working on calculations in Excel	
64	27-Nov	13:30	17:15	Working on calculations in Excel	
65	28-Nov	12:30	16:30	Working on calculations in Excel	
66	29-Nov	11:15	16:30	Working on calculations in Excel	Including OGG meeting and interview with Zaki
67	30-Nov	12:30	16:00	Writing intermediate report	
68	3-Dec	9:00	9:30	Betting books from library	
69	3-Dec	11:00	12:00	Meeting Simon and Martin at DTPA lab	Learning about ROS
70	3-Dec	14:00	16:30	Working on calculations	
71	4-Dec	11:00	17:15	Reading photogrammetry book	Finding Nikon specifications, sensor size, focal length
72	5-Dec	13:00	16:45	Studying Matlab and April tags	
73	6-Dec	12:00	16:45	OGG meeting and processing last results	
74	7-Dec	11:00	13:00	Making new plan for a new setup	
75	10-Dec	12:00	16:00	Making new plan for a new setup	
76	11-Dec	19:00	22:00	Building Solidworks model and working on presentation	
77	12-Dec	20:30	23:45	Working on presentation	

78	13-Dec	9:00	16:45	Presenting twice and making plan, visiting DTPA lab	
79	14-Dec	9:00	16:30	Visiting DTPA lab and doing new calibration experiment	
80	17-Dec	9:45	16:30	Building new setup	
81	18-Dec	16:15	18:00	Analyzing new data	
82	19-Dec	9:30	13:45	Interview Yanji, getting first measurements	
83	20-Dec	11:00	17:00	Doing measurements, calculating, OGG meeting	
84	21-Dec	9:00	17:15	Calibration and 3 measurements in the wave tank	
85	23-Dec	14:00	14:30	Doing data analysis	
86	2-Jan	10:00	12:15	Working on Matlab script	
87	2-Jan	18:15	23:30	Working on Matlab script	
88	3-Jan	14:00	16:45	Checking Journal of Energy Storage, writing preliminary report	
89	4-Jan	9:30	13:00	Getting calculations clear	
90	4-Jan	14:30	15:45	Spellchecking	
91	7-Jan	14:30	16:30	Creating models in Viseo and explaining Matlab script	
92	8-Jan	12:00	16:30	Explaining calibration and writing about setup	
93	9-Jan	11:30	16:30	Interview Marijn, building calibration device	Discussing Matlab model
94	10-Jan	10:00	17:30	Working on Matlab model, meeting, interview with Wout, and discussing Matlab with Yanyi	
95	10-Jan	21:00	01:00	Writing preliminary report	
96	11-Jan	9:00	17:00	Calibration experiment and writing report	
97	14-Jan	9:00	16:30	How to implement calibration	
98	15-Jan	9:00	12:45	Interview Yanyi calibration, last experiment	
99	15-Jan	19:45	23:00	Calibration	
100	16-Jan	9:15	16:45	Calibration	
101	16-Jan	19:45	22:45	Calibration	
102	17-Jan	10:00	16:30	Presentation and re-checking calculations	
103	18-Jan	9:15	17:15	Writing report and making new calculations	
104	18-Jan	19:15	22:30	Working on calculation of y	
105	19-Jan	9:15	14:30	Working on Matlab script	
106	20-Jan	10:30	14:30	Sketches in PowerPoint and writing report	
107	20-Jan	19:30	22:30	SolidWorks and writing report	
108	21-Jan	9:15	16:45	Discussing Matlab results, presentation workshop, writing report	
109	21-Jan	20:00	22:30	Writing report	
110	22-Jan	9:00	19:30	Writing report	
111	22-Jan	21:15	00:30	Writing report	
112	23-Jan	10:00	17:00	Finishing report	